Lecture Notes in Electrical Engineering 572

Qilian Liang
Wei Wang
Jiasong Mu
Xin Liu
Zhenyu Na
Bingcai Chen *Editors*

Artificial Intelligence in China

Proceedings of the International Conference on Artificial Intelligence in China

Springer

Lecture Notes in Electrical Engineering

Volume 572

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering - quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Associate Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Executive Editor (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada:

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries:

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** Indexing: The books of this series are submitted to ISI Proceedings, EI-Compendex, SCOPUS, MetaPress, Web of Science and Springerlink ****

More information about this series at http://www.springer.com/series/7818

Qilian Liang · Wei Wang ·
Jiasong Mu · Xin Liu · Zhenyu Na ·
Bingcai Chen
Editors

Artificial Intelligence in China

Proceedings of the International Conference on Artificial Intelligence in China

Springer

Editors

Qilian Liang
Department of Electrical Engineering
University of Texas at Arlington
Arlington, TX, USA

Jiasong Mu
Department of Electronic
and Communication Engineering
Tianjin Normal University
Tianjin, China

Zhenyu Na
School of Information Science
and Technology
Dalian Maritime University
Dalian, Liaoning, China

Wei Wang
Tianjin, China

Xin Liu
School of Information and Communication
Engineering
Dalian University of Technology
Dalian, Liaoning, China

Bingcai Chen
School of Computer Science
and Technology
Dalian University of Technology
Dalian, Liaoning, China

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-15-0186-9 ISBN 978-981-15-0187-6 (eBook)
https://doi.org/10.1007/978-981-15-0187-6

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Contents

Global Descriptors of Convolution Neural Networks for Remote Scene Images Classification

Q. Wang[1], Qian Ning[1,2(✉)], X. Yang[1], Bingcai Chen[3,4], Yinjie Lei[1], C. Zhao[1], T. Tang[1,2,3,4], and R. Hu[1,2,3,4]

[1] College of Electrical & Information Engineering,
Sichuan University, Chengdu 610065, China
ningq@scu.edu.cn

[2] School of Physics & Electronics,
Xinjiang Normal University, Urumqi 830054, China

[3] School of Computer Science & Technology at Dalian University
of Technology, Dalian 116024, China

[4] School of Computer Science & Technology at Xinjiang Normal University,
Urumqi 830054, China

Abstract. Nowadays, the deep learning-based methods have been widely used in the scene-level-based image classification. However, the features automatically obtained from the last fully connected (FC) layer of single CNN without any process have little effect because of high dimensionality. In this paper, we propose a simple enhancing scene-level feature description method for remote sensing scene classification. Firstly, the principal component analysis (PCA) transformation is adopted in our research for reducing redundant dimensionality. Secondly, a new method is used to fuse features obtained by PCA transformation. Finally, the random forest classifier applying to classification makes a significant effect on compressing the training procedure. The results of experiments on the public dataset describe that feature fusion with PCA transformation performs great classification effect. Moreover, compared with the classifier softmax, the random forest classifier outperforms the softmax classifier in the training procedure.

Keywords: Remote sensing image (RSI) · Global feature descriptors · Feature fusion · Scene classification

1 Introduction

More recently, in the field of RSI investigation, RSI scene classification [1] is one of the most important processes. For RSI classification, image semantic understanding is generally reflected by feature descriptors. Therefore, the key to classifying is features. In our research, we focus on the investigation of RSI feature

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 1–11, 2020.
https://doi.org/10.1007/978-981-15-0187-6_1

descriptors, premeditating the feature extraction as a critical step to obtain a great classification performance. Existing RSI features extraction research methods are primarily divided into local features extracting methods and global features extracting methods. For the former, there exist two types of methods, corner feature extraction methods and edge feature extraction methods. In the past, corner feature descriptor was the main job in feature extraction, such as features from accelerated segment test (FAST) [2], oriented fast and rotated brief (ORB) [2]. However, the curves of edge are discontinuous. Hence, they have limitations on the description of scene semantic information.

Numerous algorithms have been introduced for edge feature descriptor extraction. Signature of Histograms of Orientations (SHOT) [3] and Raster Operations Units (ROPS) [4] are based on the histogram statistics for local descriptors extraction, but they are low expression for semantic information in most of the experiments. In summarize, local descriptors extracting methods are considerably depending on the experiences of researchers, and higher requirements for extracted feature descriptors are necessary for these methods.

Among all kinds of methods to extract features, comprehensive feature descriptors can be obtained by deep learning-based methods, which help to enhance the ability of image description greatly. In 2012, Krizhevsky et al. [5] proposed a sensational deep learning neural network for image classification, which picks up the 2012 image recognition contest champion. From then on, the CNN architecture detonated the application boom of neural networks. Simultaneously, more deep CNN architectures were proposed after AlexNet [5], such as VGGNet [6], ResNet [7]. Due to the convenience of extracting features and the better result of classification, they are applied widely in many aspects of image recognition. Liu et al. [8] proposed to concatenate features extracted from convolutional layers of CaffeNet and VGG-VD16 to deep descriptors.

In this paper, we propose a method to accelerate the speed of RSI classification model training with significant performance increase. Specifically, we first obtain features from the last FC layer of two deep CNN models. Then we reduce the feature vector dimension utilizing PCA transformation. For accelerating the training speed and exploring a better effect on classification result, we propose an optimize and simple way to generate feature vector by concatenating the features from two types of different CNN models.

2 Proposed Method

The details of the proposed method are shown in Fig. 1. The main content consists of the following three parts: global descriptors extracting, feature fusion, and reducing dimensions. The procedure is organized as follows. The first part extracts global feature descriptors from the last fully connected layers, including a sample of the training set and the used pre-train CNN. Two types of pre-train CNN, VGGNet-16 and ResNet-50, are introduced to extract global feature descriptors. The second part is the details of the proposed method.

For the first step (Fig. 1a), the training set and testing set are used as the input of a CNN mode to extract the global feature descriptors from the last fully connected layer.

For the third step (Fig. 1b), the features of every input data obtained from the first part are concatenated to form the complete global feature descriptors. With a PCA transformation to reduce the dimension, the last classification results are given by the random forest classifier.

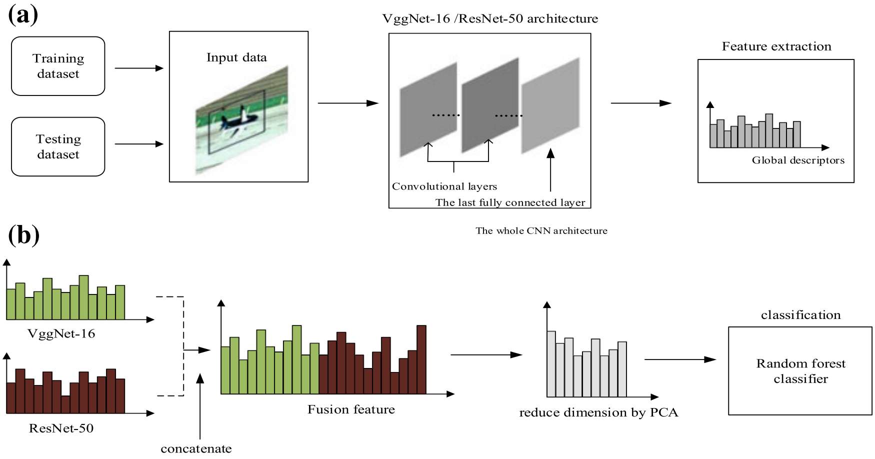


Fig. 1. Framework of proposed method. **a** Is the process of feature extraction. **b** Proposed method: concatenating features followed by PCA transformation.

2.1 Global Descriptors Extracting

In recent years, many CNN architectures have been proposed. Most of them perform a great effect on a large testing set. Such as VGGNet performs better on classification than AlexNet or CaffeNet. ResNet can obtain significant accuracy with deeper architecture and fewer parameters.At the first step, RSIs enter into VGGNet-16 and ResNet-50, respectively, for feature extracting, and the result is the global descriptor, which refers to the relationship between the extracted features and the entire image.

Feature extraction: A pre-train CNN mode serves as the feature extractor in many researches. As using the CNN as feature extractor, the minimal image shift has no effect on the last feature vectors because of the properties of convolution and pooling calculation. Hence, the obtained features have powerful fit abilities and make no influence on the classification result. In addition, because of this stability, it fits to every kind of image for feature extraction. When we apply a CNN for feature extraction, a popular feature extraction strategy is extracting an activation vector from the last fully connected layer (including the classifier layer) [9].

2.2 Feature Fusion and the Dimension Reduction

Existing deep learning pre-trained CNN models are used as the feature extractor to extract the feature from the final FC layers (include the last classification layer). Although the feature can be utilized to train the classifier directly, effective features extracted by only a single CNN pre-trained model are not enough, which will lead to a disappointing effect. Hence, an efficacious way to solve this question, feature fusion, is proposed.

Feature fusion: Deep feature fusion is a new solution to handle complex data. In 2014, two MIT engineers developed deep feature synthesis [10]. Most prediction decisions rely on the features descriptors based on input images in the vision classification tasks. Hence, it is necessary to overcome the obstacles of data dependence. Feature fusion refers to concatenating global features descriptors extracted from several different pre-trained CNN models. Vectors obtained by these models expand the dimension of the final vector by vector-spliced which is an efficient method to mitigate data dependence. In addition, the key advantage of feature fusion is new features obtained in this process, which can improve the performance of classification.

Reduce dimension: Dimensionality reduction, as the name implies, means feature selection and feature extraction. Since principal component analysis [11] was successfully proposed, applying PCA transformation to reduce dimension becomes a mainstream tendency. In the proposed method, PCA, as an important processing technology, is introduced for feature descriptors distinguishing and dimension reducing. With PCA more consummate, the field of data it can handle becomes wider. It is also the main technology for compressing the time of training process without losing the quality of a model. The main process for PCA is to transform the original data onto a set of linearly independent representations of each dimension through linear transformation, which reduces the dimension of input data set while maintaining the feature of the largest contribution of the data set in the data set. The final result is the key feature components of all the features.

3 Experiments and Analyses

3.1 UC Merced Land Use Dataset Description

In this section, we investigate the performance of the proposed methods on the "UC Merced Land Use" dataset[1] [12] extracted from large images from the US Geological Survey National Map Urban Area Imagery collection for various urban areas around the country. This dataset contains 21 classes. Each class includes 100 images with the size of 256 × 256 pixels in the color space of red–green–blue with different space structure, color distribute, region cover, and object cover. Every image is operated by rotating.

[1] http://vision.Ucmerced.edu/datasets/landuse.html.

3.2 Experimental Introduction

For input data, 75% images in each class serve as the training set and the remaining serve as the testing set. In experiments, the global feature descriptors are extracted from two pre-trained CNN models consisting of VGGNet-16 and ResNet-50, which are trained on the ImageNet dataset. The dimensions of global feature descriptors are 1×1000. The random forest is applied for training and classification. The confusion matrices performed the result of our proposed method; the training time is analyzed on the different methods to train.

3.3 Discuss Different CNN Models with PCA

The comparison between two types of CNN-based features classification is shown in Tables 1 and 2. The result of the classification on VGGNet-16-based and ResNet-50-based features without PCA transformation is displayed in Table 1 and the classification details of two types of CNN-based features with PCA transformation are performed in Table 2. The ratio of PCA transformation is set as 95%.The comparison in tables demonstrates that VGGNet-16 performs better classification effect. VGGNet-16-based features are better discrimination, especially in some similar categories, such as "beach," "parking lot," and "runway", which is because the initial parameters in VGGNet-16 are much richer.

From two tables, we can see that using PCA transformation performs greatly for both types pre-trained CNN models, especially for the ResNet-50, which has improved about 8%. In general, PCA transformation help to improve the description ability of the global features.

3.4 Analysis of the Proposed Method

In this section, we research the confusion matrix on the 21-classes public remote dataset. Figure 2 describes the confusion matrix of proposed method, which includes feature fusion and PCA transformation. The confusion matrix is analyzed via using 25% of the training dataset. As confusion matrix is shown, the entry in the ith row and jth column means the rate of RSI belongs to the ith class and classifies to jth class. The classification results are proposed as percentages.

In Fig. 2, the average accuracies of classification for proposed method are 86.78%. It performs great ability (the accuracy of classification $\geq$90%) on the classes, such as "airplane," "agricultural," "baseballdiamond," "chaparral," "forest," "golf course," "harbor," "overpass," "tennis court," "river." The features extracted from these RSI include considerable information, which helps classify correctly.

Moreover, some classes obtain poor classification effect, such as "dense residential," "medium residential." This is the reason that high dimensional features of these classes are too similar to distinguish. In addition, several classes include plenty of building elements, which results an error decision.

Table 3 presents the comparison between state-of-art methods and our proposed method. These existing methods are detailed in [8, 12–14]. As we can see

Table 1. Classification accuracies without PCA transformation. OA: overall average

Class	Airplane	Beach	Agricultural	Baseball-diamond	Buildings	Chaparral	Dense-residential	Forest	Freeway	Golfcourse	Harbor
ResNet-50	0.875	0.313	0.5	0.688	0.375	0.938	0.125	0.875	0.688	1.0	0.563
VggNet-16	1.0	0.688	0.75	0.438	0.625	1.0	0.5	1.0	0.813	0.875	0.813
Class	Intersection	Medium-residential	Mobile-homepark	Overpass	Parkinglot	River	Runway	Sparse-residential	Storage-tanks	Tennis-court	OA
ResNet-50	0.313	0.25	0.625	0.625	0.313	0.938	0.313	0.25	0.5	0.813	0.554
VggNet-16	0.5	0.563	0.5	0.688	0.635	0.75	0.375	0.625	0.625	0.813	0.693

Table 2. Classification accuracies with PCA transformation. OA:overall average

Class	Airplane	Beach	Agricultural	Baseball-diamond	Buildings	Chaparral	Dense-residential	Forest	Freeway	Golfcourse	Harbor
ResNet-50	1.0	0.438	0.438	0.25	0.563	0.875	0.188	1.0	0.625	0.875	0.875
VggNet-16	1.0	0.688	0.688	0.75	0.625	1.0	0.5	1.0	0.875	0.938	0.75
Class	Intersection	Medium-residential	Mobile-homepark	Overpass	Parkinglot	River	Runway	Sparse-residential	Storage-tanks	Tennis-court	OA
ResNet-50	0.438	0.75	0.688	0.688	0.325	0.688	0.625	0.313	0.563	0.875	0.639
VggNet-16	0.438	0.688	0.688	0.5	0.75	0.75	0.188	0.563	0.688	0.813	0.715

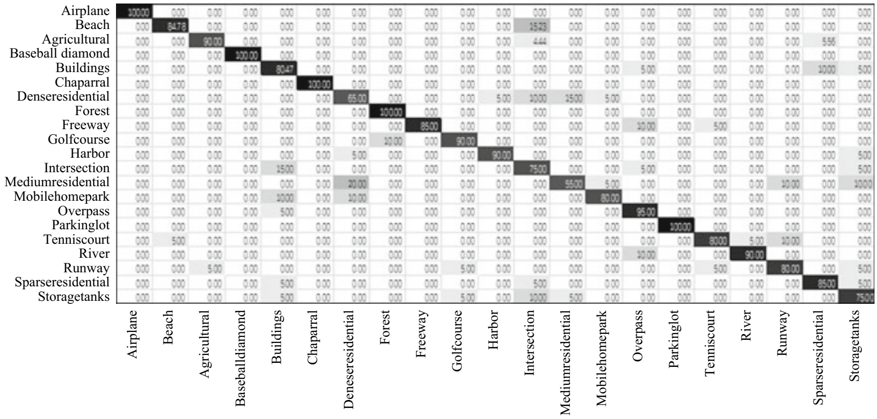


Fig. 2. Confusion matrix for 21-category performed result of the proposed method (86.78%)

that our proposed method enhances the classification effect 1.69% over the best-existed results in [8,12–14]. To sum up, our approach achieves a more favorable effect on this dataset because of the combination of feature fusion and PCA transformation.

Table 3. Overall classification accuracies on dataset

	Approaches	Overall accuracies (%)
State-of-art	BOVW [12]	76.81
	Texture [12]	76.91
	spck++ [14]	77.38
	Approach of [13]	75.33
	Strategy 1 of [8]	85.09
	Our method	**86.78**

4 Discussion of Time

In this part, the time spent on the training process with different classifiers is compared. As Table 4 shown, applying Caffe framework to train the CNN architectures, VGGNet-16 and ResNet-50, based on GPU for acceleration need 4376 and 2437 s, individually. However, our proposed method with random forest for classifying based on the 21-category dataset just needs 24.81 seconds, which shortens over one hundred times. In addition, the classification accuracies of our method are 86.78%, which is higher than VGGNet-16 and ResNet-50. Moreover, after analyzing, PCA transformation helps to reduce the time for training, too.

Table 4. Time for training a model on dataset

Classifier	Pretain model	Accuracy (%)	Training time (s)
Softmax	VGGNet-16	78.3	4376
	ResNet-50	82.4	2437s
Random forest (with PCA)	VGGNet-16	70.8	15.73
	ResNet-50	63.7	10.45
	Our method	86.78	24.81
Random forest (no PCA)	VGGNet-16	72.3	54.07
	ResNet-50	56.9	55.45
	Fusion-feature	85.24	52.00

In Fig. 3a, two curves display two situations about the proposed method with several numbers of trees in a random forest. With the number increasing, the spending time increases, too. The gray line demonstrates the features without PCA transformation, training spending is slower than the other situation. In addition, as Fig. 3b shown, while the dimension (For proposed method, corresponding to the PCA ratio 90–99%, the dimension number of PCA transformation is separately 104, 116, 130, 147, 168, 194, 227, 274, 346, 487) number of PCA transformation increasing, the training process gets longer, too. On the other hand, when the PCA transformation ratio obtains 95% (The dimension numbers is 194), the classification accuracy attains the highest, which is 86.78%. Overall, whatever for time spending or classification performance, our proposed method performs a better effect on the total datasets.

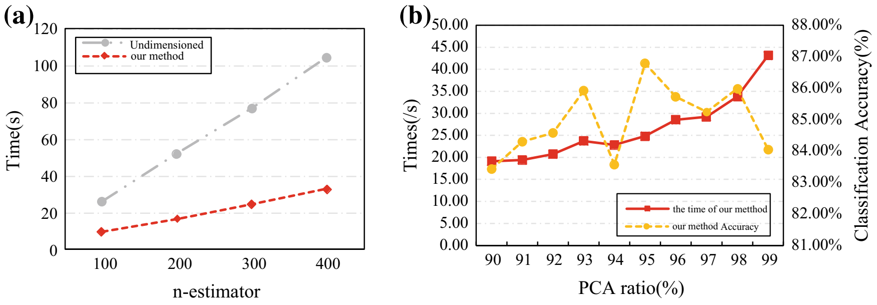


Fig. 3. **a** Shows time changing with different n_estimators; **b** shows the Pca transformation ratios influence the changing of training time and classification accuracy

5 Conclusion

In this paper, we investigate the global feature descriptors extracted from the last FC layer of CNN pre-trained models. Comparing to a single-model feature, the fusion features are more representative. Both pre-trained CNN-based features, VGGNet-16 and Resnet-50, are proposed to form the last global feature descriptors. The proposed method focuses on the most contribution features in both different CNN models. The result of experiment illustrates that feature fusion with a suitable dimension number of PCA transformation can enhance the performance of classification. Comprehensive evaluations of the public RSI scene classification perform the model training efficiency.

References

1. Bian X et al (2017) Fusing local and global features for high-resolution scene classification. IEEE J Sel Top Appl Earth Obs Remote Sens 10(6):2889–2901
2. Rublee E et al (2011) ORB: an efficient alternative to SIFT or SURF. In: 2011 IEEE international conference on computer vision (ICCV). IEEE
3. Salti S, Tombari F, Di Stefano L (2014) SHOT: unique signatures of histograms for surface and texture description. Comput Vis Image Underst 125:251–264
4. Tombari, F, Salti S, Di Stefano L (2010) Unique signatures of histograms for local surface description. In: European conference on computer vision. Springer, Berlin, pp 356–369
5. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
6. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
7. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
8. Liu N et al (2018) Exploiting convolutional neural networks with deeply local description for remote sensing image classification. IEEE Access 6:11215–11228
9. Nogueira K, Penatti OAB, dos Santos JA (2017) Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recogn. 61:539–556
10. Kanter JM, Veeramachaneni K (2015) Deep feature synthesis: Towards automating data science endeavors. In: 2015 IEEE international conference on data science and advanced analytics (DSAA), 36678. IEEE, pp 1–10
11. Ke,Y, Sukthankar R (2004) PCA-SIFT: a more distinctive representation for local image descriptors. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004, vol 2. IEEE
12. Yang Y, Newsam S (2010) Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems. ACM, pp 270–279

13. Liu Q, Hang R, Song H, Li Z (2018) Learning multiscale deep features for high-resolution satellite image scene classification. IEEE Trans Geosci Remote Sens 56(1):117–126
14. Yang Y, Newsam S (2011) Spatial pyramid co-occurrence for image classification. In: 2011 IEEE international conference on computer vision (ICCV). IEEE, pp 1465–1472

Plant Diseases Identification Based on Binarized Neural Network

Xiufu Pu[1], Qian Ning[1,2](✉), Yinjie Lei[1], Bingcai Chen[3,4], Tiantian Tang[2], and Ruiheng Hu[2]

[1] College of Electronics and Information Engineering, Sichuan University,
Chengdu 610065, China
ningq@scu.edu.cn

[2] College of Physics and Electronics Engineering, Xinjiang Normal University,
Urumqi 830054, China

[3] College of Computer Science and Technology Engineering,
Dalian University of Technology, Dalian 116024, China

[4] College of Computer Science and Technology Engineering,
Xinjiang Normal University, Urumqi 830054, China

Abstract. Although the use of the convolutional neural network (CNN) improved the accuracy of object recognition, it still had a long-running time. In order to solve these problems, the training and testing datasets were split at four different proportions to reduce the impact of inherent error. Using model fine-tuning, the model converged in a small number of iterations, and the average recognition accuracy of BWN test can reach 96.8%. In the segmented dataset, the recognition accuracy of the former was 4.7 percentage points higher than the latter by comparing color dataset and grayscale dataset, which proved that a certain amount of color features will have a positive impact on the model. The segmented dataset was 0.9 percentage points higher than the color dataset; it shows that the model focused more on features of contour and texture by eliminating the background of images. The experiments showed that the binarized convolutional neural network can effectively improve recognition efficiency and accuracy compared with traditional methods.

Keywords: Agricultural diseases · Binarized model · Image classification

1 Introduction

Plant diseases are one of the three natural disasters in China. In China, more than 250 billion kg of grain, fruits, vegetables, oil, and cotton is lost every year.

Visual inspection is one of the main traditional methods for diagnosing plant diseases. However, there are two problems: The judgments made by farmers based on experience are not all correct and the situation of plants will be worse without timely and effective treatment for diseases [1]. In order to realize the diagnosis of agricultural diseases rapidly and accurately, researchers have explored methods for identifying multiple plant diseases [2], using machine learning and image processing technology. Convolutional neural network is good at extracting the features of contour and texture

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 12–19, 2020.
https://doi.org/10.1007/978-981-15-0187-6_2

of leaves. Generally speaking, the deeper the network, the more the parameters and the larger models. It spends a long time to classify objects. In order to solve these problems, this paper proposes an effective way to classify plant diseases by using binarized convolutional neural networks. It aims to speed up the operation by binarizing the parameters.

2 Related Work

Kan et al. [3] extracted the contours and texture features of the leaves by radial basis function (RBF) neural network. The average recognition rate is 83.3%. Tan et al. [4] can effectively identify lesion area of soybean by calculating the color value of leaves and creating a multilayered BP neural network. The average recognition accuracy can reach 92.1%. Zhang et al. [5] used the cloning algorithm and K-nearest neighbor algorithm to classify leaves, which achieved a recognition rate of 91.37%. Wang et al. [6] used the support vector machine (SVM) to identify the leaves; the accuracy of the classifier can reach 91.41%.

Dyrmann et al. [7] used convolutional neural networks to classify plant images taken with mobile phones, and the average recognition accuracy reached 86.2%. Mohanty et al. [8] carried out experiments on 26 diseases of 14 kinds of plant; they choose two models, three datasets, and five datasets with different proportions, which also achieved good results. Lee et al. [9] explored how the CNN extracts features of leaves. They tested different methods and contrasted various experimental results. The results show that the CNN has a better effect on classification.

3 Dataset

This paper obtained 54,306 leaf images from PlantVillage by color, grayscale, and segmentation, which contains a total of 38 plant types. According to the number of different types of leaves in the dataset, the number of different kinds of leaf images ranges from 64 to 1166. In order to make up for the shortcomings, this paper expands the dataset by enhancing the exposure of the blade, changing the color of the image, and rotating the image. Rotating the image is to eliminate the effects of inherent bias [10]. Then, the images are labeled by category, and center cropped to a size of 224×224.

4 Convolutional Neural Networks

Generally speaking, to classify two objects there are two steps: data forward transmission and weight updating. When starting to train a model, there are two choices: one is to completely reset the parameters and train from scratch; the other is model fine-tuning. And the latter was used in this paper. Generally, the basic structure of CNN includes a feature extraction layer and a feature mapping layer. The former includes a convolution layer and a pooling layer, where each neuron is connected to the acceptance domain of the previous layer, and then extracts features of that acceptance domain [11]. The process of forward transmission is shown in Formula 1.

$$f_{\text{out}} = f(w\ldots(wf(wx+b)+b)\ldots+b) \tag{1}$$

The back-propagation process is actually a process of weight update. In this paper, SGD and Adam are used to update the weight. The process of feature extraction and weight update is continuous, until the global optimal solution is found. In most experiments, the results are local optimum; it is necessary to adjust the learning rate and select the loss function, so that the network can find the global optimum solution.

The parameter update is shown in Formula (2).

$$W_{ij}^{l} = W_{ij}^{l} - \alpha \frac{\partial}{\partial W_{ij}^{(l)}} J(W, b) \tag{2}$$

5 Binarized Convolutional Neural Network

In the article [12], the parameters with single-precision floating point are converted into parameters that only occupy 1 bit, which theoretically reduces the memory space by 32 times. Moreover, the speed of the model is accelerated to about twice as fast. Symbolic functions can be expressed as:

$$\operatorname{sign}(x) = \begin{cases} 1 & \text{if}\, x > 0 \\ -1 & \text{if}\, x \leq 0 \end{cases} \tag{3}$$

The binarized convolution neural network converts the weights and the activation values of hidden layers into 1 or −1. There are two ways of conversion: deterministic method and stochastic method. The former is simple and intuitive. If the weight or activation value is greater than 0, it is converted to 1; if it is less than 0, it is converted to −1. The latter calculates a probability p for the input. When p is greater than a threshold, it is +1, otherwise it is −1. Since the random numbers generated by hardware, which is more difficult to implement. Therefore, the first method is adopted in this paper.

In this paper, we take a method of approximating the real weights by using a binarized weight B and a scale factor α. The process of conversion is shown in Formula (4), where ⊕ represents the convolution operation of input and binarization weights.

$$I * W \approx (I \oplus B)\alpha \tag{4}$$

The binarized VGG16 network is used for experiments in this paper. Same as ordinary convolutional neural networks, the binarized network also includes the input layer, the hidden layer, and the output layer. The hidden layer includes convolution layer and pooling layer, where binarized convolution and pooling operation are used.

The forward transmission process of the binary network can be divided into four steps: first let the input pass through a Batch Normal, then binarize the input value, and binary convolution and pooling are used. Finally output the classification through a classifier. The weight is updated with full precision during the training process.

The forward transmission of binarized network:

$$x^k = \sigma\left(\mathrm{BN}\left(W_b^k \cdot x^{k-1}\right)\right) = \mathrm{sign}\left(\mathrm{BN}\left(W_b^k \cdot x^{k-1}\right)\right) \tag{5}$$

The process of binarized model update is shown in Fig. 1.

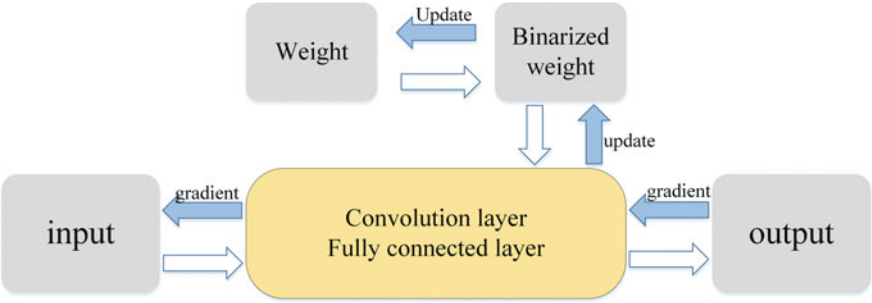


Fig. 1. Schematic diagram of binarized model training

6 Training and Discussion

6.1 Experiment Platform

The experimental environment is Ubuntu 16.04 LTS 64-bit system, using PyTorch as the deep learning open-source framework and using python as the programming language. The computer memory is 64G, equipped with Intel Xeon(R) CPU E5-1640 v4 3.6 GHz x12 processor. The graphics card is a NVIDIA GTX1080Ti.

6.2 Parameter Selection in Experiment

The train and test dataset are divided into multiple batches in this paper; each batch has 32 images. The full-precision model uses SGD to optimize the model with a learning rate of 0.005 and regularization coefficient of 5e−4. The learning rate become 0.1 times of the original per 20 epoch. The binarized model uses Adam to optimize the model with learning rate of 0.001, regularization coefficient of 1e−5. The learning rate become 0.1 times of the original per 30 epoch.

In order to prevent over-fitting, four different proportions are set: train set: 80, 60, 30, 20%; test set: 20, 30, 60, 80%. The more the sample ratio, the smaller the influence of experimental inherent on the results. At the same time, in order to make a fair comparison, the hyper-parameters are standardized.

In order to compare the effect of the dataset under different conditions, the collected images are divided into color dataset, gray dataset, and segmented dataset. The segmented dataset eliminates the influence of the background on the picture.

6.3 Analysis and Discussion

The results are shown in the line chart. It can be seen from the chart, under the same condition, the average accuracy of the full-precision model is slightly higher than that of the binarized model. Due to the high initial learning rate, the accuracy of the two networks is improved rapidly before 20 epochs, and eventually, it tends to be stable.

It can be seen from the green polyline, because of the fine-tuning of the model the initial accuracy of the full-precision model is higher. The color, segmentation, and grayscale datasets, respectively, reach the accuracy of 0.6, 0.7, and 0.5. For the red polyline, even if the parameter of trained model is used, the binarized parameter leads to lower initial accuracy, but the accuracy tends to be stable after 30 epochs. Four proportion datasets are set in this paper. It can be seen from the last line chart that the green polyline and red polyline have a higher accuracy, followed by yellow and blue polyline. It proves that the more the training sample, the more features can be extracted to train networks (Fig. 2).

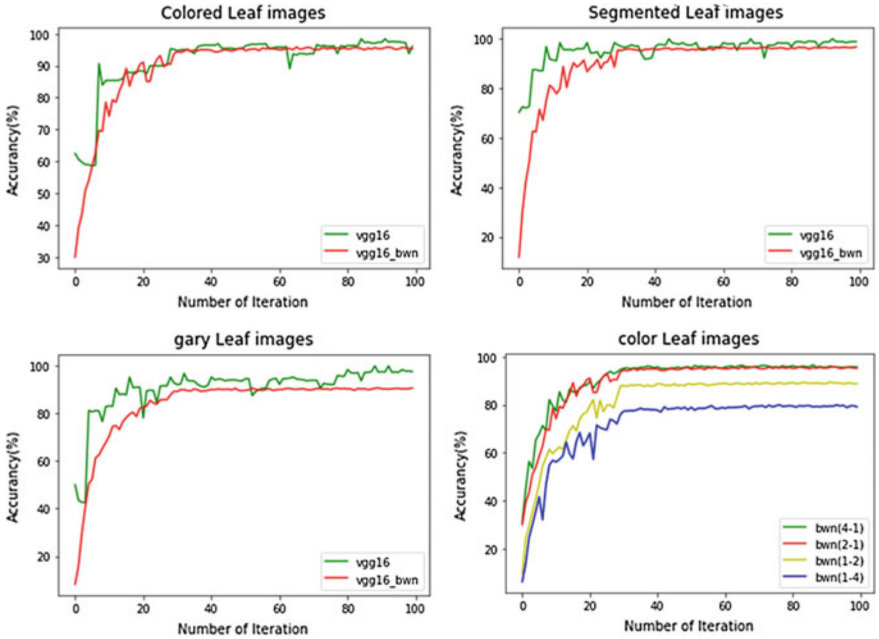


Fig. 2. Line chart of experimental results

In this paper, it is shown in Table 1 that the convolutional neural network is more suitable for plant detection compared with the traditional methods. Most of the traditional methods rely on features of manual extraction but such features cannot fully reflect the diseases. The convolutional structures can extract features automatically as it has the ability to eliminate interference caused by noise. In this paper, softmax is used to classify leaves, and the average accuracy can reach 99.0% on segmented datasets.

Table 1. Comparison of experimental results from different methods

Identification methods	Average accuracy (%)
RBF recognition	83.3
K recognition	90.0
BP recognition	92.1
SVM recognition	91.1
CNN recognition	99.0

The VGG16 has a huge amount of float parameters, which can extract picture features more comprehensively. When the floating point parameters are binarized, the model loses part of the features, which makes the features blurred and reduces the expression ability of the model. From another point of view, it speeds up the calculation of the model. In the experiment, the average recognition accuracy of the full-precision model is slightly higher than that of the binarized model. The former can reach 99.0%, and the latter can reach 96.8%. In terms of time, the speed of forward transmission of the latter is 2.7 ms per picture, which is about twice as fast as the former. That is to say the binarized model gets faster speed by losing part of its accuracy (Table 2).

Table 2. Comparison of full-precision model and binarized model

Neural networks	Segmented images	
	Classification accuracy (%)	Transmission rate (ms)
Full-precision network	99.0	2.7
Binarized network	96.8	1.5

In this paper, three kinds of datasets are chosen. It can be seen from Table 3 that the model performs best under the segmented dataset. The average accuracy can reach 96.8%. But the accuracy of the color dataset is 0.9% lower than the segmented dataset, which indicates that the model pays more attention to the features of leaf diseases. The accuracy of grayscale dataset is reduced by 4.7% compared with the color dataset, which means that in addition to some physical features such as contours and veins, color can also have a positive effect on the plant identification.

Table 3. Comparison of experimental results from different datasets

Network types	Datasets (accuracy)		
	Color (%)	Segmented (%)	Gary (%)
Binarized networks	95.9	96.8	91.2

Not all the data can be used for training. In the experiment, the overexposed images are generated due to the randomness of the parameters, which directly covered the features of texture and contour. The model could not extract useful features, so the accuracy is not high. The overexposed images should be deleted.

6.4 Conclusion

In this paper, the PlantVillage dataset and extended dataset are selected, and the binarized model is used to identify plant diseases. The experiment shows that the full-precision model and the binarized model both have the best performance under the segmented dataset, which can reach high accuracy and spend less time. Comparing the three datasets, the physical features such as leaf outline and plant meridians, the features of color, and background also have a great impact on the model. Comparing the convolutional models with the traditional models, the former can extract more details for training, also it can adapt to a complex environment.

The binarized model can work well in experiment, and the calculation speed is twice as fast as the full-precision model, which provides a basis for plant disease research.

References

1. Cao X, Zhou Y et al (2016) Progress in monitoring and forecasting of plant diseases. J Plant Prot 3:1–7
2. Zhang R, Wang Y (2016) Research on machine learning with algorithm and development. J Commun Univ China (Sci Technol) 23(2):10–18
3. Kan J, Wang Y, Yang X et al (2010) Plant recognition method based on leaf images. Sci Technol Rev 28(23):81–85
4. Tan F, Ma X (2009) The identification of plant diseases based on leaf images. J Agric Mech Res 31(6):41–43
5. Zhang N, Liu W (2013) Plant leaf recognition method based on clonal selection algorithm and K nearest neighbor. J Comput Appl 33(07):2009–2013
6. Wang L, Huai Y, Peng Y (2007) Method of identification of foliage from plants based on extraction of multiple features of leaf images. J Beijing For Univ 165:535–547
7. Dyrmann M, Karstoft H, Midtiby HS (2016) Plant species classification using deep convolutional neural network. Biosyst Eng 151:72–80
8. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419
9. Lee SH, Chan CS, Mayo SJ et al (2017) How deep learning extracts and learns leaf features for plant classification. Pattern Recogn 71:1–13

10. Zhong Z, Zheng L, Kang G et al (2017) Random erasing data augmentation. arXiv preprint arXiv:1708.04896
11. Ghazi MM, Yanikoglu B, Aptoula E (2017) Plant identification using deep neural networks via optimization of transfer learning parameters. Neurocomputing 235:228–235
12. Courbariaux M, Bengio Y (2016) BinaryNet: Training deep neural networks with weights and activations constrained to +1 or −1. arXiv:1602.02830

Hand Detection Based on Multi-scale Fully Convolutional Networks

Yibo Li, MingMing Shi, and Xiangbo Lin(✉)

Faculty of Electronic Information and Electrical Engineering,
Dalian University of Technology, Dalian, China
linxbo@dlut.edu.cn

Abstract. Accurate hand detection is a challenging task because of large variations of hand images in real-world scenarios. We present a simple yet powerful multi-scale fully convolutional network structure that yields fast and accurate hand detection on challenging VIVA dataset. The proposed model directly detects and locates hands in driver's cab of various size, shape, appearance, and illumination in full images without time-consuming region proposal step. The simple model with the well-designed loss functions promotes the proposed approach to achieve very good hand detection results.

Keywords: Hand detection · Convolutional neural network · Multi-scale fusion

1 Introduction

Hand detection from RGB images is a fundamental task in a lot of applications, such as virtual reality, human activity monitoring, and robotic behavior demonstration. Though many progresses have been made [1–3], they are still far from mature due to plenty of difficulties in practice. Hands in an image often occupy small regions with resolutions not high enough. Often their detection is seriously influenced by highly occlusions, varied shape and appearances, different lighting conditions, and viewpoints. Due to the less accurate detection performance, labor consuming manual detection has to be used in many practical scenes. In this paper, we focus on hand detection problem in driver's cab environment by exploring more efficient and accurate hand detection method.

Traditional methods use handcraft features to detect human hands in the images. These approaches suffer from limited discriminative capabilities and will not get reliable detections in complex situations. Along with the success of deep learning in the computer vision field, convolutional neural networks (CNN) quickly spread in multiple research areas. Many powerful object detection CNN frameworks are proposed, such as variants of R-CNNs and YOLOs. However, these CNN networks are usually effective to detect relatively large objects on several public datasets with limited category of objects. This paper focuses on hand detection, and a specific network is proposed using multi-scale fully convolutional structure. Both global and local context information are efficiently synchronized and simultaneously represent the hand features. In addition to geometric coordinates of the hand bounding box, a ROI region-related score map is used to form the loss function. A balanced cross-entropy loss is used to facilitate a

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 20–27, 2020.
https://doi.org/10.1007/978-981-15-0187-6_3

simpler training procedure. The proposed hand detection network structure can be seen in Fig. 1. The experiments are presented on the challenging ‘Vision for Intelligent Vehicles and Applications (VIVA) Challenge’ hand databases [1], and our method yields very good results in the hand detection problem.

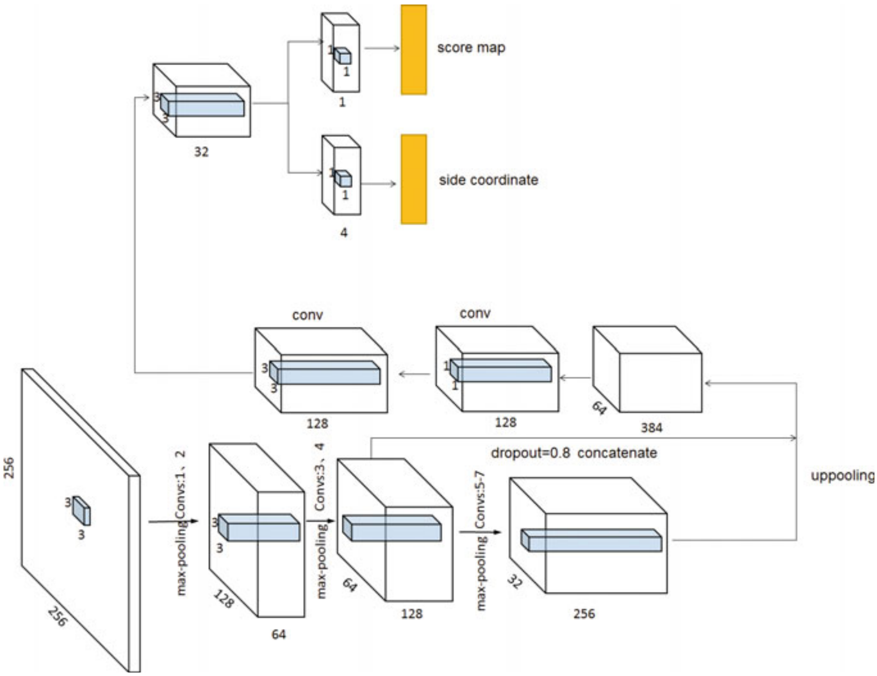


Fig. 1. Overall network structure

2 Related Works

Using skin information to detect hand in RGB images is an effective strategy, which has been proved by many previous methods. Mittal et al. [2] propose a two-stage method. The detectors based on context, skin, and sliding window shape with a classifier are used to propose hand bounding boxes and to get a final confidence score. However, detections from images under poor illumination encounter problems due to the bad skin-based features. Ohn-Bar et al. [3] extract HOG features from RGB-D multimodal data with linear kernel SVM to find the hand positions for analyzing the driver’s activities. Since their main aim is to describe the driver’s state, less attention is paid on the detection accuracy. Li and Kitani [4], Zhu et al. [5] propose shape-aware structured forests to detect hand region and get good performance in egocentric videos. However, such pixel-wise scanning in the image is quite time consuming. Other methods for hand detection use the human graph structure as spatial context. But, they need the visibility of most parts of human [6, 7].

All in all, the previous approaches have a common need to extract suitable handcraft features beforehand, which is a challenging task. Thanks to the development of neural networks, the new deep neural network approaches provide a hopeful way to solve the problems. However, the main concerns of these popular networks are detecting relatively large objects in the images. Their abilities are limited when they are employed to detect small objects. More specific network structures should be designed for different applications, such as hand detection. Le et al. [8], Hoang Ngan Le et al. [9] solve the problem by synchronizing the global and the local context features using Faster R-CNN for semantic detection. Ding et al. [10] adopt multi-scale CNN networks to detect hand. Their region proposals are generated at multiple scales; then the feature maps extracted from different layers are fused to get the hand bounding boxes. Deng et al. [11] design a network to detect hand region and hand in-plane rotation firstly and then give the final hand detection results by feature sharing. Although these RPN-based methods achieve better hand detection results than traditional models, they need applying hundreds of times per region sub-network, which is time consuming. Different from the above-mentioned networks, our new hand detection network is efficient, as well as better or comparative detection accuracy.

3 Methods

3.1 Overview of the Approach

The object of our work is to detect the hand and report their positions from RGB images. Since the hand sizes, shapes, and appearances exist in large variants, feature maps from different levels should be integrated to preserve both the global and local cues. So, we use multi-scale structure to extract feature maps with different resolution, then gradually merge these feature maps to form identifiable features. Supervised by the object region confidence and the coordinates of the bounding boxes, the network can predict the hand region accurately. The whole network model is shown in Fig. 1. The model is lightweighted, pixel-wise scoring, no time-consuming candidate proposal step, and very efficient.

3.2 Network Structure

For object detection, region proposal-based CNN is an important branch. The classical and representative networks are R-CNN and its variants. R-CNN applies deep convolutional layers as feature extractor to classify given region proposals. It is tailored by support vector machines (SVM) for object detection and bounding box regression. R-CNN can achieve accurate results, but is quite time consuming. Fast R-CNN and Faster R-CNN accelerate the detections by feature sharing, ROI pooling, and special region proposal network, but they still lack high efficiency due to the dependence on the external region proposal methods. Moreover, they are difficult to detect relatively small hands in cars because of the ROI pooling.

To improve efficiency, in our model, as shown in Fig. 1, the region proposal network is discarded. The object region is determined by the confidence of every pixel using the fully convolutional network, similar to the general design of DenseBox [12]. The proposed model has two modules. The first module is used for feature extraction. Since the hand size, shape, appearance, and illuminations have large variance, we merge the feature maps from different scales to integrate both the global and the local information. To reduce the computation overhead due to merging too many channels on large feature maps, we use three successive 1 × 1, 3 × 3, 3 × 3 convolutional layers to lower the subsequent computational cost. The second module is the output part, which includes two branches. The first branch outputs the confidence of each pixel within the hand bounding boxes, and the second branch outputs the vertex coordinates of the hand bounding boxes.

3.3 Loss Function

The loss function has two terms, as described in (1)

$$L = L_c + L_r. \tag{1}$$

where L_c evaluates the probability that a pixel locates inside the bounding box or not, and L_r evaluates the errors between the regressed position and the ground truth.

L_c is defined using cross-entropy loss, given by

$$L_c = -\alpha p^*(1-p)^\gamma \log p - (1-\alpha)(1-p^*)p^\gamma \log(1-p). \tag{2}$$

where p^* represents the true class of a pixel, p represents the predicted probability of a pixel inside the hand bounding box, $\alpha = 1 - \frac{N(p^*=1)}{N(p^*)}$ is used to balance the impacts of the positive and negative samples, and γ is the adjustable parameter. In our experiments, the best choice of γ is $\gamma = 2$.

L_r is defined using smoothed-$L1$ loss, given by

$$L_r = \text{smoothed}_{L1}\left(C_i - C_i^*\right) \tag{3}$$

$$\text{smoothed}_{L1}(x) = \begin{cases} 0.5x^2 & |x| < 1 \\ |x| - 0.5 & \text{others} \end{cases}. \tag{4}$$

where C_i and C_i^* represent regressed vertex coordinates of the hand bounding box and the ground truth, respectively.

3.4 Implementations

The size of input images is 128 × 128. The network is trained end to end using Adam optimization algorithm with minibatch size 24. The initial learning rate is 10^{-3}, decays 10 times every 20,000 minibatches, and stops at 10^{-5}.

4 Experiments and Results

4.1 Dataset

The dataset is from the VIVA Challenge, which consists of 54 videos from seven viewpoints with annotations of hands of drivers and passengers using 2D bounding boxes. These videos were collected in naturalistic driving environments with illumination changes, cluttered backgrounds, quick hand motions, and common occlusion. All the images are divided into 5500 training and 5500 testing data for standard evaluation.

4.2 Data Augmentation

Rich data is better for training a good model in deep neural networks. To get more training data, we did data augmentation using rotation, translation, Gaussian blurring, and sharping operations.

- *Augmentation rule 1*: The ratio of brightness enhancement is (1.2–1.5), the scaling factor is (0.7–1.5), and the upper limit of the translation is 40 pixels along x axis and 60 pixels along y axis.
- *Augmentation rule 2*: Boundary clipping range is (0–16) pixels. Randomly select 50% images and did horizontally flip.
- *Augmentation rule 3*: Vertically flip the images with added Gaussian blurring. The kernel size is (0–3.0).
- *Augmentation rule 4*: Rotate the images randomly. The largest rotation angular is 45°. Add Gaussian white noise with standard deviation 0.2. Randomly select 50% images and do the sharping process.

After data augmentation, the total amount of training data is 22,000. They are split into training subset and validation subset. The ratio is 9:1.

4.3 Label Generation

Since the score of each pixel is computed in the output part, the original labels in the bounding boxes should be further processed to fulfill the requirements. The original bounding boxes are zoomed out lightly to get much tighter object regions. The labels are set to be 1 for those pixels within the zoomed bounding boxes and 0 for other pixels.

4.4 Evaluation Criteria

The proposed model is evaluated on VIVA dataset using several criteria.

The visual inspections are listed in Fig. 2. It shows good hand detection results in some typical instances, such as varied illuminations, different hand shapes and sizes, and different hand numbers in one image.

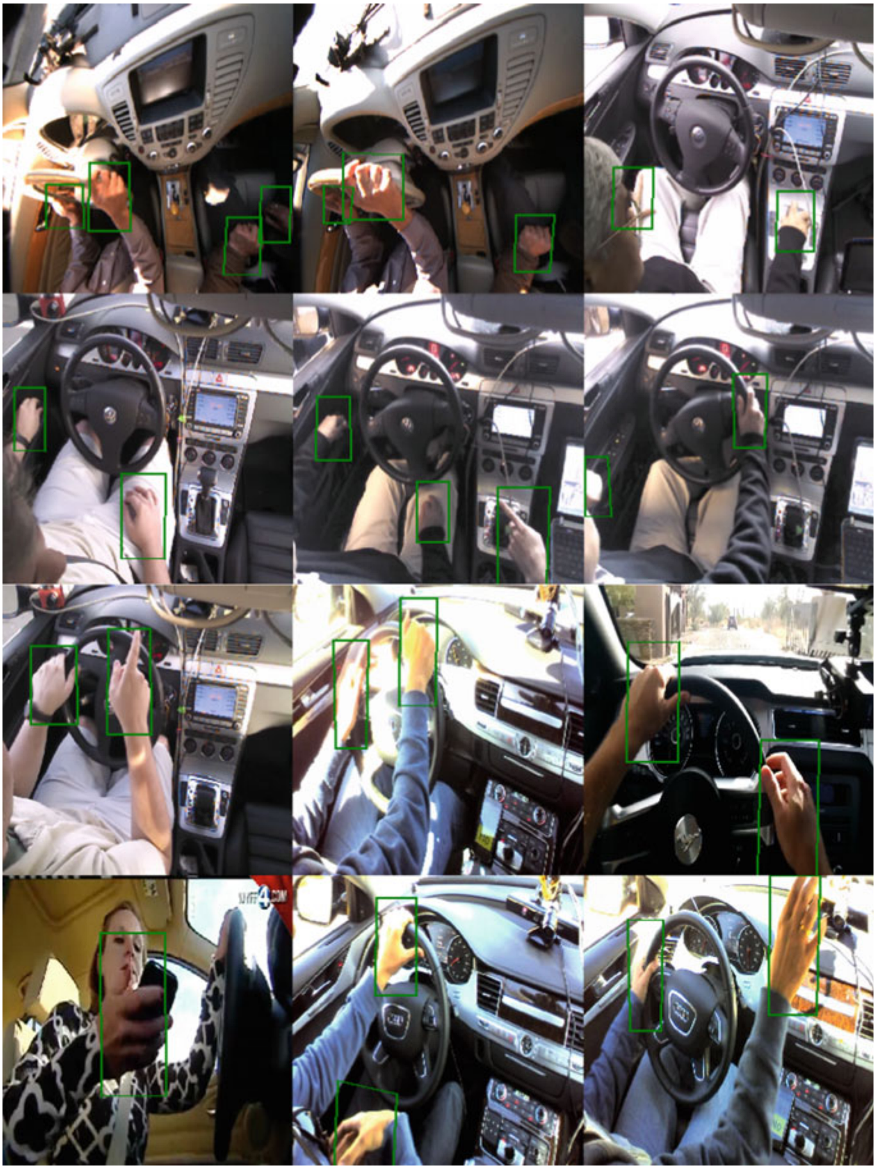


Fig. 2. Typical hand detection results

The average precision (AP), average recall (AR) rate, *F*-score, and detection time FPS are used as the quantitative criteria.

Let True Positive (TP) represents a correct detection, False Positive (FP) represents a false detection, and False Negative (FN) represents missed detection; the AP, AR, and *F*-score are defined as follows:

$$\mathrm{AP} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{5}$$

$$\mathrm{AR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{6}$$

$$F = \frac{2 * \mathrm{AP} * \mathrm{AR}}{\mathrm{AP} + \mathrm{AR}} \tag{7}$$

The proposed approach is compared with some state-of-the-art methods [9, 10, 13, 14] on VIVA dataset. The results are shown in Table 1. Our detection is faster than the other methods, the AP value is the highest, and the AR value is the second best.

Table 1. Quantitative comparisons of different methods on VIVA dataset

Methods	AP (%)	AR (%)	*F*	FPS
Ours	98.3	86.7	92.1	42
[9]	94.8	74.7	83.6	4.7
[10]	93.5	91.4	92.4	5
[13]	73.3	69.9	71.6	–
[14]	65.1	47.1	54.7	–

5 Conclusions

A hand detector from full images using a single neural network is proposed in this paper. By incorporating proper loss functions and pixel-wise scoring, the proposed model can detect human hand under different conditions, such as occlusions, varied illuminations, varied hand pose, shape, and size. The proposed model is simple, accurate, and efficient, which has been proved by the experimental results on challenging VIVA dataset.

References

1. The Vision for Intelligent Vehicles and Applications (VIVA) Challenge, Laboratory for Intelligent and Safe Automobiles, UCSD. http://cvrr.ucsd.edu/vivachallenge/
2. Mittal A, Zisserman A et al (2011) Hand detection using multiple proposals. In: Hoey J, McKenna S, Trucco E (eds) Proceeding of the British machine vision conference, vol 75. BMVA Press, pp 1–11
3. Ohn-Bar E, Martin S et al (2014) Head, eye, and hand patterns for driver activity recognition. In: The 22nd international conference on pattern recognition (ICPR). IEEE Press, Stockholm, pp 660–665
4. Li C, Kitani K (2013) Pixel-level hand detection in ego-centric videos. In: Proceedings of computer vision and pattern recognition (CVPR). IEEE Press, Portland, pp 3570–3577

5. Zhu X, Jia X et al (2014) Pixel-level hand detection with shape aware structured forests. In: Cremers D, Reid I, Saito H, Yang MH (eds) Processing of Asian conference on computer vision (ACCV). LNCS, vol 9006. Springer, Cham, pp 64–78
6. Karlinsky L, Dinerstein M et al (2010) The chains model for detecting parts by their context. In: Proceedings of computer vision and pattern recognition (CVPR). IEEE Press, San Francisco, pp 25–32
7. Ghiasi G, Yang Y et al (2014) Parsing occluded people. In: Proceedings of computer vision and pattern recognition (CVPR). IEEE Press, Columbus, pp 2401–2408
8. Le THN, Zheng Y et al (2016) Robust hand detection in vehicles. In: 23rd international conference on pattern recognition (ICPR). IEEE Press, Cancun, pp 573–578
9. Hoang Ngan Le T, Quach KG et al (2017) Robust hand detection and classification in vehicles and in the wild. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW). IEEE Press, Honolulu, pp 1203–1210
10. Ding Lu, Wang Yong et al (2019) Multi-scale predictions for robust hand detection and classification. Multimed Tools Appl. https://doi.org/10.1007/s11042-019-08080-4
11. Deng X, Yuan Y et al (2018) Joint hand detection and rotation estimation by using CNN. IEEE Trans Image Process 27(4):1888–1900
12. Huang L, Yang Y et al (2015) DenseBox: unifying landmark localization with end to end object detection. arXiv: 1509.04874
13. Zhou T, Pillai V et al (2016) Hierarchical context-aware hand detection algorithm for naturalistic driving. In: IEEE 19th international conference on intelligent transportation systems (ITSC). IEEE Press, Rio de Janeiro, pp 1291–1297
14. Das N, Ohn-Bar E et al (2015) On performance evaluation of driver hand detection algorithms: challenges, dataset, and metrics. In: IEEE 18th international conference on intelligent transportation systems. IEEE Press, Las Palmas, pp 2953–2958

Research on UAV Cluster's Operation Strategy Based on Reinforcement Learning Approach

Yi Mao(✉) and Yuxin Hu

State Key Laboratory of Air Traffic Management System and Technology,
Nanjing 210007, China
mao_y@nuaa.edu.cn

Abstract. It is of necessity to formulate an overall UAV regulation scheme that covers each UAV's path and task implementation in the operation of UAV cluster. However, failure to fully realize consistency with pre-planned scheme in actual task implementation may occur considering changes of task, damage, addition or reduction of UAVs, fuel loss, unknown circumstance and other uncertainties, which thus entail a simultaneous online regulation scheme. By predicting UAV's 4D track, posture, task and demand of resources in regulation and clarifying data set in UAV cluster operation, quick planning of UAV's flight path and operation can be realized, thus reducing probability of scheme adjustment and improving operation efficiency.

Keywords: UAV · Operation strategy · Reinforcement learning approach

1 Introduction

UAV cluster operation is a complex multitask system that is affected and restricted by UAV's functional performance, load, information context, support equipment and other factors. Dynamic regulation of UAV cluster operation involves strategic training and optimization at the stage of gathering in takeoff, formation in flight, transformation of flight form, dispersion and collection of multiple UAVs of different types and structures [1–5].

Traditional training approaches, in general, include linear planning, dynamic planning, branch-and-bound method, elimination method and other conventional training optimization approaches frequently adopted in operational researches. In the application of optimal approach, the problem is often simplified for the sake of mathematical description and modeling so as to formulate an optimized regulation scheme [6–9]. UAV's specification, load, ammunition, coverage of support resources and battlefield context are all critical factors that impose influences in UAV cluster operation. Coupled with randomness and dynamicity in operation and a number of re-regulation tasks, UAV cluster operation is a NP-hard problem, characterized with great difficulties in solution, long time spent on solution and inability to realize simultaneous online UAV strategic cluster training [10–13].

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 28–35, 2020.
https://doi.org/10.1007/978-981-15-0187-6_4

The paper focuses on how to realize interactive learning, obtain knowledge and improve operation strategies in cluster operation assessment system so as to adapt to the environment and reach the ideal purpose [14–17]. UAV-borne computer is not informed of subsequent act, and it can only make judgment by trying every movement. Remarkably featured by trial-and-error-based search and delayed reward, reinforcement learning realizes enhancement of optimal actions and optimal movement strategies by making judgment on the environment's feedback about actions and instructing subsequent actions based on assessment. In this way, UAV cluster operation can better adapt to the environment [18].

2 UAV Cluster's Task Allocation and Modeling

Task allocation in UAV cluster operation aims at confirming UAV cluster's target and attack task, designing the path and realizing maximal overall performances and least cost in cluster attack. This is a multi-target optimization problem with numerous restrictions. Optimization indicators include: maximal value return of target, maximal coverage of target, minimal flight distance and least energy consumption. Restrictions include a certain definition ratio of target surveillance, necessity of target's location within UAV platform's attack diameter, limitations of each prohibition/flight avoidance zone and a certain number of UAV platforms that surveil the same target (no exceeding the limit), etc. [19].

Considering the features of multiple goals and restrictions, based on multi-target optimization theory, the paper sets up an overall multi-target integral planning model concerning UAV cluster's automatic task allocation.

Set the number of UAV platforms and targets as N_{v} and N_{T}, respectively. Decision-making variable in design is: $x_{i,j} \in \{0,1\}, i = \{1,2,\ldots,N_{\mathrm{T}}\}$. 1 means that UAV platform i targets at j, and it is the opposite in case of 0. Therefore, mathematical model of task allocation in UAV cluster attack is established.

Target function

(1) Cluster's least flight time f_1

One important indicator in UAV cluster's attack task allocation is "least time of UAV cluster's task implementation as much as possible," i.e., realizing shortest path of UAV platform allocated with a task.

$$P_{\mathrm{Fix}M_i} = \begin{cases} \mathrm{e}^{\frac{-R}{R_h}}, & R \leq R_h \\ 0, & R > R_h \end{cases} \tag{2.1}$$

Thereinto, M_i is the total number of targets allocated to UAV platform i.

(2) Cluster's total flight time f_2

Another important indicator in UAV platform's allocation of collaborated attack is "least cost as much as possible," in which energy consumption is the key factor. Energy consumed in flight is related to flight distance and time. The less the time is, the less the energy is consumed. Thus, indicator function is about the shortest flight time of UAV cluster.

$$\min f_2 = \sum_{i=1}^{N_v} \sum_{k=1}^{M_i} T_k \tag{2.2}$$

In order to ensure accuracy and feasibility of planned result when calculating the quantity of fuel burning/battery energy, initial path planning is indispensable, so is the calculation of energy consumption along initially planned path.

(3) Maximization of target's value f_3

Maximization of target's value is also an important indicator in task allocation on UAV platform. Under the condition of adequate attacking force, it is critical to realize maximization of target's values. The indicator function is as below:

$$\max f_3 = \sum_{i=1}^{N_v} \sum_{k=1}^{M_i} V_{\mathrm{k}} \tag{2.3}$$

Thereinto, V_{k} is the value of target K.

(4) Maximal target coverage f_4

Concerning task allocation on UAV platform, in addition to maximization of target's value, maximization of target coverage is also of great importance. Indicator function is as below:

$$\max f_4 = \frac{\sum_{i=1}^{N_v} M_i}{N_t} \tag{2.4}$$

In fact, UAV cluster's attack task allocation is about multi-target integral planning. With regard to multi-target planning, each target's relative weight can be confirmed according to decision-making intention, and multi-target integral planning can be transited to single-target planning:

$$\min f = \gamma_1(\alpha_1 f_1 + \alpha_2 f_2) + \gamma_2(\beta_1(1 - f_3) + \beta_2(1 - f_4)) \tag{2.5}$$

Thereinto, $0 \le \gamma_1, \gamma_2 \le 1, \gamma_1 + \gamma_2 = 1$

$$0 \le \alpha_1, \alpha_2 \le 1, \alpha_1 + \alpha_2 = 1$$
$$0 \le \beta_1, \beta_2 \le 1, \beta_1 + \beta_2 = 1$$

f_1, f_2 and f_3 are all normalized. Different valuations of α_1, α_2, β_1, β_2, γ_1 and γ_2 reflect preference of decision-making. Adjustment can be conducted automatically in accordance with the algorithm. For example, when attacking force is adequate, in general, β_1 = 0, β_2 = 1; while when attacking force is inadequate, $\beta_1 = 0.5$, $\beta_2 = 0.5$.

Restrictions

(1) Number of UAV platforms

The restriction imposes limitations on the number of attacking UAVs on UAV platform. The number should not exceed the total number of UAVs to be allocated with tasks on the platform.

$$\sum_{j=1}^{N_1} x_{i,j} \leq N_v \tag{2.6}$$

(2) Attack diameter

When cluster on UAV platform executes attack tasks, diameter should be the primary restriction to be taken into consideration, i.e., flight diameter on UAV platform must be within the attack diameter.

$$x_{i,j} l_{i,j_1} + \sum_{k=2}^{M_i} x_{i,j_k} l_{j_{k-1} j_k}, j_k \leq D_i \tag{2.7}$$

$l_{j_{k-1} j_k}$ refers to the distance between UAV platform and the first allocated target. $l_{j_{k-1} j_k}$, j_k refers to the distance between successive two targets allocated by UAV platform. Di refers to UAV platform's attack diameter $j = \{1, \ldots, M_i\}$.

Attack height

When cluster on UAV platform executes attack height, height should be the primary restriction to be taken into consideration, i.e., flight height on UAV platform must be within the attack height.

$$x_{i,j} H_j \leq H_i \tag{2.8}$$

H_j refers to the height of target j; H_i refers to the ascending limit of UAV platform i.

(4) Attack force

Target allocated to each UAV platform shall not exceed its attack force.

$$\sum_{k=1}^{M_i} x_{i,j_k} \leq \mathrm{Attack}_i \tag{2.9}$$

Attack_i is each UAV platform's attack load, i.e., the maximal number of tasks to be executed.

3 Optimization of Operation Strategic Training Modeling

One key assumption is that UAV cluster's operation strategy is based on the interaction between UAV and environment. Such an interaction can be regarded as MDP (Markov decision process). Optimization of operation strategy based on reinforcement learning can adopt solutions to Markov problems. Supposing that the system is observed at time $t = t_{1,}t_{2,}\ldots, t_n$, one operation strategy's decision process is composed of five elements:

$$\left\langle S, A(S), P_{ss}^{a}, R_{ss}^{a}, V|S, S' \in S, \alpha \in A(S) \right\rangle \tag{3.1}$$

Each element's meaning is as below:

① S is the non-empty set composed of all possible states of the system. Sometimes, it is also called "system's state space," which can be a definite, denumerable or arbitrary non-empty set. S, S' are elements of S, implying the states:

② $s \in S$, $A(s)$ is the set of all possible movements at states.

③ When system is at the states at decision-making time t, after executing decision a, the probability of system's S' state at the next decision-making point t + 1 is P_{ss}^{a}. The transition probability of all movements constitutes one transition matrix cluster:

$$P_{ss}^{a} = \Pr\{S_{t+1} = s' | S_t = S, a_t = a\} \tag{3.2}$$

④ When system is at the states at decision-making time t, after executing decision a, the immediate return obtained by the system is R. This is usually called "return function":

$$R_{ss}^{a\prime} = E\{r_{t+1} | S_t = S, a_t = a, s_{t+1} = s'\} \tag{3.3}$$

⑤ V is criterion function (or objective function). Common criterion functions include: expected total return during a limited period, expected discounted total return and average return. It can be a state value function or state-movement function.

Based on the characteristics of UAV cluster's operation, UAV, battlefield's environment, load platform, information resources and other supporting resources are segmented in the establishment of Markov state transition model (MDP). Thereinto, MDP model's state variables include usability of platform and load, position of UAV, priority of multi-tasks and residual quantity of fuel. Usability refers to whether UAV and load are usable or not, and residual usability, whether takeoff/recycling equipment is usable or not. Variables of position state include current position, position of takeoff/recycling equipment and position of 4D track of task execution in sky. Priority is used to regulate UAVs' takeoff and landing sequence as well as priority of multiple tasks. Residual quantity of fuel can be classified into multiple levels to judge the order of UAVs' landing and priority of tasks. Finally, MDP model's state space can be obtained:

$$S = \prod_{i \epsilon A} B_i \times \sum_{j \epsilon \text{pos}} \text{PR}_m \prod_{i \epsilon A} B_i \times \sum_{n \epsilon \text{level}} \text{LE}_n \tag{3.4}$$

Thereinto, B refers to equipment's usability; A is the set of equipment; PS is UAV's position state; pos is the set of position states; PR is the priority of tasks; pri is the set of priority tasks; LE is the level of residual fuel; level is the set of levels of residual fuel.

By setting up a return function, as convergence conditions in learning, an optimized regulatory strategy can be generated by Q learning approach. The change of states corresponding to regulatory strategies is UAV regulation plan.

4 Research on UAV Cluster's Operation Strategy Algorithm

The purpose of reinforcement algorithm is to find out one strategy π so that the value of every state $V^{\pi}(S)$ or $(Q^{\pi}(S))$ can be maximized, that is:

Find out one strategy $\pi{:}S \rightarrow A$ so as to maximize every state's value:

$$V^{\pi}(S) = E\{r_1 + \gamma r_2 + \ldots + \gamma^{i-1} r_i + \cdots | s_0 = s\} \tag{4.1}$$

$$Q^{\pi}(s,a) = E\{r_1 + \gamma r_2 + \ldots + \gamma^{i-1} r_i + \cdots | s_0 = s, a_0 = a\} \tag{4.2}$$

$$v^*(s) = \max_{\pi}(v^{\pi}(s)) \tag{4.3}$$

$$Q^*(s,a) = \max_{\pi}(Q^{\pi}(s,a)) \tag{4.4}$$

Thereinto, means immediate reward at time t. $\gamma(\gamma \epsilon [0.1])$ is attenuation coefficient. $V^{\pi}(S)$ and $Q^{\pi}(s,a)$ are optimal value functions. Corresponding optimal strategy is:

In reinforcement learning, if optimal value function $Q^*(V^*)$ has been estimated, there are three movement patterns for option: greedy strategy, ε-greedy strategy and softmax strategy. In greedy strategy, movements with highest Q value are always selected, i.e., $\pi^* = \arg\max Q^*(s,a)$. In ε-greedy strategy, movements with highest Q value are selected under a majority of conditions, and random selections of movements occur from time to time so as to search for the optimal value. In softmax strategy, movements are selected according to weight of each movement's Q value, which is usually realized by Boltzmann machine. In the approach, the movements with higher Q value correspondingly have higher weight. Thus, it is more likely to be selected.

The mechanism of all reinforcement learning algorithms is based on the interaction between value function and strategy. Value function can be made use of to improve strategy; value function learning can be realized and value function can be improved by making use of assessment of strategy. In such an interaction in reinforcement learning, UAV cluster's operation strategy gradually concludes optimal value function and optimal strategy.

5 Conclusion

At present, there are mainly two types of reinforcement learning algorithms to solve UAV cluster's independent intelligent operation problems: first, value function estimation, which is adopted in reinforcement learning researches mostly extensively with quickest development; second, direct strategy space search approach, such as genetic algorithm, genetic program design, simulated annealing and other evolution approaches.

Direct strategy space search approach and value function estimation approach can be both used to solve reinforcement learning problems. Both improve Agent strategy by means of training learning. However, direct strategy space search approach does not use strategy as a mapping function from state to movement. Value function is not taken into consideration. In another word, environment status is not taken into consideration.

Value function estimation approach concentrates on value function; that is, environment status is regarded as a core element. Value function estimation approach can realize learning based on the interaction between Agent and environment, regardless of quality of strategy—either good or bad. On the contrary, direct strategy space search approach fails to realize such a segmented gradual learning. It is effective for reinforcement learning with enough small strategy space or sound structure that facilitates easiness to find out an optimal strategy. In addition, when Agent fails to precisely perceive environment status, direct strategy space search approach displays great advantages. By modeling UAV cluster's independent intelligent operation, we can find that value function estimation is better for large-scale complex problems because it can make better use of effective computing resources and reach the goal of solving UAV cluster's independent intelligent operation.

References

1. Raivio T (2001) Capture set computation of an optimally guided missile [J]. J Guid Control Dyn 24(6):1167–1175
2. Fei A (2017) Analysis on related issues about resilient command and control system design. Command Inf Syst Technol 8(2):1–4
3. Smart WD (2004) Explicit manifold representations for value-function approximation in reinforcement learning. In: AMAI
4. Keller PW, Mannor S, Precup D (2006) Automatic basis function construction for approximate dynamic programming and reinforcement learning. In: Proceedings of the 23rd international conference on Machine learning. ACM, pp 449–456
5. Dearden R, Friedman N, Russell S (1998) Bayesian Q-learning. In: AAAI/IAAI, pp 761–768
6. Chase HW, Kumar P, Eickhoff SB et al (2015) Reinforcement learning models and their neural correlates: an activation likelihood estimation meta-analysis. Cogn Affect Behav Neurosci 15(2):435–459
7. Botvinick M (2012) Hierarchical reinforcement learning and decision making. Curr Opin Neurobiol 22(6):956–962
8. Smith RE, Dike BA, Mehra RK et al (2000) Classifier systems in combat: two-sided learning of maneuvers for advanced fighter aircraft. Comput Methods Appl Mech Eng 186(2):421–437

9. Salvadori F, Gehrke CS, Oliveira AC, Campos M, Sausen PS (2013) Smart grid infrastructure using a hybrid network architecture. IEEE Trans Smart Grid 3(4):1630–1639
10. Peng CH, Qian K, Wang CY (2015) Design and application of a VOC-monitoring system based on a Zigbee wireless sensor network. IEEE Sens J 15(4):2255–2268
11. Sara GS, Sridharan D (2014) Routing in mobile wireless sensor network: a survey. Telecommun Syst 57(1):51–79
12. Peters M, Zelewski S (2008) Pitfalls in the application of analytic hierarchy process to performance measurement. Manage Decis 46(7):1039–1051
13. Potts AW, Kelton FW (2011) The need for dynamic airspace management in coalition operations. Int C2 J 5(3):1–9
14. Yunru LI (2017) Joint tactical information system and technology development. Command Inf Syst Technol 8(1):9–14
15. Hoffman R, Jakobovits R, Lewis T et al (2005) Resource allocation principles for airspace flow control. In: Proceedings of AIAA guidance, navigation & control conference
16. Mukherjee A, Grabbe S, Sridhar B (2009) Arrival flight scheduling through departure delays and reroutes. Air Traffic Control Q 17(3):223–244
17. McLain TW, Beard RW (2005) Coordination variables, coordination functions, and cooperative timing missions. J Guid Control Dyn 28(1):150–161
18. Lau SY, Naeem W (2015) Cooperative tensegrity based formation control algorithm for a multi-aircraft system. In: American control conference, pp 750–756
19. Shi L, Liu W (2017) UAV management and control technology based on satellite navigation spoofing jamming. Command Inf Syst Technol 8(1):22–26

Analysis and Experimental Research on Data Characteristics of BDS Positioning Error

He Li(✉), Yi Mao, Yongjie Yan, and Xiaozhu Shi

The 28th Research Institute of China Electronics Technology Group Corporation,
Nanjing 210007, China
{lihesky1988,yyj_cet28,xidianshi1988}@163.com,
mao_y@nuaa.edu.cn

Abstract. Aiming at the problem that the integrity evaluation of satellite navigation and its augmentation system needs to analyze the characteristics of positioning error data, this paper proposes to use the time series analysis method to analyze the characteristics of the positioning error data of BDS single-point positioning and differential positioning from multiple angles of self-correlation, extremity and thick tail, which can provide the basis for deducing the integrity risk value by establishing an accurate positioning error distribution model. The result shows that the self-correlation of BDS differential positioning error is significantly lower than that of single-point positioning error. And the error of the two positioning methods in vertical and horizontal directions shows the characteristic of thick tail. Compared with the normal distribution, the single-point vertical positioning error has the most serious thick tail, reaching 420.3% and the differential horizontal positioning error has the lightest thick tail, only 22.9%. This indicates that the differential positioning method has better positioning performance and integrity.

Keywords: Satellite navigation · Positioning error · Data characteristic · Integrity evaluation

1 Introduction

With the sustained growth of global air transport business, civil aviation requires higher and higher positioning performance and integrity of satellite navigation system. Due to the influence of satellite clock error, ephemeris error, ionospheric delay, tropospheric delay and multipath effect, the positioning error of satellite navigation system may exceed the threshold set by International Civil Aviation Organization (ICAO). And the probability that the system can not detect the transfinite events which lead to the misleading information of the aircraft operation is called integrity risk [1]. Nowadays, with the rapid development of BeiDou Satellite Navigation System (BDS) and its augmentation system, the integrity evaluation of the system has been paid more and more attention. It is an effective evaluation method by establishing the distribution model of positioning error to derive integrity risk value. The data characteristics of positioning error are the important basis for choosing a reasonable distribution model. At present, most domestic literatures focus on the temporal and spatial characteristics of

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 36–45, 2020.
https://doi.org/10.1007/978-981-15-0187-6_5

BDS positioning error in the whole country. Few articles analyze the characteristics of BDS positioning error from the positioning domain. Paper [2] uses the continuous observation data of the domestic regional monitoring network to show that the positioning error is relatively stable in time scale and uneven in space scale. Paper [3] analyzes the main factors affecting the dynamic positioning error of BDS and evaluates the positioning performance of the system. The research methods of the above literature have reference significance for analyzing the data characteristics of BDS positioning error.

In this paper, the static single-point positioning error and differential positioning error of BDS are taken as the sample data. And the time series analysis method is used to study the positioning error characteristics of the two positioning methods from the angles of self-correlation, extremity and thick tail. The thick tail of positioning error is qualitatively and quantitatively verified based on the distribution statistics and kurtosis coefficients, respectively. The analysis results can provide a basis to establish an accurate error distribution model in the integrity evaluation of BDS.

2 The Error Sources of BDS Positioning

The origin of BDS positioning error can be divided into three sources as shown in Fig. 1.

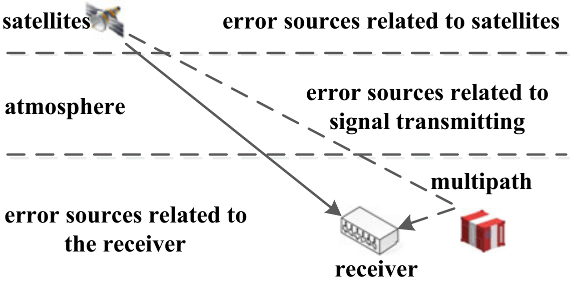


Fig. 1. Error sources of BDS positioning

(1) Error related to the satellites: These errors mainly include satellite clock error and ephemeris error which are caused by the inability of ground monitoring station to make absolutely accurate measurements and predictions of satellite orbit and frequency drift of the satellite clock [4, 5]. The ephemeris error varies very slowly with time and has strong spatial and temporal correlation.

(2) Error related to the signal transmitting: Signal transmitting from the satellite to the receiver needs to pass through the atmosphere. The influence of atmosphere on signal transmitting is atmospheric delay which is usually divided into ionospheric delay and tropospheric delay with high spatial correlation [6].

(3) Error related to the receivers: Receivers may be subject to varying degrees of multipath effects and electromagnetic interference. This part of error also includes the noise of the receiver itself. Multipath error does not show normal distribution but sinusoidal motion with a period of several minutes as the satellite moves [7].

The above error can be classified into deviation and noise according to the characteristics of magnitude and speed of changing. The deviation varies slowly in a certain period of time. For example, the magnitude of ionospheric delay generally remains unchanged in a few seconds or even minutes. The noise changes quickly and is difficult to measure. But its mean, variance, correlation function and other statistical indicators in a certain period of time can be calculated.

3 Characteristic Analysis and Experimental Study of BDS Positioning Error

The data characteristics of positioning error are the guarantee of choosing a reasonable distribution model and improving the accuracy of BDS integrity risk evaluation. BDS has two positioning methods: Single-point positioning and differential positioning. The error of single-point positioning can reach to tens of meters without any compensation which is difficult to meet the positioning performance requirements of many occasions. Differential positioning uses the spatial and temporal correlation of satellite clock error, ephemeris error and atmospheric delay error to eliminate the measurement error of user stations and improve the positioning performance by calculating the error correction of reference stations with known positions. Formulas (1) and (2) are pseudo-range observer expressions $\rho_u^{(i)}$ and $\rho_{uc}^{(i)}$ for single-point positioning and differential positioning, respectively.

$$\rho_u^{(i)} = R_u^{(i)} + c(\Delta t_u - t) + g_u^{(i)} + I_u^{(i)} + T_u^{(i)} + \varepsilon_u^{(i)} \tag{1}$$

$$\rho_{uc}^{(i)} = R_u^{(i)} + c\Delta t_{ur} + g_{ur}^{(i)} + I_{ur}^{(i)} + T_{ur}^{(i)} + \varepsilon_{ur}^{(i)} \tag{2}$$

Among them, superscript i represents satellite number, subscript u, r represent user receiver and reference receiver, respectively. $R_u^{(i)}$ is the real distance from satellite i to the user receiver. Δt_u, t, $g_u^{(i)}$, $I_u^{(i)}$, $T_u^{(i)}$, $\varepsilon_u^{(i)}$ represent clock error of the user receiver, satellite clock error, ephemeris error, ionospheric error, tropospheric error, noise, respectively, $\rho_{uc}^{(i)}$, $g_{ur}^{(i)}$, $I_{ur}^{(i)}$, $T_{ur}^{(i)}$, $\varepsilon_{ur}^{(i)}$ represent the corresponding correction residues and satellite clock error can be eliminated completely. $g_{ur}^{(i)}$, $I_{ur}^{(i)}$, $T_{ur}^{(i)}$ are nearly to zero under short baseline. c is the speed of light. Differential positioning could eliminate the amount of common error.

Because the ionospheric delay error, tropospheric delay error and multipath effect error show non-zero mean and asymmetric non-Gaussian characteristics, the superposition of the error makes the positioning error projected to the position domain shows non-Gaussian characteristics.

3.1 Experimental Environment and Positioning Data Processing

The purpose of the experiment is to collect the sample data of single-point positioning and differential positioning. The experimental equipment includes four sets of BDS reference receivers and antennas, one data processor, one set of very high-frequency data broadcast (VDB) transmitter and antenna, one set of BDS test receiver and antenna and one test computer. Some equipment is shown in Fig. 2.

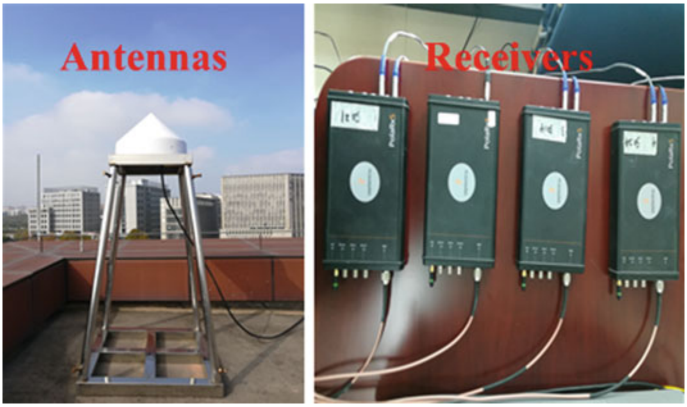


Fig. 2. Some experimental equipment

Four reference receiver antennas are erected at four calibrated corners on the top floor, respectively, and the test receiver antenna is placed at a calibrated measuring point. Data processor uses BDS data of four reference receivers to calculate differential correction. Test computer realizes single-point positioning and differential positioning by applying correction in positioning calculation. The schematic diagram of the whole experimental device is shown in Fig. 3. The actual position coordinates of the reference receiver antennas and the test receiver antenna are shown in Table 1.

The calculating period of the test computer is 1 s and 20,000 sample data expressed in WGS-84 rectangular coordinate system are collected according to the two methods of single-point positioning and differential positioning, respectively. Positioning error $(\Delta x, \Delta y, \Delta z)$ which includes the horizontal positioning error Δd and the height positioning error Δh is the difference between the calculated value and the true value of the measured point. The positioning error vector in station-origin coordinate system $(\Delta e, \Delta n, \Delta u)$ can be obtained by using the transformation Formula (3):

$$\begin{bmatrix} \Delta e \\ \Delta n \\ \Delta u \end{bmatrix} = \begin{bmatrix} -\sin\lambda & \cos\lambda & 0 \\ \sin\phi\cos\lambda & -\sin\phi\sin\lambda & \cos\phi \\ \cos\phi\cos\lambda & \cos\phi\sin\lambda & \sin\phi \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{3}$$

$\Delta e, \Delta n, \Delta u$ are eastward error, northward error, upward error, respectively. λ, ϕ is the longitude and latitude of the measured point. Thus, we can deduce the horizontal

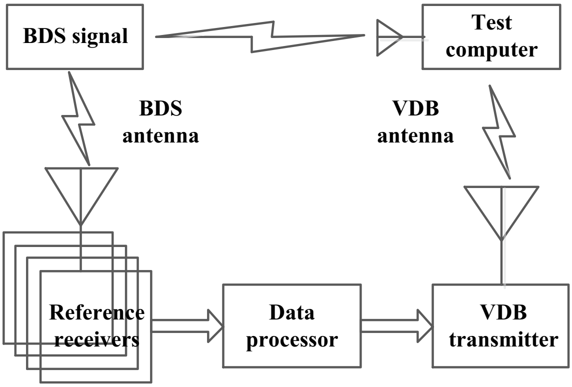


Fig. 3. Schematic diagram of the whole experimental device

Table 1. Position coordinates of the reference receiver antennas and test receiver antenna

Coordinates	*x* (m)	*y* (m)	*z* (m)
Reference point 1	−2,613,360.05	4,740,275.02	3,362,287.98
Reference point 2	−2,613,353.52	4,740,285.43	3,362,275.35
Reference point 3	−2,613,327.20	4,740,297.81	3,362,282.69
Reference point 4	−2,613,329.96	4,740,288.58	3,362,300.42
Measured point	−2,613,328.61	4,740,296.76	3,362,284.98

positioning error (Formula 4) and the height positioning error (Formula 5) from the positioning error vector:

$$\Delta d = \sqrt{\Delta e^2 + \Delta n^2} \tag{4}$$

$$\Delta h = \Delta u \tag{5}$$

Due to the space limit of the paper, only the vertical positioning error of the two positioning methods is shown in Figs. 4 and 5.

3.2 The Self-correlation of Positioning Error

Because there are some correlations among the slow variable error such as ionospheric delay error, tropospheric delay error and multipath effect error included in the source of positioning error, there will be some self-correlation in the positioning error. Before modeling the positioning error distribution in BDS integrity evaluation, it is necessary to determine that the error samples approximately meet the requirements of independent and identical distribution. The positioning error of satellite navigation system can be regarded as a set of time series. The stationarity (approximate independence) of data can be judged by calculating the self-correlation function of the error sequence with

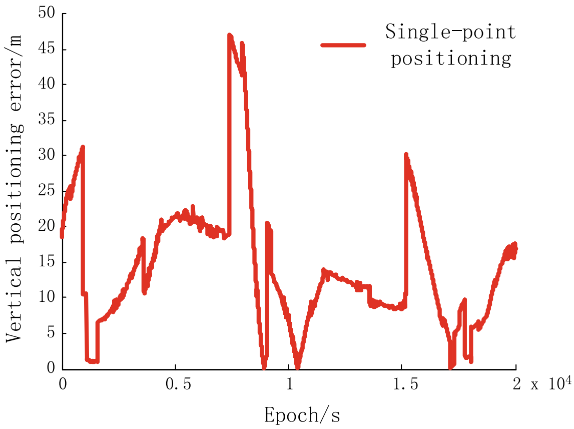


Fig. 4. Single-point positioning error sequence

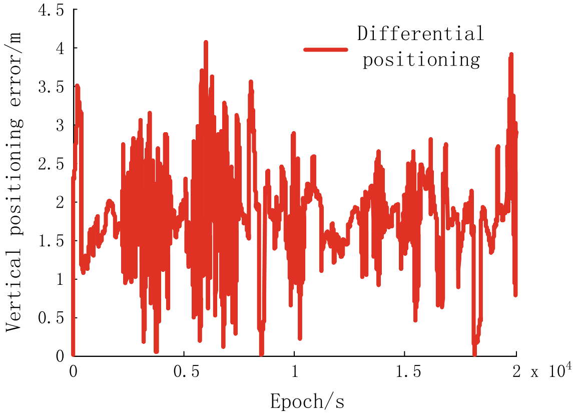


Fig. 5. Differential positioning error sequence

time series analysis method. Formula (6) is used to calculate the self-correlation function.

$$\rho_k = \frac{\sum_{i=1}^{n-k} (x_i - \overline{x})(x_{i+k} - \overline{x})}{\sum_{i=1}^{n} (x_i - \overline{x})^2} \tag{6}$$

Among them, k is the order of delay and n is the number of samples. If the value of self-correlation function is 1 at zeros and close to 0 at other positions, it indicates that the sequence is likely to be stationary and the distribution model of error sequence can be established directly. If the value of self-correlation function is obviously not zero at non-zeros, it indicates that the sequence is very likely to be non-stationary. This requires increasing the sampling interval to reduce the correlation before establishing the distribution model. Since the characteristics of horizontal and vertical positioning errors are identical, this paper only calculates the self-correlation function of vertical positioning error of single-point positioning and differential positioning, as shown in Figs. 6 and 7.

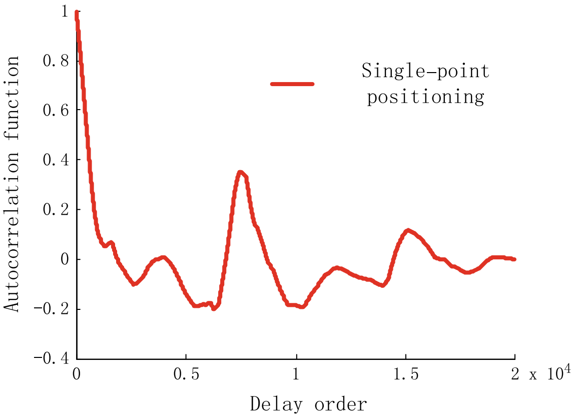


Fig. 6. Self-correlation function of single-point positioning error sequence

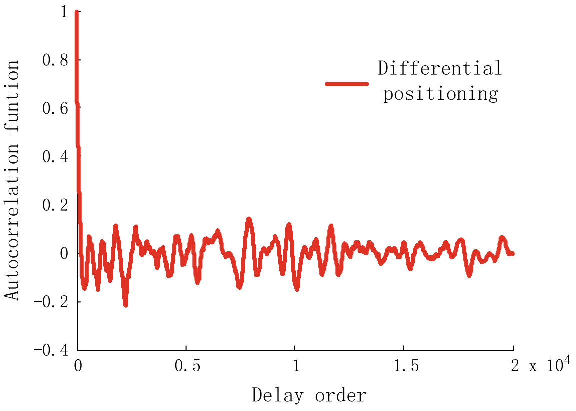


Fig. 7. Self-correlation function of differential positioning error sequence

It can be seen qualitatively from Figs. 6 and 7 that the self-correlation of the single-point positioning error sequence is high and the samples are not independent which makes it difficult to meet the requirements of modeling. However, the self-correlation of the differential positioning error sequence is low and the samples are nearly independent. So the distribution model can be directly established for integrity evaluation.

3.3 The Extremity of Positioning Error

According to the RTCA DO-253 document standard [8], the integrity risk value of satellite navigation ground-based augmentation system (GBAS) is 10^{-8}–10^{-7}. If the frequency estimation is used to calculate the integrity risk, the required sample number should be 10^8 at least. Even if the sampling frequency is 1 Hz, the acquisition time will reach several years. So we can see that the event of positioning error exceeding the limit is extreme and it is difficult to appear in short-term observation experiments. Therefore, the integrity evaluation of the system needs to deduce the integrity risk value by establishing the distribution model of positioning error.

3.4 The Thick Tail of Positioning Error

The integrity risk evaluation of satellite navigation system mainly focuses on the characteristics of the distribution tail of positioning error as the occurrence of larger error. Therefore, whether the positioning error has a thick tail is the key to choose the distribution model. Compared with the normal distribution, if the tail descent rate of a given distribution is slower than that of the normal distribution, it can be judged that the distribution is thick-tailed. The hist function of MATLAB software can be used to make intuitive statistics on the distribution of vertical positioning error in Sect. 3.1 and the result is shown in Figs. 8 and 9.

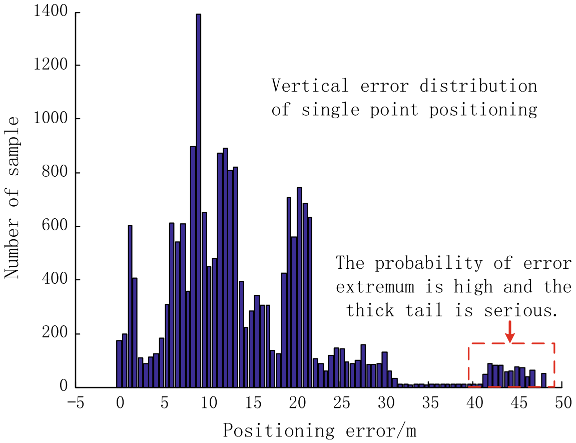


Fig. 8. Vertical error distribution of single-point positioning

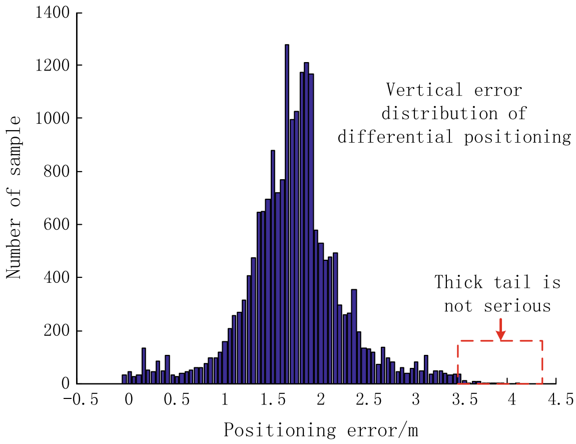


Fig. 9. Vertical error distribution of differential positioning

From Figs. 8 and 9, it can be qualitatively concluded that the thick tail of single-point positioning vertical error is serious and error exceeding event is easy to occur. While the tail of differential positioning vertical error is very light and the system has low integrity risk. Thick-tailed feature of the sample distribution can also be quantitatively judged by the kurtosis as shown in Formula (7):

$$\beta = \frac{\sum_{i-1}^{n} (x_i - \bar{x})^4}{(n-1)s^2} \tag{7}$$

β is the kurtosis of sample. x, n are the number of sample. $\bar{x}$, s is the mean and variance, respectively. Kurtosis is a dimensionless variable and linear transformation does not change the kurtosis value of the variable. Kurtosis is used to characterize the degree of dispersion of the variable. The thicker the tail distribution is, the larger the kurtosis is. The kurtosis of the normal distribution is 3. For a given variables, if the kurtosis is greater than 3, the distribution of the variable is thick-tailed. $(\beta - 3)/3$ can be used to characterize the degree of thick tail of the variable distribution relative to the normal distribution. The result of calculating the kurtosis and the degree of thick tail of the experimental data in Sect. 3.1 is shown in Table 2.

Table 2. Kurtosis of positioning error

Error type	Kurtosis	Thick tail or not	The degree of thick tail (%)
Single-point vertical error	13.6107	Yes	420.3
Single-point horizontal error	3.9181	Yes	30.6
Differential vertical error	4.8195	Yes	60.7
Differential horizontal error	3.6868	Yes	22.9

As shown in Table 2, the positioning error of the two positioning methods has the characteristics of thick tail. Compared with the normal distribution, the single-point vertical positioning error has the highest thick tail, reaching 420.3% and the differential horizontal positioning error has the lowest thick tail, only 22.9%. It indicates that the integrity risk of single-point positioning is higher than that of differential positioning.

4 Conclusion

Establishing an accurate positioning error distribution model is the key to BDS integrity evaluation and the characteristics of positioning error are the basis for selecting the distribution model. Based on the analysis of positioning error sources, this paper studies the self-correlation, extremity and thick tail of positioning error with the measured data of BDS single-point positioning and differential positioning by using time series analysis method. The experimental results show that the self-correlation of single-point positioning error is high and it is difficult to directly establish the evaluation model. And the distribution tail is heavy which is prone to error exceeding the limit and increases the integrity risk of the system. Because of eliminating most of the common error, differential positioning error has low self-correlation and is easy to model. And evaluate the integrity of the system, and the distribution tail is light, which reduces the integrity risk of the system.

References

1. Dan ZQ (2015) Research on integrity evaluation method of GBAS. Beijing Univ Aeronaut Astronaut 22–24
2. Che ZW, Chang ZQ, Fan MJ (2017) Performance analysis of Beidou navigation positioning service. Glob Positioning Syst 42(1):2–4
3. Zhang FZ, Liu RH, Ni YD (2018) The testing and analysis of BDS dynamic positioning precise. Glob Positioning Syst 43(1):43–45
4. Li CE (2018) Spatial-temporal variation of land desertification in Xinjiang. Sci Surv Map 43 (9):33–34
5. Yang X, Xu AG, Zhu HZ (2017) The precision analysis of BDS precise orbit and precise clock. Sci Surv Mapp 42(12):8–9
6. Hu H (2013) Research on theory and realization of GNSS precise point positioning. China Min Univ 28–52
7. Xie G (2012) GPS principle and receiver design. National Defense Industry Press
8. FAA Specification (2008) Category I local area augmentation system ground facility. RTCA DO-253C

Design of Elderly Fall Detection Based on XGBoost

Min Xiao[1](✉), Yuanjian Huang[1], Yuanheng Wang[1],
and Weidong Gao[2]

[1] School of Computer Science and Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China
xiaomincloud@163.com, {huangyj, jahracal}@bupt.edu.cn
[2] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China
gaoweidong@bupt.edu.cn

Abstract. Aging has become a serious problem facing the whole world. Falling is the leading cause of injury and death in the elderly. This paper proposes a fall detection algorithm based on machine learning XGBoost and full-field positioning. Using the data of gyroscope and acceleration sensor, we exploit the "full-field positioning" to increase the dimension of input data and propose a method "maximum satisfaction rate" to mark and train the threshold of data. The experimental results show that this design has obtained high accuracy on falling detection and perfect balance between sensitivity and specificity.

Keywords: Fall detection · XGBoost machine learning · Gyroscope · Full-field positioning

1 Introduction

The WHO has pointed out that at present world's aging trend is becoming more and more serious. Until 2017, the elderly aged 65 and over accounted for 8.696% of the world's population [1]. The physiology of the human body is degraded as we grow older. And coupled with the complexity of the living environment, the elderly are prone to accidents such as falls. Moreover, some elderly people are alone at home, and no one is around when they fall, which causes serious consequences. The 2018 World Health Organization showed that falls are the second leading cause of accidental or unintentional injury deaths worldwide. Besides, adults older than 65 years of age suffer the greatest number of fatal falls [2]. Thus, a family using automatic device is in urgent need to protect the elderly from falling injury.

In recent years, there have been many research results of fall detection in the elderly at home and abroad: wearable fall recognition alarm system, fall alarm system based on embedded vision, elderly fall detection based on STM32 system, fall detection belt based on accelerometer, wearable fall monitoring, and so on. All fall detection products can be divided into two categories: visual inspection-based methods and sensor-based methods.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 46–57, 2020.
https://doi.org/10.1007/978-981-15-0187-6_6

(1) Vision-based methods: such as “multi-camera video surveillance system” [3]. First of all, they have great limitations and cannot be popularized in every household, which can only be applied to specific occasions. Secondly, the visual product judges whether the fall is theoretically based on the appearance of the human body falling, and the detection accuracy is difficult to be improved in essence.
(2) Sensor-based methods have fewer features to judge first, which are generally based on acceleration or angular velocity, or both. Such as the “wearable sensors for reliable fall detection,” it just judges the falling by the threshold of acceleration [4]. And “elderly fall monitoring method and device” [5] pointed out that using acceleration and angle to monitor falling, and “fall detection analysis with wearable MEMS-based sensors” [6] has consistent ideology with our paper to some degree. However, it just proposed a method and the execution of method is unknown.

Based on the survey of the above products or papers: If we want to put such products into wide application, there are two main problems to be solved: The first is the high accuracy with the balance between sensitivity and specificity, and the second is household, which means simplicity and convenience. Considering that XGBoost has higher accuracy and lower false-positive rate than other algorithm, it is a kind of gradient boosting which has proven many times to be an effective prediction algorithm for both classification and regression task, such as crude oil price forecasting [7], DDoS attack detection [8], and so on. Thus, we decide to use XGBoost as the basis of detection.

2 Proposed Falling Detection Based on XGBoost

2.1 The Process of Establishing the Model

The falling process includes:

(1) The horizontal acceleration relative to the heaven and earth coordinate system increases: This phenomenon occurs because during the movement of the human body, the horizontal acceleration increases and the speed decreases due to being tripped.
(2) The *X*-axis angular velocity and *Y*-axis angular velocity relative to the heaven and earth coordinate system increase.
(3) The vertical acceleration relative to the heaven and earth coordinate system increases: This is caused by the interaction between human body and the ground.
(4) The static process after falling, during which the human body’s pitch angle will reduce (that is, the included angle between the human body and the horizontal ground becomes smaller).

Generally, unless the person faints after falling, it is difficult to detect this very weak stationary process. Process (2) is a necessary condition for the fall to occur; that is, it is considered to have fallen when the body is dumped. The occurrence of the process (3) substantively causes harm to the human body, and the alarming of damage is the purpose of detecting the fall. The data we collected using gyroscopes and accelerometers is shown in Fig. 1. Process (4) also has the same effect when “lying down,” so the angle is introduced as an auxiliary judgment condition.

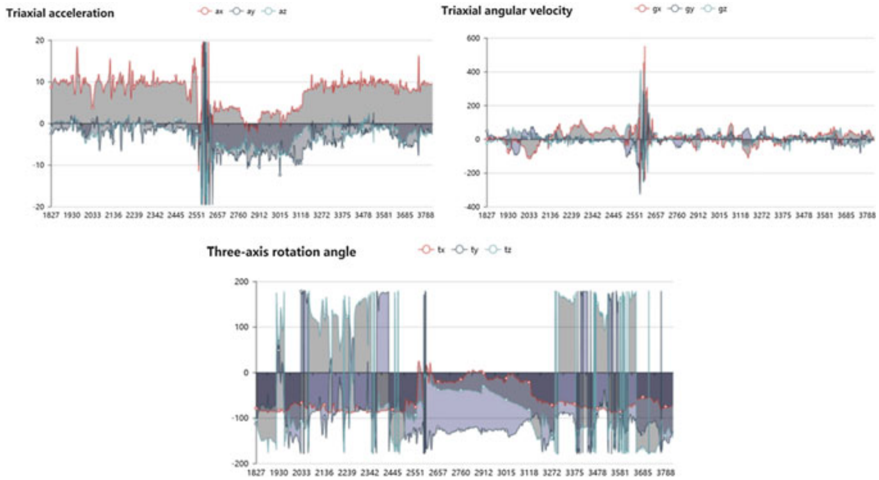


Fig. 1. Triaxial acceleration, triaxial angular velocity, triaxial rotation during the falling

The triaxial acceleration, triaxial angular velocity, triaxial rotation angle during the fall are shown in Fig. 1 (the abscissa unit is 10 ms, and the ordinate unit is m/s^2).

To detect whether a fall has occurred, the core is to distinguish between daily life behavior and fall behavior. Thus, we collected a large amount of data from different scenes in daily life, including running, squatting, going upstairs and downstairs, walking, lying down, and jumping.

Based on the various types of data collected above, we first draw some brief conclusions:

(1) The essence of falling damage to the human body is “force.” According to Newton’s first law, the greater the acceleration, the greater the external force the human body receives, and the greater the damage. Then, the combined acceleration “azz” is used as one of the judgment conditions for determining the occurrence of the falling.

$$\mathrm{azz} = \sqrt{a_x^2 + a_y^2 + a_y^2} \tag{1}$$

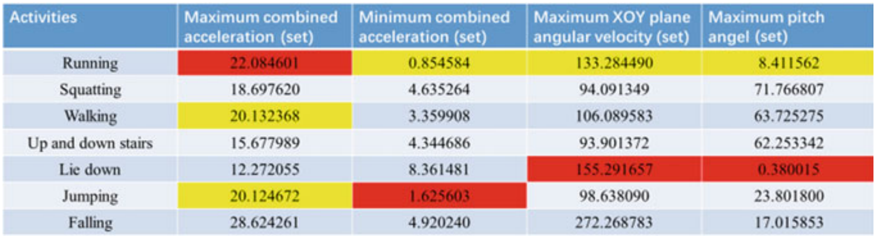
(2) The falling causes the human body to change from an “upright” state to a “squat” state. During this process, the angular velocity in the horizontal direction changes. Thus, the change of angular velocity is measured by the horizontal angular velocity “gyro.” The horizontal angular velocity is [4]:

$$\text{gyro} = \sqrt{g_x^2 + g_y^2} \tag{2}$$

(3) During the process of collecting data, we found that in some cases, the occurrence of acceleration peaks has a certain degree delay compared to the angular velocity peaks. The result of the analysis is the same: When the human body falls, the upper body is first dumped. In this process, the angular velocity reaches a peak at a certain moment; when the human body touches the ground, the acceleration reaches a peak. In addition, the acceleration can also reach an acceleration close to the fall when running and jumping; thus, we consider the importance of angular velocity > the importance of acceleration.

(4) According to the data analysis (Table 1), in order to distinguish the acceleration and angular velocity between falling and daily activities, the angle is introduced to judge the falling. Due to the "90° Euler angle limitation" and the complicated triangulation in real time, we choose to use the quaternion method to calculate the angle. Quaternion: An object can reach any attitude by rotating around an axis to a certain angle. In the process of calculating the angle, the "full-field positioning" algorithm is used combined with the real-time acceleration correction on angular velocity: According to the previous gravity unit vector and the gravity unit vector obtained by this measurement, the cross-product of two is obtained for correcting angular velocity. Then, quaternion (q_0, q_1, q_2, q_3) is updated according to "Runge–Kutta of the first order," so that the triaxial rotation angles "t_x", "t_y", and "t_z" relative to the heaven and earth coordinate system are [9, 10]:

Table 1. Three-dimensional data in different scenes



Activities	Maximum combined acceleration (set)	Minimum combined acceleration (set)	Maximum XOY plane angular velocity (set)	Maximum pitch angel (set)
Running	22.084601	0.854584	133.284490	8.411562
Squatting	18.697620	4.635264	94.091349	71.766807
Walking	20.132368	3.359908	106.089583	63.725275
Up and down stairs	15.677989	4.344686	93.901372	62.253342
Lie down	12.272055	8.361481	155.291657	0.380015
Jumping	20.124672	1.625603	98.638090	23.801800
Falling	28.624261	4.920240	272.268783	17.015853

$$t_x = a\sin(-2 * q_1 * q_3 + 2 * q_0 * q_2) * 57.3 \tag{3}$$

$$t_y = a\tan 2(2 * q_2 * q_3 + 2 * q_0 * q_1, -2 * q_1 * q_1 - 2 * q_2 * q_2 + 1) * 57.3 \tag{4}$$

$$tz = a\tan 2(2 * q_1 q_2 + 2 * q_0 q_3, -2 * q_2 q_2 - 2 * q_3 q_3 + 1) * 57.3 \tag{5}$$

(5) In the same way as (2), we hope the angle changes horizontally; thus, we use the human pitch angle to judge the falling. The *X*-axis rotation angle is "t_x", and the *Y*-axis rotation angle is "t_y", whereby the XOY plane rotation angle "agz" is:

$$\mathrm{agz} = \arctan\left(\frac{1}{\sqrt{\tan^2 t_x + \tan^2 t_y}}\right) \tag{6}$$

2.2 Notation

Notation	Meaning
a_x	*X*-axis acceleration
a_y	*Y*-axis acceleration
a_z	*Z*-axis acceleration
g_x	*X*-axis angular velocity
g_y	*Y*-axis angular velocity
g_z	*Y*-axis angular velocity
gyro	XOY plane angular velocity
t_x	*X*-axis rotation angle
t_y	*Y*-axis rotation angle
t_z	*Z*-axis rotation angle
agz	Pitch angle
Win	Detection window size
T	Sampling frequency
$\mathrm{Meet}_{\mathrm{azz}}[t]$	The number of "azz" meets the threshold in [*t*-300, *t*]
$\mathrm{Meet}_{\mathrm{gyro}}[t]$	The number of "gyro" meets the threshold in [*t*-300, *t*]
$\mathrm{Meet}_{\mathrm{agz}}[t]$	The number of "agz" meets the threshold in [*t*-300, *t*]
BORDER_AZZ	The threshold of "meet(azz)" in window
BORDER_GYRO	The threshold of "meet(gyro)" in window
BORDER_AGZ	The threshold of "meet(agz)" in window
q_0, q_1, q_2, q_3	Real-time quaternion

2.3 Falling Detection Model's Establishment

(1) "Sliding window": This paper proposes the "sliding window processing data" method to store updated data. The data storage includes whether the updated data satisfies the condition of the fall and the overall data of the window after updating the data. The single data is judged according to the XGBoost training model. We set the data judgment window size to "Win"; that is, we use the "Win" group data collected from a certain moment to present to determine whether a fall has occurred at present. Use a variable "forsample" to save the location of the update at present, and slide to the next position after the update. At the same time, an array of "Result_all [5]" is used to save the number of results which is judged to be "True."

(2) "Two sets of falling detection model": A simple analysis of the data in "2. Model Establishment Process" shows that the angular velocity is the physical quantity that best describes the falling and the most accurate judgment of falling. Therefore, we believe that angular velocity reaching the threshold is first when falling occurs.

During the actual test, it was found that the acceleration of the sliding fall was greater than the acceleration of the forward fall. We theoretically analyze: Most of the subjects subconsciously supported the ground in the forward fall, which alleviates the impact. Thus, the acceleration was not big. Therefore, we put forward the idea of "grouping judgment": Under the premise of angular velocity characteristics, two sets of acceleration judgments are made: One is when the pitch angle reaches the threshold, and the acceleration reaches a relatively small threshold, it will be considered to have a side fall or a back fall; the other is when the acceleration reaches a relatively large threshold, it is considered to have a forward fall. And the "grouping judgment" optimizes the judgment model.

In addition, we can use two sets of falling detection models to balance sensitivity and specificity [11]. Specificity and sensitivity are defined as follows:

$$\text{Sens} = \frac{\text{True falling}}{\text{True falling } + \text{ False falling}} \tag{7}$$

$$\text{Spec} = \frac{\text{non-falling}}{\text{non-falling } + \text{ False non-falling}} \tag{8}$$

Assuming: For n sets of data, the actual falling data accounted for "*f*"; the accuracy of classifier 1 to falling and non-falling data is (p_{11}, p_{12}); the accuracy of classifier 2 to falling and non-falling data is (p_{21}, p_{22}); assume that under this model, the applying frequency weight of the two models is w_1, w_2.

When there only exists classification 1:

$$\text{Sens} = \frac{n \cdot f \cdot p_{11}}{n \cdot f \cdot p_{11} + n \cdot (1 - f) \cdot (1 - p_{12})} \tag{9}$$

$$\text{Spec} = \frac{n \cdot (1 - f) \cdot p_{12}}{n \cdot (1 - f) \cdot p_{12} + n \cdot f \cdot (1 - p_{11})} \tag{10}$$

After introducing classification 2, the overall specificity and sensitivity can be balanced by adjusting the accuracy of classification 2:

$$\text{Sens}' = \frac{n \cdot f \cdot \frac{w_1 \cdot p_{11} + w_2 \cdot p_{21}}{w_1 + w_2}}{n \cdot f \cdot \frac{w_1 \cdot p_{11} + w_2 \cdot p_{21}}{w_1 + w_2} + n \cdot (1 - f) \cdot \left(1 - \frac{w_1 \cdot p_{12} + w_2 \cdot p_{22}}{w_1 + w_2}\right)} \tag{11}$$

$$\text{Spec}' = \frac{n \cdot (1 - f) \cdot \frac{w_1 \cdot p_{12} + w_2 \cdot p_{22}}{w_1 + w_2}}{n \cdot f \cdot \frac{w_1 \cdot p_{12} + w_2 \cdot p_{22}}{w_1 + w_2} + n \cdot (1 - f) \cdot \left(1 - \frac{w_1 \cdot p_{11} + w_2 \cdot p_{21}}{w_1 + w_2}\right)} \tag{12}$$

① If $p_{21} > p_{11}$, that is comparing to classifier 1, classifier increases the accuracy for falling data. Then, the corresponding accuracy for non-falling data will be reduced: $p_{22} < p_{12}$. According to (11)(12), Sens' ↓ and Spec ↑.

② If $p_{21} > p_{11}$, that is comparing to classifier 1, classifier reduces the accuracy for falling data. Then, the corresponding accuracy for non-falling data will be increased: $p_{22} < p_{12}$. According to (11)(12), Sens' ↑ and Spec ↓.

2.4 Data Threshold Training Based on XGBoost

According to the data analysis results, and combined with the actual situation, we use 3 s data (setting T = 100 Hz, then Win = 300, the corresponding window data size is 300 groups) as a critical window to determine whether a person has fallen; that is, the feature value of the fall occurred within the last 3 s. Before training for XGBoost, we first need to manually mark our falling data. In order to make the training results more accurate, we make preset values for the threshold.

According to the data collected by daily behavior, the three-dimensional data in different scenarios is shown in Table 2.

Table 2. Three-dimensional data preset

The angular velocity threshold of single group	gyro > 133.2844 deg/s
The acceleration threshold of single group	azz < 3.35 m/s^2 or azz > 12.2720 m/s^2
The pitch angle threshold of single group	agz < 38.3898°

The data characteristics of the normal fall are: The combined acceleration increases, the XOY plane angle velocity increases, and the human pitch angle becomes smaller. As shown above, the red color marked data is the closest to the fall feature, and the yellow color marked data is relatively closer to the fall feature. It can be observed that the running data characteristics are the closest to falling. We expected that it will be better for threshold to differentiate the falling and daily activities. At the same time, considering the sensitivity of the fall detection comprehensively, we preset the three-dimensional data threshold as shown in Table 2.

In this project, the difficulty of data training is how to choose the feature area where the fall occurs. Therefore, we propose a "maximum satisfaction rate" method: The number of single-group data satisfying the above threshold conditions within a certain 3 s: azz < AC2 OR azz > AC1, gyro > AG, agz < AN, is represented as meet_azz[t], meet_g[t], meet_agz[t]; then:

$$\text{meet}_{\text{azz}}[t] = \sum_{t-300}^{t} \text{int } (\text{azz}[t]\langle AC2\text{azz}[t]\rangle AC1) \tag{13}$$

$$\text{meet}_{\text{gyro}}[t] = \sum_{t-300}^{t} \text{int}(\text{gyro}[t] > AG) \tag{14}$$

$$\mathrm{meet}_{\mathrm{agz}}[t] = \sum_{t-300}^{t} \mathrm{int}(\mathrm{agz}[t] < AN) \tag{15}$$

For a fall process, we think that when meet_azz[t], meet_gyro[t], and meet_agz[t] all reach the maximum value, the presets are adjusted according to the size of each "satisfaction rate" meet[t] in different scenarios. At last, the falling is considered to occur in a time interval [t-300, t]. Thus, among these 300 sets of data, only the data satisfying the above threshold condition is marked.

After the marking is completed, we put all three-dimensional data into XGBoost for training. At the same time, we introduce the following three concepts: accuracy, recall, and f1_score to measure the training effect. TP is the number of falling data (positive class), TN is the number of non-falling data (negative class), FP is the number of negative class divided into positive class, and FN is the number of positive class divided into negative class. Thus, we define [12]:

$$\mathrm{accuracy} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{16}$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{17}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{18}$$

$$\frac{2}{\mathrm{F1\_score}} = \frac{1}{\mathrm{Precision}} + \frac{1}{\mathrm{Recall}} \tag{19}$$

The 25,000 sets of training data contain 10 times falling marked data. Besides, it does not include daily activities' data, the reason is that for single-group data, the falling data accounts for a little part, and the daily activities' data value is usually close to that of the falling. Thus, the core of difference will be laid in the value of the "satisfaction rate." First, the training result of single set of data is as follows [13, 14]:

According to the decision tree obtained by XGBoost training, the simplified results are shown in Table 3 (Fig. 2):

For the above-mentioned preprocessed single-group data threshold, the number of data sets satisfying the above threshold in 3 s is collected in the same way. The maximum values in different scenarios are as follows (Table 4):

Similarly, based on the preset threshold, the adjustment is: BORDER_AZZ = 30, BORDER_GYRO = 8, BORDER_AGZ = 25. The values satisfy 90.90, 72.72, 88.89% of the database fall data.

Table 3. XGBoost three-dimensional threshold training results

Single data angular velocity threshold	gyro > 13 3.6826 deg/s
Single data acceleration threshold 1	azz < 3,9453 m/s^2 or azz > 13.3214 m/s^2
Single data pitch angle threshold	agz < 19.9387° or 29.37185° < agz < 38.3750°

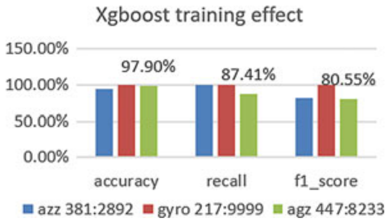


Fig. 2. XGBoost training effect

Table 4. Maximum "window satisfaction rate" in different scenarios

Activities	Maximum "window satisfaction rate" of combined acceleration (set)	Maximum "window satisfaction rate" of XOY plane angular velocity (set)	Maximum "window satisfaction rate" of pitch angle (set)
Running	205	3	40
Squatting	61	0	0
Walking	94	1	0
Upstairs and downstairs	62	0	0
Lie down	0	1	300
Jumping	92	0	22
The improved average "satisfaction rate" when falling	35	24	38

3 Performance Evaluation

For the threshold of acceleration 1, a person weighted 60 kg will receive a force of 770 N at the peak of the acceleration. The pressure applied by falling is the same as a free fall egg from 5th floor. It can be seen that the threshold is reasonable, and under this force, human body is indeed injured. And it is same as threshold 2 (Table 5).

The overall fall accuracy is relatively high, and the sensitivity and specificity are relatively balanced; thus, the detection effect is excellent. According to the data

Table 5. Parameter of the flowchart

Single set of data angular velocity threshold	gyro > 272.2687 deg/s
Single set of data acceleration velocity threshold 1	azz < 3.9453 m/s^2 or azz > 12.8214 m/s^2
Single sat of data acceleration velocity threshold 2	azz < 3.9453 m/s^2 or azz > 15.3214 m/s^2
Single set of data angle threshold	agz < 19.9387° or 29.37185° < agz < 38.3750°
The threshold of "window satisfaction rate": BORDER AZZ	30
The threshold of "window satisfaction rate": BORDER. GYRO	8
The threshold of "window satisfaction rate": BORDER AGZ	25
The accuracy of fall detection (based on 91 sets of data, which includes 59 sets of non-falling data and 32 sets of tailing data)	90.11%
Sensitivity	89.66%
Specificity	90.32%

analysis, the reason for the misjudgment to fall data is that the tester's action is too intense, making the action close to falling, thereby alarming. However, the false alarm with a small proportion can be ignored because it will not cause harmful consequences. For the case of falling but no alarm, we analyze: Some falls belong to "pseudo-fall"; that is, all the required features have occurred, but the data has not reached the threshold, and although it will not cause harmful consequences theoretically, the accuracy of this aspect needs to be improved.

The algorithm flowchart is shown in Fig. 3.

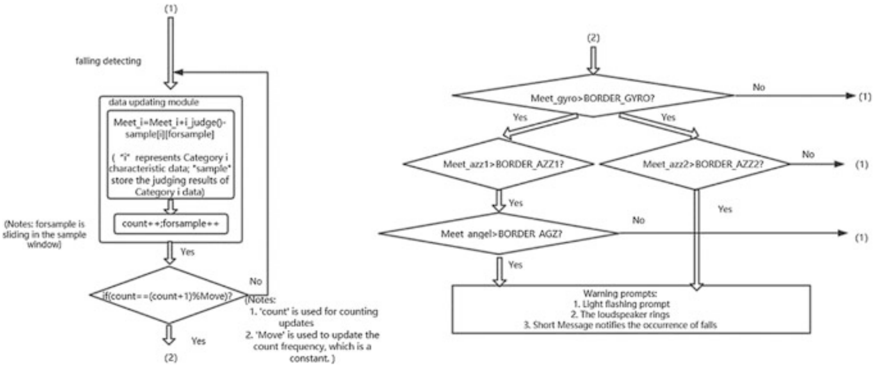

(1)
falling detecting
data updating module
Meet_i=Meet_i+i_judge()-sample[i][forsample]
("i" represents Category i characteristic data; "sample" store the judging results of Category i data)
count++;forsample++
(Notes: forsample is sliding in the sample window)
Yes
if(count==(count+1)%Move)?
No
(Notes:
1. 'count' is used for counting updates
2. 'Move' is used to update the count frequency, which is a constant.)
Yes
(2)
(2)
Meet_gyro>BORDER_GYRO?
No
(1)
Yes
Yes
Meet_azz1>BORDER_AZZ1?
Meet_azz2>BORDER_AZZ2?
No
(1)
Yes
Yes
Meet angel>BORDER AGZ?
No
(1)
Yes
Warning prompts:
1. Light flashing prompt
2. The loudspeaker rings
3. Short Message notifies the occurrence of falls

Fig. 3. Flowchart of falling algorithm

4 Conclusions

In this paper, we study fall detection algorithm based on machine learning. We extract the combined acceleration, the angular velocity of XOY plane, and the pitch angle as the characteristics of the falling process, in which pitch angle is introduced to distinguish the two sets of falls. In the preprocessing of data, we exploit the "full-field positioning" to obtain the pitch angle. Then, we proposed "sliding window method" to update the data. To solve the difficulty in locating the occurrence of falling, we proposed the "maximum satisfaction rate" method, which affects the marking of falling process in turn. Then, the XGBoost is used for threshold accurately training. Finally, the falling detection model we designed has received high accuracy with perfect balance between the sensitivity and specificity.

Acknowledgements. This work is supported by the 2016 Computer and Science Institute, Beijing University of Posts and Telecommunications.

This work is also supported by National Key R&D Program of China under grant number SQ2018YFC200148-03.

References

1. "Falls". World Health Organization. 16 January 2018. https://www.who.int/news-room/fact-sheets/detail/falls
2. The World Bank (2017) Population ages 65 and above (% of total). https://data.worldbank.org/indicator/SP.POP.65UP.TO.ZS
3. Shieh W-Y, Huang J-C (2009) Speedup the multi-camera video-surveillance system for elder falling detection. In: 2009 International conference on embedded software and systems. IEEE
4. Chen J et al (2006) Wearable sensors for reliable fall detection. In: 2005 IEEE Engineering in medicine and biology 27th annual conference. IEEE
5. Petelenz TJ, Peterson SC, Jacobsen SC (2002) Elderly fall monitoring method and device. U. S. Patent No. 6,433,690. 13 Aug 2002
6. Yuan X et al (2015) Fall detection analysis with wearable MEMS-based sensors. In: 2015 16th International conference on electronic packaging technology (ICEPT). IEEE
7. Gumus M, Kiran MS (2017) Crude oil price forecasting using XGBoost. In: 2017 International conference on computer science and engineering (UBMK). IEEE
8. Chen Z et al (2018) XGBoost classifier for DDoS attack detection and analysis in SDN-based cloud. In: 2018 IEEE international conference on big data and smart computing (BigComp). IEEE
9. Diebel J (2006) Representing attitude: euler angles, unit quaternions, and rotation vectors. Matrix 58(15-16):1–35
10. Shoemake K (1985) Animating rotation with quaternion curves. In: ACM SIGGRAPH computer graphics, vol 19, no 3. ACM
11. Noury N et al (2007) Fall detection-principles and methods. In: 2007 29th Annual international conference of the IEEE engineering in medicine and biology society. IEEE

12. Bourke AK, O'brien JV, Lyons GM (2007) Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. Gait Posture 26(2):194–199
13. Chen T et al (2015) Xgboost: extreme gradient boosting. R package version 0.4–2, pp 1–4
14. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM

Segmentation of Aerial Image with Multi-scale Feature and Attention Model

Shiyu Hu[1], Qian Ning[1,2](✉), Bingcai Chen[3], Yinjie Lei[1], Xinzhi Zhou[1], Hua Yan[1], Chengping Zhao[1], Tiantian Tang[2], and Ruiheng Hu[2]

[1] Sichuan University, Chengdu, Sichuan, China
hu_shiyu@163.com, ningq@scu.edu.cn, yinjie@scu.edu.cn, xz.zhou@scu.edu.cn, yanhua@scu.edu.cn, sc_zcp@scu.edu.cn
[2] Xinjiang Normal University, Urumqi, Xinjiang, China
[3] Dalian University of Technology, Dalian, China
china@dlut.edu.cn

Abstract. Aerial image labeling plays an important part in the mapping of maps with high precision. The knowledge about the range and intensive degree of aerial building segmentation is necessary for urban planning. Fully convolutional networks (FCNs) have recently shown state-of-the-art performance in image segmentation. In order to get better aerial images segmentation performance, we use a method of combing FCNs with multi-scale features and attention model in order to carry out segmentation automatically in aerial images. Attention model gives each scale feature added extra supervision to achieve better segmentation. Here, U-net and FCN-8s are used as original semantic segmentation model to train with multi-scale images and attention models. The datasets use different proportions of Inria Aerial Image Labeling Dataset, including two semantic classes: building and not building. The results show that the semantic segmentation model combined with multi-scale features and attention model has higher segmentation accuracy and better performance.

Keywords: Image segmentation · Aerial image labeling · Fully convolution neural networks · Attention model · Multi-scale feature

1 Introduction

Aerial image labeling contributes to the mapping of land cover and change detection and is used in areas such as forestry and urban planning. Due to the complexity of aerial image data, aerial image labeling has the following challenges:

- Occlusion—Partial or full occlusion can be caused by other objects such as tree, while partial occlusion which occurs more often is mostly caused by trees.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 58–66, 2020.
https://doi.org/10.1007/978-981-15-0187-6_7

- Unbalanced distribution—The building categories are unevenly distributed, some are densely or sparsely distributed, and there are no buildings in the image.
- Small size—In airborne imagery, the size of some buildings compared to other objects in the image is quite small. In some of the case, building just consists of only a few pixels.
- Shadow—Shadow creates a different illumination over buildings causing changes in their appearance. This reason, like the occlusion, could reduce the accuracy of automatic building labeling algorithms, especially deep learning methods which need a lot of training samples.
- Complex background—Structures such as load resemble with high similar building labeling.

These challenges raise the level of difficulty when it comes to image segmentation of aerial building image positioning and detection. Even for well-performing models, there is a high degree of uncertainty in the segmentation results [1]. The validity of the second part will be affected by the accuracy of image segmentation. Therefore, it is meaningful to use the state-of-the-art model to improve the accuracy.

2 Related Work

Semantic pixel-wise segmentation is always an active topic of research. In 2014, Berkeley [2] proposed a fully convolutional neural networks (FCNs) for segmentation. Based on the convolutional neural network, FCN changes the fully connected layer to 1×1 convolution layer. The success of FCN for semantic segmentation has more recently led researchers to exploit feature learning capabilities for segmentation. Badrinarayanan et al. [3] purposed SegNet, a typical encoder–decoder structure, achieving high scores for road scene understanding which is efficient both in terms of memory and computational time. A series of networks named DeepLab also used the convolution layer with dilation. DeepLab V1 [4] used deep convolutional neural networks (DCNNs) and fully connected conditional random field (fcCRF) to solve the problems. DeepLab V2 [5] added atrous spatial pyramid pooling (ASPP) based on V1, which enables segmentation on multiple scales. DeepLab V3 [6] proposed a module that consists of atrous convolution with various rates and batch normalization layers and experimented with laying out the modules in cascade or in parallel.

It is known that multi-scale feature is useful for computer vision task [7]. Farabet et al. [8] employed a Laplacian pyramid and share-net, passed each scale through a shared network, and fused the features of all scales. Eigen and Fergus [9] fed input images of three scales to DCNNs. The DCNNs at different scales have different structures, and this model required two-step process.

In computer vision, attention models are widely used in image classification [10] and object detection [11]. Chen et al. [12] proposed a mechanism of attention, which combine attention model and multi-scale. They learn an attention model

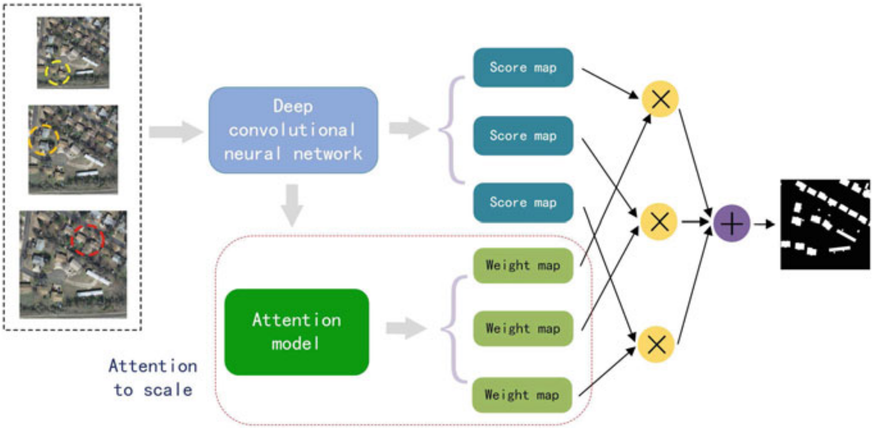


Fig. 1. Introduction of model.

that softly weights the features from different input scales when predicting the semantic label of a pixel. The final output of the model is produced by the weighted sum of score map across all the scales.

Here, we concentrate on building marking pixel-wise semantic segmentation using high-resolution aerial images, and our proposal is based on combing FCNs with multi-scale and attention model for this segmentation task. We use two different depths of baseline networks in experiment and combining them with models of multi-scale features and attention mechanisms as our new segmentation model.

3 Attention Model for Scales

There is a common method, which is share-net, resizes the input image to several scales, passes through a deep network sharing weights, and then computes the final prediction based on the fusion of the resulting multi-scale features. As shown in Fig. 1, we resize the input image to three scales and pass them through the same network to obtain score maps of different scales (the output of the last layer before softmax). Lastly, the fusion feature map feeds into attention model to generate weight map and multiplies the weight map with the fused feature map to get the final feature map. Herein, the attention model used here allows us to judge the importance of features at different locations and scales to achieve better segmentation.

3.1 Attention Mechanism

Attention mechanism developed from human visual research: Different attentions of the different parts are different, when focusing on a certain target or scene. Similarly, the most relevant parts of the statement and description change as the description changes. There are two types of attention methods: soft attention and hard attention, and its output vector distribution is soft and another one is hot, which will affect the selection of context directly. So attention model can improve network performance by focusing on the most relevant features as need. It is different from the attention model applied in two-dimensional and time [13], and attention model combined with multi-scale features is applied to semantic segmentation to improve the performance.

3.2 Attention Model for Scales

Herein, the attention model used is based on multi-scale features, which learns to softly weights for each scale and pixel. And this attention model is differentiable, so it trains end-to-end. As shown in Fig. 2, suppose an input image is resized to several scales $s \in \{1, \ldots, S\}$. Each scale is passed through the FCN (weights are shared across all scales) and produces a score map $f_{i,c}^{s}$ for scale, where i ranges over all the spatial positions, and $c \in \{1, \ldots, C\}$ where C is the number of classes of interest. The score map $f_{i,c}^{s}$ is resized to the same resolution by bilinear interpolation. $g_{i,c}$ is the weights sum of the score maps at (i, c) for all scales, so

$$g_{i,c} = \sum\nolimits_{s=1}^{s} \omega_i^s \bullet f_{i,c}^{s} \tag{1}$$

The weight ω_i^s is computed by

$$\omega_i^s = \frac{\exp(h_i^s)}{\sum_{t=1}^{s} \exp(h_i^t)} \tag{2}$$

where h_i^s is the score map produced by attention model in position i for scale s. The proposed attention model consists of two layers: the first layer is convolution operation that has 512 filters with kernel size 3×3; the second layer has S filter with kernel size 1×1, where S is the number of scales employed. The weight ω_i^s produced by attention model reflects the importance of feature at position i for scale s. Note in formulation 1, the average-pooling and max-pooling are two special cases: the weights ω_i^s will be replaced by $1/S$ for average-pooling; while the summation becomes the max operation and $\omega_i^s = 1 \forall s, i$ in the case of max-pooling. The attention model computes a soft weight for each scale and position, and it allows the gradient to be backpropagated through. Therefore, attention model can be trained with FCN part end-to-end and implementing the model adaptively to find the best weights on scales.

4 Experiments

4.1 Extra Supervision

In our experiments, we learn the network parameters by comparing the final output of the model with the corresponding ground truth for each image at the pixel-level. The final output is produced by performing a softmax operation on

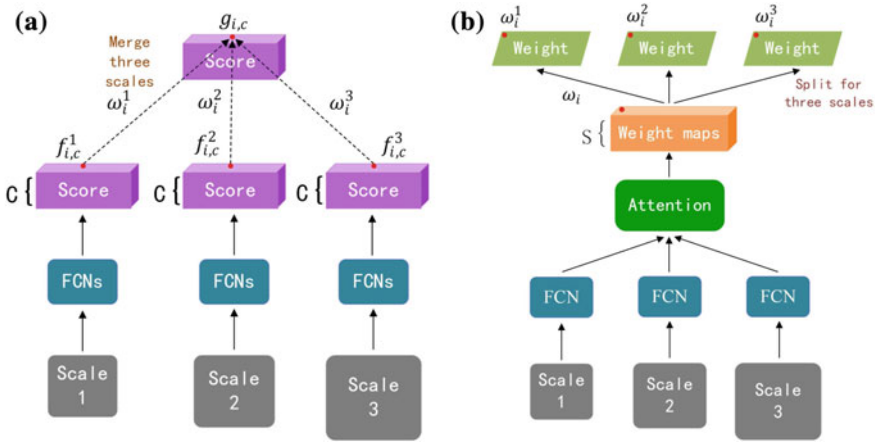


Fig. 2. Attention model implementation process left: merge feature map right: generate weight map.

the merged score maps across all the scales. To the optimization problem of finding, the minimum value in the BCE loss is solved by Adam optimizer and back-propagation process. At the same time, the experimental data is divided into three different proportions to obverse the effect of convergence speed. Adam with mini-batch is used for training. We set the initial learning rate of 0.001, and epoch is 100. The learning rate is multiplied by 0.1 after 2000 iterations. We use a momentum of 0.9 and weight decay of 0.0005. Due to hardware limitation, the mini-batch size id set to 5 when training the basic network and 2 when training the network with attention model.

4.2 Network Architectures

Our network is based on the publicly available model: FCN-8s [2] with VGG16 as a backbone network and U-net [14]. They all have proven effective in semantic segmentation. FCN-8s use all of the pre-trained convolutional layer weights from VGG-16 as pre-trained weights; for U-net network parameters, Gaussian initialization is used to initialize parameters.

4.3 Datasets

The experiments were conducted using images acquired by the Inria Aerial Image Labeling Dataset, which covers multiple urban areas from densely populated areas to alpine towns, with high spatial resolution. Therefore, in aerial image labeling, the goal is to classify each pixel as building marking class (foreground) or non-building marking (background). The training set includes 5 regions, each region contains 36 tiles, numbered 1–36. According to the datasets, we remove the first five images of every location from the training set. To submit results, use the exact same file names as the input color images, and output 0/255 8-bit single-channel files in the validation set. The training set and validation set are divided into three different proportions for training, 31:5, 26:5, and 21:5. Different proportions of datasets can affect model convergence. Since the dataset is a high-resolution image, the image is cut into 512 × 512 size. When using the attention model for training, three scales are required and resize the input images to add extra scales: 256 × 256, 1024 × 1024.

4.4 Evaluation

Concerning the evaluation criteria, we use the Intersection over Union (IoU) of positive (building) class, which are widely used in semantic segmentation task. In these metrics, n_{ij} is the pixel number belonging to class i which has been predicted as class j and n_{cl} stand for the number of classed with $t_i = \sum_j n_{ij}$ representing the total number of pixels belonging to class i. We use the dice similarity coefficient also due to the heavy unbalance in the dataset. The number of pixels belonging to each class does have an effect on these two criteria. X and Y represent prediction and ground truth, respectively. The criteria are derived as follows:
IoU of positive class:

$$\mathrm{IoU} = \frac{\mathrm{area}\,(C) \cap \mathrm{area}\,(G)}{\mathrm{area}\,(C) \cup \mathrm{area}\,(G)} \tag{3}$$

Dice similarity coefficient:

$$\frac{1}{n_{cl}} \sum\nolimits_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \tag{4}$$

4.5 Results

From Table 1, for the dataset with ratio of 31:5, the network combines multi-scale features and attention mechanism. The IoU and dice coefficients of the deeper neural network U-net are higher (IoU is 0.11 higher and dice coefficient is 0.9 higher), due to deeper networks which are able to extract higher-level features. By comparing FCN-32s and U-net combining multi-scale features and attention mechanisms respectively, found that our model, IoU and Dice coefficient are improved, and the deeper network U-net segmentation effect is better (IoU is 0.784 Dice coefficient is 0.879). The results show that the segmentation performance of semantic segmentation model combined with multi-scale features and attention mechanism is improved.

The result of the segmentation is shown in Fig. 3. All images are the same size: 512 × 512, from top to bottom: aerial image, basic network result, the combined attention model result, and ground truth; from left to right, the training results of the model in the dataset at different proportion (the ratio is from small to large). The segmentation results prove that the network with multi-scale and

Table 1. Result of experience.

Segmentation model	IoU	Dice
FCN-8s	0.653	0.769
FCN-8s and attention	0.675	0.796
U-net	0.762	0.857
U-net and attention	0.784	0.879

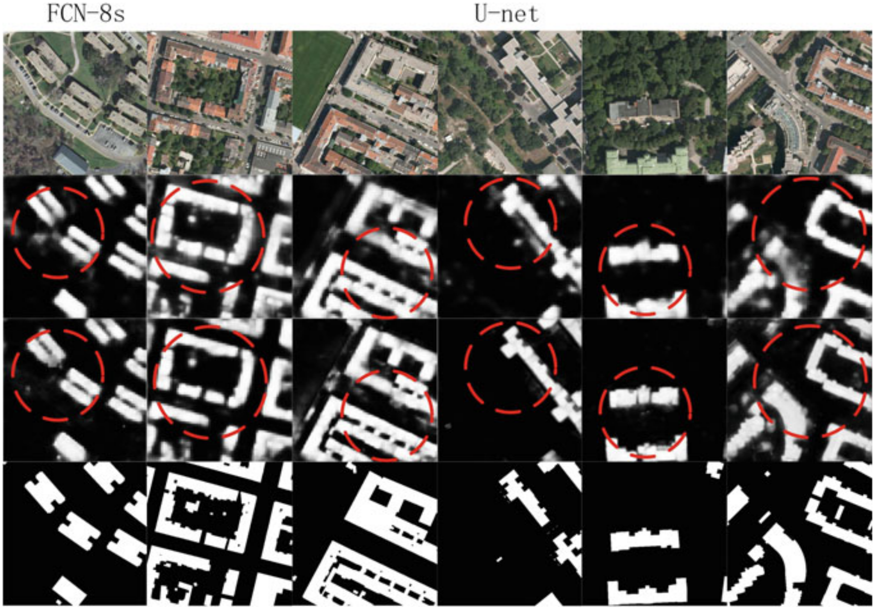


Fig. 3. Experimental result.

attention mechanism has a better segmentation effect, and boundary is clearer. The labeling part of the resulting figure can better reflect that the model combined with attention mechanism. By comparing the segmentation effects of every image, our model effectively reduces the mistakes of dividing roads into buildings, and the degree of blurring of the boundary; it is more accurate for building shape segmentation. However, the method can be further improved due to some segmentation mistakes. For example, when dataset ratio is 21:5 and the segmentation network is U-net(third column), our model has a clearer segmentation boundary than baseline network, but by comparing it with ground truth, our model marks the road with similar building pixel values as building, resulting in segmentation mistakes. In general, the results prove that the network model combined with multi-scale and attention mechanism has a better segmentation effect and clearer architectural detail segmentation.

5 Conclusion

In this work, we combine multi-scale features and attention mechanisms to achieve aerial image segmentation with high accuracy and robustness. In this method, multi-scale features are used, and the attention model is combined with FCN-8s and U-net, respectively, to make the model adaptive, to find the optimal weight on the scale, and to achieve end-to-end training, which compared with our FCNs based-line network. The experimental results show that the multi-scale features are better than the single-scale segmentation; attention model can add additional supervision for better model performance by generating a soft weight at different positions of each scales. Therefore, the segmentation network model combining with multi-scale features and attention mechanism is applied to aerial image labeling, which can effectively improve the segmentation effect.

References

1. Lagrange A, Le Saux B, Beaupere A et al (2015) Benchmarking classification of earth-observation data: from learning explicit features to convolutional networks. In: Geoscience and remote sensing symposium (IGARSS). IEEE, Milan, pp 4173–4176
2. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Boston, pp 3431–3440
3. Badrinarayanan V, Kendall A, Cipolla R (2015) Segnet: a deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561
4. Chen LC, Papandreou G, Kokkinos I et al (2014) Semantic image segmentation with deep convolutional nets and fully connected CRFs. arXiv preprint arXiv:1412.7062
5. Chen LC, Papandreou G, Kokkinos I et al (2018) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans Pattern Anal Mach Intell 40(4):834–848
6. Chen LC, Papandreou G, Schroff F et al (2017) Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587
7. Arbelaez P, Maire M, Fowlkes C et al (2011) Contour detection and hierarchical image segmentation. IEEE Trans Pattern Anal Mach Intell 33(5):898–916
8. Pinheiro PH, Collobert R (2013) Recurrent convolutional neural networks for scene parsing, 2. arXiv preprint arXiv:1306.2795
9. Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proceedings of the IEEE international conference on computer vision. IEEE, Santiago, pp 2650–2658
10. Cao C, Liu X, Yang Y et al (2015) Look and think twice: capturing top-down visual attention with feedback convolutional neural networks. In: Proceedings of the IEEE international conference on computer vision. IEEE, Santiago, pp 2956–2964
11. Ba J, Mnih V, Kavukcuoglu K (2014) Multiple object recognition with visual attention. arXiv preprint arXiv:1412.7755
12. Chen LC, Yang Y, Wang J et al (2016) Attention to scale: scale-aware semantic image segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, Las Vegas, pp 3640–3649

13. Sharma S, Kiros R, Salakhutdinov R (2015) Action recognition using visual attention. arXiv preprint arXiv:1511.04119
14. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention. Springer, Munich, pp 234–241

Recurrent Neural Detection of Time–Frequency Overlapped Interference Signals

Qianqian Wu, Zhuo Sun(✉), and Xue Zhou

Wireless Signal Processing and Network Laboratory, Beijing University of Posts and Telecommunications, Beijing 100876, China
{wuqq, zhuosun, 277210680}@bupt.edu.cn

Abstract. For interfering signals overlap with normal signals in both time and frequency domain, it is difficult to detect them. Therefore, this paper proposes a novel bidirectional recurrent neural network-based interference detection method. By utilizing the ability of recurrent neural network of extracting the nonlinear features of the time series context, the model can get a prediction of following signal samples and calculate the difference between prediction signal and original signal to do interference detection. The proposed method can achieve a better sensitivity and determine the exact location of the complete interfering signal. In the experiment part, we demonstrate the efficacy of this method in multiple typical scenarios of time–frequency overlapped wireless signals.

1 Introduction

The development of wireless communication makes it be widely used in various fields, but the electromagnetic environment is complex and changeable, and the reliability of the communication system is still threatened by interference. Therefore, the anti-interference technology is necessary to ensure the communication reliability, and the interference detection technology is the basis and key of the communication anti-interference technology.

The purpose of interference detection is to determine whether there are interfering signals in received signals and then feed back to the transmitter or command center to take effective anti-interference measures. Traditional wireless interference detection approaches include time domain and transform domain-based energy detection algorithm (e.g., consecutive mean excision (CME) and forward consecutive mean excision (FCME)). Some approaches analyze the received signal strength indicator (RSSI) samples in the frequency and time domain [1,2] or to perform a cyclostationary signal analysis and blind signal detection and other spectrum sensing techniques [3]. These methods can well detect many kinds of interfering signals. However, there are many interfering signals whose power is small and frequency is the same as original signals in the

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 67–75, 2020.
https://doi.org/10.1007/978-981-15-0187-6_8

actual communication environment. When these interfering signals are superimposed on the original signal, the time and the frequency domain features do not change significantly. We call these interfering signals as time–frequency overlapped signals. Obviously, the time–frequency overlapped signals are hard to detect by above methods.

The research found that the neural network can better extract the time and frequency domain feature of signals. The feature can be used as signal fingerprint to classify signals [4,5]; therefore, many interference detection methods based on deep learning emerged in recent years. A sequential autoencoder framework is introduced to distinguish normal and interfering signal in [6]. They calculate the difference between original signals and reconstructed signals by autoencoder framework. In [7,8], predicting future signals from known signals by utilizing the prediction function of recurrent neural network and then taking the difference between original and predictive signals as a feature to do interference detection. However, these methods have not good detection performance under low signal-to-interference-plus-noise ratio (SINR) conditions, especially for time–frequency overlapped interfering signals.

In this paper, we present a bidirectional recurrent neural network (BI-RNN)-based interference detection structure, which can utilize the correlation between the front and back sampling points to predict the data of the intermediate position. The structure can do bidirectional training and has the better prediction performance; hence, the difference between predictive and original signals at the interference-containing part is larger. And then in prediction process, we adopt special training labels for noise reduction; therefore, the detection accuracy is improved under low SINR condition. Finally, in the detection process, we use the feature correlation classification instead of simply taking a decision threshold, which reduces the contingency of the decision process and can completely identify the complete interfering signal.

The rest of this paper is organized as follows. In Sect. 2, the interfering signal and the principle of interference detection are described. Then, the neural network model and the special training method are explained in Sect. 3. Section 4 shows the performance of interference detection and analysis of influencing factors. Finally, Sect. 5 concludes the paper and suggests future work.

2 Problem Formulation

When there is no interference, the received signal can be represented by Eqs. 1, and 2 and the received signal contains interference.

$$r(t) = s(t) + n(t), \tag{1}$$

$$r_j(t) = s(t) + n(t) + j(t). \tag{2}$$

where $s(t)$ is a carrier signal transmitted by transmitter, $n(t)$ represents noise, and $j(t)$ is a interfering signal(In this paper, we consider $j(t)$ as the time–frequency overlapped interfering signal.) The aim of interference detection is to find the exact position of $j(t)$.

As shown in Fig. 1, a predictor f is trained by normal signal r and we can get the predictive signal $\widehat{r}$ from f. Then, calculate the difference e between r and $\widehat{r}$ by Eq. 3 and e can be considered as a feature to train a classifier for interference detection. Similarly, when the interference-containing signal r_j uses f to predict, we can get the predictive signal $\widehat{r_j}$ and the difference e_j is calculated by Eq. 4. It is obvious that e_j is greater than e, because the presence of interference destroys the correlation of the original signal, which makes f unable to accurately predict the interference-containing signal. Finally, the classifier can find interfering signal based on the difference between e and e_j.

$$e = \left| r - \widehat{r} \right|, \tag{3}$$

$$e_j = \left| r_j - \widehat{r_j} \right|. \tag{4}$$

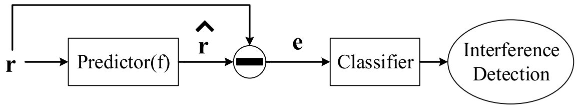


Fig. 1. General process of interference detection

3 BI-RNN-Based Prediction of Time Series Signal

It can be seen from Sect. 2 that the core of the interference detection method proposed in this paper is the construction of prediction model. Research shows that RNN can extract the nonlinear features of time series [9], so choose RNN predictor here. As a special RNN, bidirectional Long Short-Term Memory (BI-LSTM) adds a set of weight parameters for backward calculation; therefore, it can also utilize the data information after the sample points to be predicted for prediction. For that reason, we built the BI-LSTM-based model for signal prediction.

3.1 Prediction Model

At the very beginning, we should preprocess the signal into a form suitable for BI-LSTM. Supposing the time series of received signal is $Y = \left\{ y_1, y_2, y_3, \ldots, y_t, y_{(t+1)}, \ldots, y_{(M)} \right\}$, choose $Y_{\text{train}} = \{ y_1, y_2, y_3, \ldots, y_t, y_{(t+C+1)}, \ldots, y_{(M)} \}$ as training set (C and M are fixed values) and $Y_{\text{label}} = \{ y_{(t+1)}, y_{(t+2)}, \ldots, y_{(t+C)} \}$ as label to predict $Y_{\text{pre}} = \left\{ \widehat{(y_{(t+1)})}, \widehat{(y_{(t+2)})}, \ldots, \widehat{(y_{(t+C)})} \right\}$. Through a lot of iterations calculation, the network will constantly adjust the weights to make Y_{label} and Y_{pre} closer and closer. Then the computed Y_{error} is used to determine if it is interference-containing signal. Y_{error} is defined as $Y_{\text{error}} = |Y_{\text{label}} - Y_{\text{pre}}|$.

The specific process of signal prediction is shown as Fig. 2. Divide N time-banks of M time steps shifting by C time steps between adjacent chunks. If the total signal sampling points is T, we can calculate the number of time-banks by $N = \frac{T-M}{C} + 1$.

For each time-bank, remove p consecutive sample points inside, and set the remaining sample points as one input. So the output from each input is the predictive p samples, and the output from the N inputs is a continuous predictive signal. In theory, the larger the value of M, the better the prediction performance. In Fig. 2, we set $M = 110$ and $p = 1$.

The reason why we use BI-LSTM is that BI-LSTM adds a delay between the input and the target to give the network some time to add future context information for prediction [10,11]. The hidden layers of the Bi-LSTM store two sets of parameters, one for forward calculation and the other for backward calculation, and the final outputs depend on both parameters. But experiments have shown that when the sequence after the predicted value is too long, the network will pay more attention on the behind part and the predictive performance will worsen. Therefore, should choose the appropriate value (here we set 10). In order to verify the validity of the proposed method, we compared it with the predictive model of LSTM mentioned in [7]. The BI-LSTM predictive model is described as follows.

- The stacked BI-LSTM sequence predictor model is implemented with a 3-layer BI-LSTM followed by a fully connected layer culminating in a linear activation for output. The dropout between each layer is 0.2 and chooses mean squared error loss function (MSE) as the loss function. Besides, the number of hidden layer is 128, and batch size is 64.

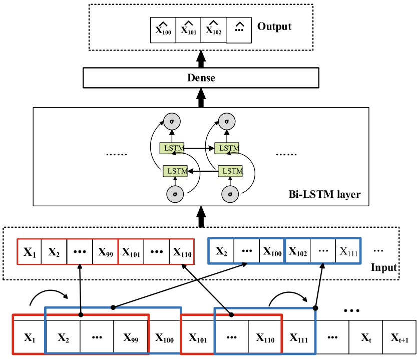

$\hat{X}_{100}$
$\hat{X}_{101}$
$\hat{X}_{102}$
$\hat{\cdots}$
Output
Dense
σ
LSTM
......
Bi-LSTM layer
X_1 X_2 ··· X_{99} X_{101} ··· X_{110}
X_2 ··· X_{100} X_{102} ··· X_{111}
···
Input
X_1 X_2 ··· X_{99} X_{100} X_{101} ··· X_{110} X_{111} ··· X_t X_{t+1}

Fig. 2. Stacked BI-LSTM prediction model for signal prediction

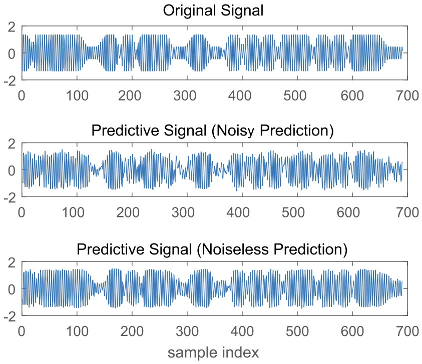


Fig. 3. Predictive signal by using BI-LSTM

3.2 Model Training for Noiseless Prediction

During the research, it was found that when the SINR is relatively small, the network extract the signal features hardly; therefore, the predictive signal and the original signal are very different. The implementation of speech enhancement uses LSTM by learning the correlation between noisy signals and noiseless signals in [12]; hence, we are inspired by speech enhancement to propose a novel training method which can improve the prediction performance under low SINR conditions. We call this prediction process as noiseless prediction.

In noiseless prediction, the training data is the received signal with noise, but the input label during training is the corresponding noiseless signal sampling point. This method makes the network focus only on the correlation of the signal itself, regardless of the irregularity of noise; therefore, the output is closer to the noiseless signal. Figure 3 depicts the digital modulation signal predicted by BI-LSTM, when the SINR = 6 db. The first picture is the original signal without noise for comparison, and then the second is the predictive signal when use noise signal as label, and the third is the predictive signal when use noiseless signal as label. From the figure, we can see that noiseless prediction model has better predictive performance than the noisy prediction model.

3.3 Prediction Performance

To measure the effectiveness of different neural network, the mean absolute errors (MAE) and mean absolute percentage errors (MAPE) are computed in Eqs. 5 and 6:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|x_i - \widehat{x_i}\right|, \tag{5}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{x_i - \widehat{x_i}}{x_i}\right|. \tag{6}$$

where x_i is the actual signal sampling point at i_{th}, $\widehat{x_i}$ is the predictive signal sampling point, and n is the total number of test signal sampling points.

All the predictive result of different models in this section trained and tested multiple times to eliminate outliers. We compare the prediction performance of LSTM, BI-LSTM and the case of noiseless prediction in Table 1.

We can see from Table 1 that the prediction result is getting better with the increase of SINR, but the influence becomes less obvious when the SINR reaches a certain level. This proves that Gaussian white noise does interfere with the learning ability of the network. In addition, the performance of BI-LSTM model is better than that of LSTM, although not obvious, which indicates that the sequence correlation after a certain signal sampling point will affect the prediction result of model. Finally, we can also see that noiseless prediction obtains the best results, especially under low SINR condition. This proves that noiseless prediction does have good denoise ability.

Table 1. Prediction performance of three models

	Performance					
Model	SINR	0 db	4 db	8 db	12 db	16 db
LSTM	MAE	0.437	0.422	0.277	0.203	0.118
	MAPE	0.796	0.950	0.623	0.456	0.278
BI-LSTM (noisy prediction)	MAE	0.565	0.413	0.267	0.168	0.109
	MAPE	1.099	0.945	0.620	0.454	0.275
BI-LSTM (noiseless prediction)	MAE	0.257	0.205	0.163	0.103	0.07
	MAPE	0.592	0.548	0.361	0.248	0.146

Figure 4a shows the comparison of original signal and predictive signal based on digital modulation when the SINR = 12 db. Similarly, Fig. 4b demonstrates the case of FM modulation. We can find from the pictures that the difference between the predictive signal and the original noiseless signal in the normal part is much smaller than in the interference-containing part. So the difference of predictive signal and original signal can be used for interference detection.

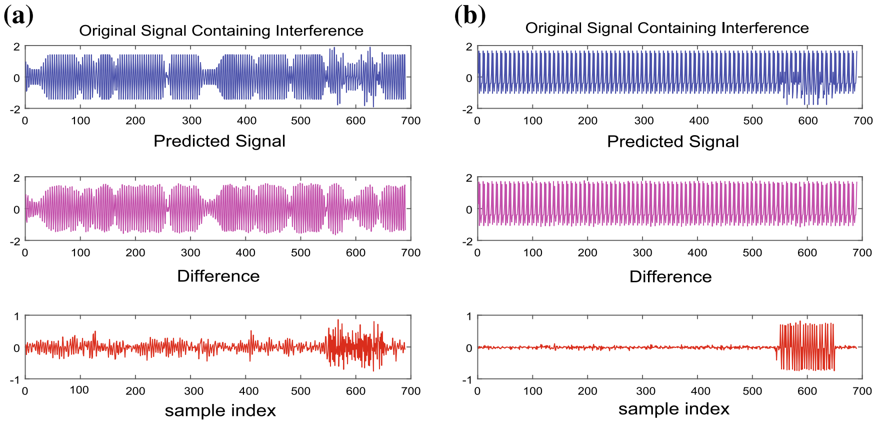


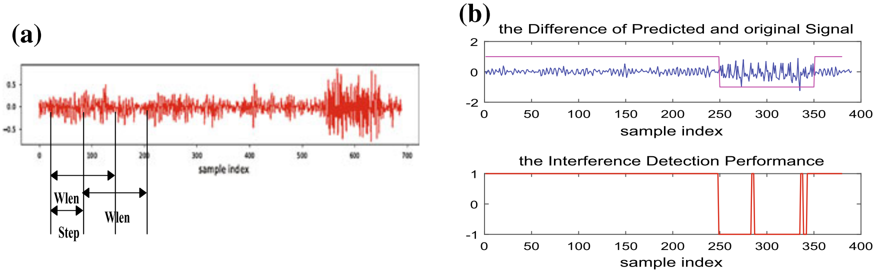
Fig. 4. **a** is the difference between predictive signal and original signal (digital modulation), and **b** is the difference between predictive signal and original signal (analog modulation)

4 Interference Detection

4.1 Classifier for Interference Detection

The features calculated by Sect. 3 can be used for interference detection. In order to overcome the factor of sample imbalance, select support vector data description (SVDD) here.

During the detection process, it is found that when the bias of each sampling point is used as the feature to train SVDD classifier, it is hard to determine the specific location of interference signal. So the feature is windowed according to the time step and is detected according to the waveform and correlations of the plurality of feature points to overcome the contingency. The windowing process is expressed as Fig. 5a. The length of the window (Wlen) we selected is 10, and the step size (Step) is 1 here. The experimental results are shown in Fig. 5b. The part marked with purple in the first picture is the interference-containing signal. The second picture shows the result of test. The part with a value of −1 is the signal that is judged to be interference, and the part with a value of 1 is the signal that is judged to be normal. It can be seen that most of the interference signal sampling points can be detected.



the Difference of Predicted and original Signal

the Interference Detection Performance

Fig. 5. **a** is the windowing process, and **b** is the interference detection performance at 12 db (QPSK modulation)

In order to evaluate performance comprehensively, four typical experimental scenarios are set as follows, which represent various types of interference that may occur in the actual communication system. In addition, to reduce the influence of the amplitude, we have done the amplitude normalization on each signal.

- Normal signal is interfered by interference signal with different modulation mode:

1. The 16QAM signal is normal and the superimposed QPSK signal with the same sampling rate, symbol rate, and carrier rate is interference (QAM-QPSK).
2. The 16QAM signal is normal and the superimposed FM signal is interference (QAM-FM).
3. The FM signal is normal, and the superimposed DSSS signal is interference (FM-DSSS).

- Normal signal is interfered by interference signal with different symbol rates:

4. The 16QAM signal is normal, and the superimposed 16QAM signal with only different symbol rate is interference (QAM-QAM).

4.2 Performance Evaluation

We repeat these experiences in three models above-mentioned and get Fig. 6 which show the F1 score of the performance of three models under different SINR. $F1_{\text{score}}$ is calculated as $F1_{\text{score}} = \frac{2PR}{P+R}$ (where P represents precision and R represents recall). Obviously, when using the noiseless signal as label, we get the best performance, especially at low SINR. In the case of noise label, although the MAE and MAPE values of Bi-LSTM are smaller than those of LSTM, the detection accuracy is not much higher.

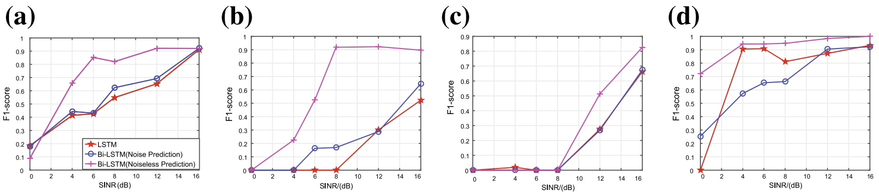


Fig. 6. Interference detection performance of three models: **a** is the detection performance of QAM-FM signal, **b** is the detection performance of QAM-QPSK signal, **c** is the detection performance of QAM-QAM signal, **d** is the detection performance of FM-DSSS signal

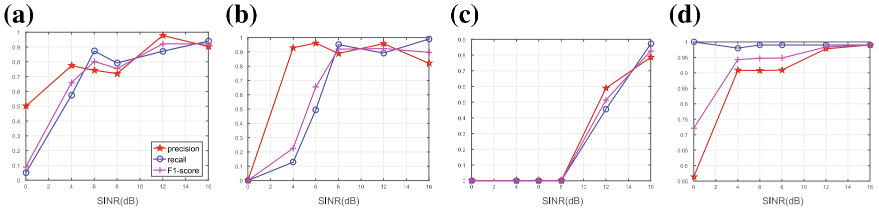


Fig. 7. Interference detection performance of BI-LSTM (noiseless prediction) under different evaluation criteria: **a** is the detection performance of QAM-FM signal, **b** is the detection performance of QAM-QPSK signal, **c** is the detection performance of QAM-QAM signal, **d** is the detection performance of FM-DSSS signal

Figure 7 shows the experimental results of BI-LSTM when the noiseless signal as label. We can see that the detection precision and recall significantly be improved with the increase of SINR. Almost all the interferences are determined to be normal when the SINR = 0 db. When the SINR is higher than 6 db, perfect detection performance is achieved. Among the four scenarios, only the QAM interference with different symbol rates is the most difficultly to detect. The DSSS interference in FM modulation is the most easily to detect, which shows that the signal of digital modulation is hard to predict because of the randomness of its own symbols.

5 Conclusion

In this paper, a more robust method is proposed for wireless signal interference detection at low SINR and is capable of localizing to each interference signal sampling point. The results show that the model can implement the interference detection which not applicable to traditional methods, and the detection accuracy is better than other neural network models in the time–frequency overlapped interfering signal. We believe the result can be used in interference detection in complex communication environments. However, the experiment proves that the detection result is relatively poor for the case where the SINR is lower than 6 db. So we will work on how to improve detection performance under low SINR condition next.

References

1. Rayanchu S, Patro A, Banerjee S (2011) Airshark: detecting non-WiFi RF devices using commodity WiFi hardware. In: ACM SIGCOMM conference on internet measurement conference
2. Lakshminarayanan K, Sapra S, Seshan S, Steenkiste P (2009) RFDump: an architecture for monitoring the wireless ether. In: International conference on emerging networking experiments & technologies
3. Chen X, Chen H, Chen Y (2011) Spectrum sensing for cognitive ultra-wideband based on fractal dimensions. In: Fourth international workshop on Chaos-fractals theories & applications
4. Merchant K, Revay S, Stantchev G, Nousain B (2018) Deep learning for RF device fingerprinting in cognitive communication networks. IEEE J Sel Top Sig Process 12(1):160–167
5. Sainath TN, Li B (2016) Modeling time-frequency patterns with LSTM vs. convolutional architectures for LVCSR tasks. In: Interspeech
6. Mirza AH, Cosan S (2018) Computer network intrusion detection using sequential LSTM neural networks autoencoders. In: 2018 26th signal processing and communications applications conference (SIU), May 2018, pp 1–4
7. O’Shea TJ, Clancy TC, McGwier RW (2016) Recurrent neural radio anomaly detection. CoRR, arXiv:1611.00301
8. Malhotra P, Vig L, Shroff G, Agarwal P (2015) Long short term memory networks for anomaly detection in time series. In: Proceedings. Presses universitaires de Louvain, p 89
9. Xiao Q, Si Y (2017) Time series prediction using graph model. In: 2017 3rd IEEE international conference on computer and communications (ICCC), pp 1358–1361
10. Cui Z, Ke R, Wang Y (2018) Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. CoRR, arXiv:1801.02143
11. Chen Y, Qian J, Yang J, Jin Z (2016) Face alignment with cascaded bidirectional LSTM neural networks. In: 2016 23rd international conference on pattern recognition (ICPR), pp 313–318
12. Kolbk M, Tan Z, Jensen J (2016) Speech enhancement using long short-term memory based recurrent neural networks for noise robust speaker verification. In: 2016 IEEE spoken language technology workshop (SLT), pp 305–311

Risk Analysis of Lateral Collision of Military and Civil Aviation Aircraft Based on Event Model

Guhao Zhao[1,2](✉), Shaojie Mao[1], Yi Mao[1], Xiaoqiang Zhang[2], and Yarong Wu[2]

[1] State Key Laboratory of Air Traffic Management System and Technology,
Nanjing 210007, Jiangsu, China
zghlupin@163.com

[2] Air Traffic Control and Navigation College, Air Force Engineering University,
Xi'an 710077, Shaanxi, China

Abstract. Collision risk analysis is an important part of airspace safety assessment. Based on Event model, the lateral position deviation probability of military aircraft is calculated. Combining with the probability model of lateral position deviation of civil aviation aircraft, the frequency of military aircraft collision box passing through separation sheet in high slope circling training is calculated. A collision risk assessment model between training flying aircraft in military training airspace and aircrafts flying in civil aviation route is constructed. Through the simulation calculation of the lateral collision risk of military and civil aviation, the size, layout, and use suggestions of military high slope circling training airspace are obtained, which can provide reference for the airspace safety assessment.

Keywords: Airspace safety assessment · Collision risk · Military training airspace · Air route

1 Introduction

In recent years, with the rapid development of civil aviation industry and the rapid growth of air traffic flow, at the same time, the renewal of Air Force Weapons and equipment is accelerating, and the demand for airspace for military flight training is also increasing. This makes the contradiction between military aviation and civil aviation in the demand for space resources increasingly prominent. On the premise of meeting the requirement of safety target grade, according to the requirement of military flight training subjects, it is an effective way to solve the contradiction between military aviation and civil aviation to delimit the airspace scope suitable for training needs and improve the utilization ratio of airspace.

In the 1960s, Reich first proposed the REICH collision risk model [1], which is used to analyze the safety of air routes. In 2003, Peter Brooker proposed Event model

Project supported by the National Natural Science Foundation of China (Grant Nos. 61601497).

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 76–86, 2020.
https://doi.org/10.1007/978-981-15-0187-6_9

[2] to assess the risk of lateral collision between civil aviation aircraft. With the advent of various collision risk models, many scholars at home and abroad have made in-depth research on them and put forward various improved methods [3]. However, the above studies only focus on the collision probability between civilian aircraft flying in the air route, and there is little literature on the collision probability between military aviation and civil aviation. In this paper, the Event model is used to model and calculate the collision risk between the civilian aviation aircraft in the air route and military aircraft in the adjacent training airspace.

2 Event Model

According to the original Event model, a separation sheet with no thickness is defined centering on aircraft "B." A collision box is defined centering on aircraft "A." In the process of passing through the separation sheet, the collision box leaves a projection on the separation sheet, which is defined as an extended collision box. According to the probability theory, the collision probability of two aircraft is equal to the product of the probability of the collision box passing through separation sheet and the probability that aircraft "B" is located in the extended collision box, as shown in Fig. 1.

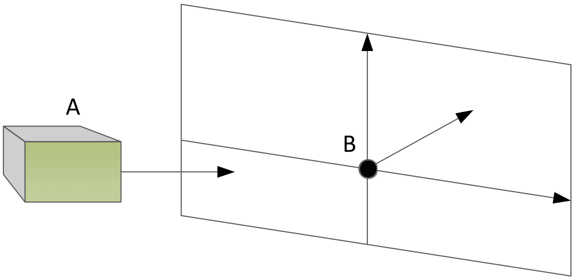


Fig. 1. Collision box passing through separation sheet

The collision probability is shown in Formula 1.

$$N_{\mathrm{ay}} = \mathrm{GERh}\frac{E(S)}{2L}\left(2\lambda_x + \frac{U2\lambda_{\mathrm{y}}}{V}\right)P_{\mathrm{z}}(0)\left(1 + \frac{W2\lambda_{\mathrm{y}}}{V2\lambda_{\mathrm{z}}}\right) \tag{1}$$

where GERh is the probability of lateral overlap of two aircrafts per hour, and L is the longitudinal separation standard. $E(S)$ is the logarithm of the aircraft flying in the same direction within a distance of $2L$. λ_x, λ_y, λ_z are the length, width, and height of the collision box, respectively. When aircraft "A" passing through the separation sheet, U, V, and W are the relative speeds of the two aircraft in the longitudinal, lateral, and vertical directions, respectively. $P_{\mathrm{z}}(0)$ is the probability of vertical overlap between two aircraft at the same flight level.

In Eq. 1, the parameter GERh is statistical data, and however, such a statistical database is missing between military and civil aircraft. And in the air route, the aircraft heading at the same flight level is the same, and the heading of the military training flight is not fixed. For the above two reasons, the Event model cannot be directly applied to the assessment of the collision risk of military and civil aviation. Therefore, the Event model needs to be modified for the characteristics of military and civil aviation flights.

3 Collision Probability Model of Military and Civil Aviation

3.1 Hypothetical Conditions of the Model

(1) Assuming that the long side of the military training airspace is parallel to the air route, military aircraft and civil aviation aircraft fly at the same altitude.
(2) The interval between military training airspace and air route is 10 km.
(3) The positions of military aircraft and civil aircraft are independent of each other, and each uses independent navigation facilities.
(4) Considering that in military flight training, most aircraft are turning when approaching the boundary of airspace, this paper calculates the collision probability of military and civil aviation aircraft by taking the high slope circling training as an example.

3.2 Event Collision Model for Military and Civil Aviation

A rectangular collision box is defined with military training aircraft A as the center. The length, width, and height of the collision box are λ_{hx}, λ_{hy}, λ_{hz}:

$$\begin{cases} \lambda_{hx} = \lambda_{mx} + \lambda_{jx} \\ \lambda_{hy} = \lambda_{my} + \lambda_{jy} \\ \lambda_{hz} = \lambda_{mz} + \lambda_{jz} \end{cases} \tag{2}$$

where λ_{jx}, λ_{jy}, λ_{jz} and λ_{mx}, λ_{my}, λ_{mz} are the fuselage length, wingspan, and fuselage of military aircraft and civil aircraft, respectively, and consider the civil aircraft as a point B. When point B is in contact with the collision box A, It can be considered that two aircraft have collided.

Unlike the traditional Event model, military aircraft and each passenger airliner can be considered as pairs of aircraft with potential collision risks. Therefore, the Event model can be rewritten as

$$\begin{aligned} N_{\mathrm{ay}} = {} & \mathrm{GERh}\frac{2E(S)}{2L}\left(\lambda_{hx} + \frac{U_{\mathrm{s}}\lambda_{hy}}{V_{\mathrm{s}}}\right)P_{\mathrm{z}}(0)\left(1 + \frac{W_{\mathrm{s}}\lambda_{hy}}{V_{\mathrm{s}}\lambda_{hz}}\right)\beta \\ & + \mathrm{GERh}\frac{2E(O)}{2L}\left(\lambda_{hx} + \frac{U_{\mathrm{o}}\lambda_{hy}}{V_{\mathrm{o}}}\right)P_{\mathrm{z}}(0)\left(1 + \frac{W_{\mathrm{o}}\lambda_{hy}}{V_{\mathrm{o}}\lambda_{hz}}\right)\beta \end{aligned} \tag{3}$$

Among them, β is the probability of military aircraft training in the military training airspace within 1 h. $2E(S)$ and $2E(O)$, respectively, indicate how many pairs of military and civil aircraft are in the same direction and in the opposite direction in the air route of length $2L$. U_S, V_S, W_S, and U_O, V_O, W_O are relative speeds in the longitudinal, lateral, and vertical directions when aircraft A and aircraft B fly in the same direction and in the opposite direction. The relative speeds of the two aircrafts in the lateral and vertical directions are the same whether they are in the same direction or in the reverse direction, so $V_S = V_O$ and $W_S = W_O$.

In the traditional Event model, GERh is defined as the frequency at which the collision box passes through the separation sheet, replacing the GERh with the statistical frequency of the civil aircraft's lateral interval of zero in one hour. However, there is no corresponding statistical data on the assessment of the risk of collision between military aircraft and civil aircraft. This paper estimates the GEHR value of military and civil aircraft based on the lateral position deviation probability model of the civil aircraft [4] and the lateral position deviation probability model of the military aircraft.

4 Military and Civil Aircraft Lateral Overlap Probability

4.1 Coordinate System

The track of flight training was plotted on an assumed XY coordinate system with the center of the track circle as origin. The X-axis is perpendicular to the training airspace boundary, and the positive half of the X-axis points to the side of the route. According to the relevant provisions, the safety interval between the training airspace and the air route is 10 km. Assume that the width of the air route is 20 km. At this time, the equation of the center line of the air route is $x = R_0 + 20$ km.

4.2 Lateral Position Deviation of Civil Aviation Aircraft

In general, the yaw of the aircraft on the route is mainly caused by the navigation error, and its probability density function follows the double exponential distribution with zero expectation [5]. which is

$$f_{\text{norm}_y}(y_1) = \frac{1}{2\sigma_y}\exp\left(-\left|\frac{R_0 + 20 - y^1}{\sigma_y}\right|\right) \tag{4}$$

where σ_y is the parameter corresponding to the yaw error probability density function, and this parameter can be determined by the RNP value -n. According to the definition of RNP, n refers to 95% of the time, the aircraft is flying within the range of n nautical miles on both sides of the route centerline. According to different RNP values, the parameter σ_y under the corresponding navigation condition can be calculated. According to the calculation of Ref. [5], when the RNP value is 4, σ_y is 1.33.

4.3 Probability Model of Lateral Position Deviation of Military Aircraft

Military aircraft are often in maneuvers during training flights. And there is no statistical data on the lateral position deviation of military aircraft. Therefore, the lateral positional deviation of military aircraft in the training airspace can only be studied through mathematical models. According to the characteristics of circling flight, its lateral deviation is affected by the turning radius, the center of the turning track, and the wind.

4.3.1 Turning Radius Probability Density Function

Pilot's motion error in military training flight follow Gaussian distribution

$$\gamma(M_{\text{last}}) = \frac{1}{\sqrt{2\pi}\sigma_{\text{p}}}\exp\left(-\frac{(M_{\text{last}} - M_{\text{shlould}})^2}{2\sigma_{\text{p}}^2}\right) \tag{5}$$

where M_{shlould} is the expected motion, M_{error} is the actual motion, and σ_{p} is the standard deviation. In the process of circling training, the factors affecting the lateral deviation of the aircraft are mainly the speed of the aircraft, the heading, the turning gradient γ, the crosswind, the position of the starting point of the circling, and the navigation accuracy. As shown in Fig. 2, it is assumed that the aircraft starts to fly in the clockwise direction from the S point, and the desired circling path is as shown by the solid line, and the circling path under the influence of the action error and the omnidirectional wind is, respectively, indicated by two broken lines.

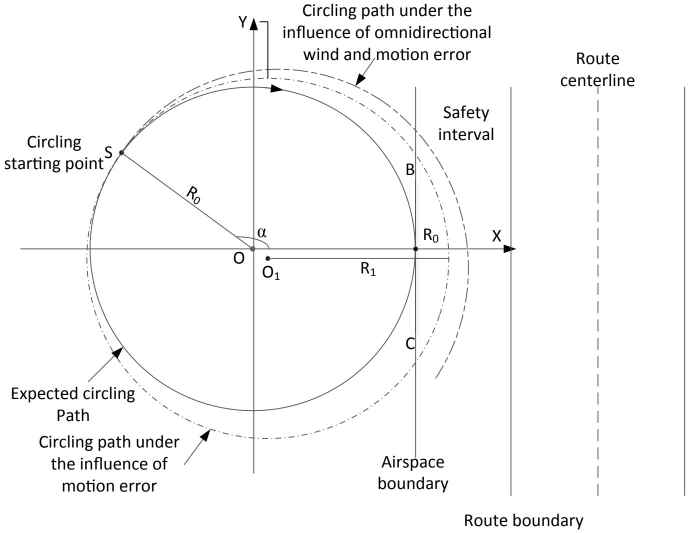


Fig. 2. The lateral spacing of the high slope circling training airspace and the route

Suppose the expected flight speed is $V = 900$ km/h, the actual speed is V', the expected turning slope is $\gamma = 45°$, and the actual turning slope is γ', both of which are affected by the pilot's motion error. Then, the turning radius under the influence of speed error and turning slope error is:

$$R_1 = \frac{(V')^2}{g\tan(\gamma')} \tag{6}$$

The actual speed and the actual turning slope obey the following probability distribution.

$$\begin{aligned} f_\gamma(\gamma') &= \frac{1}{\sqrt{2\pi}\sigma_\gamma}\exp\left(-\frac{(45-\gamma')^2}{2\sigma_\gamma^2}\right) \\ f_{\mathrm{V}}(V') &= \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{V}}}\exp\left(-\frac{(250-V')^2}{2\sigma_{\mathrm{V}}^2}\right) \end{aligned} \tag{7}$$

4.3.2 Center of the Circling Path

Suppose the aircraft starts to turn from the $S(x_{\mathrm{s}}, y_{\mathrm{s}})$ point in the heading θ, then the actual center $O_1\ (x_1, y_1)$ is

$$\begin{cases} x_1 = x_{\mathrm{s}} + R_1\cos\theta \\ y_1 = y_{\mathrm{s}} - R_1\sin\theta \end{cases} \tag{8}$$

It is assumed that the heading error is affected by navigation accuracy, airborne navigation equipment error, and pilots' operation error. All of them obey the Gauss distribution in angle. Therefore, the joint probability distribution function of heading error also follows the Gauss distribution.

$$f_\theta(\theta') = \frac{1}{\sqrt{2\pi}\sigma_\theta}\exp\left(-\frac{(\vartheta-\theta')^2}{2\sigma_\theta^2}\right) \tag{10}$$

According to the conservative values given in ICAO 8168 document, the standard deviation is theta = 2.6°, where theta is the heading angle in the ideal state and the expression is:

$$\vartheta = \begin{cases} \pi - \arctan\left(\frac{y_{\mathrm{S}}}{x_{\mathrm{S}}}\right) & x_S > 0, y_s \neq 0 \\ 2\pi - \arctan\left(\frac{y_{\mathrm{S}}}{x_{\mathrm{S}}}\right) & x_{\mathrm{S}} < 0, y_s \neq 0 \\ \frac{\pi}{2} & x_{\mathrm{S}} = 0, y_{\mathrm{S}} = R_0 \\ -\frac{\pi}{2} & x_{\mathrm{S}} = 0, y_{\mathrm{S}} = -R_0 \\ \pi & x_{\mathrm{S}} = R_0, y_{\mathrm{S}} = 0 \\ 0 & x_{\mathrm{S}} = -R_0, y_{\mathrm{s}} = 0 \end{cases} \tag{11}$$

4.3.3 Omnidirectional Wind

The plane is inevitably affected by the wind. However, a certain wind direction cannot be specified when estimating the track. Therefore, ICAO defines a wind direction that is the most unfavorable to the aircraft—omnidirectional wind. The omnidirectional wind direction is always perpendicular to the current heading of the aircraft and points in the opposite direction of the turn. Assuming that the omnidirectional wind speed is w, the real-time turning radius can be expressed as

$$R_1' = R_1 + \frac{w}{\alpha} \tag{12}$$

where α is the number of angles that the aircraft turns when it turns to a heading parallel to the airspace boundary. At this time, the military aircraft is closest to the civil aircraft. Therefore, the minimum lateral distance for defining a military and civil aircraft is

$$\Delta L = y' - x_1 + R_1' \tag{13}$$

where y' is the abscissa of the actual position of the civil aircraft.

5 Simulation of Military and Civil Aviation Lateral Deviation

There are nonlinear terms in the lateral deviation expression of military and civil aircraft, so it is difficult to obtain an analytical expression of the probability density function of the lateral spacing. Assume that each military airport has 24 fighters, each of which requires 200 h of flight training per year. There are five training airspaces in the airport area of responsibility. For each training flight, the duration of the military aircraft's activity in the airspace is half of the total flight duration. Then, the average frequency of training flights in each hourly airspace is

$$\beta = \frac{24 \times 200}{365 \times 24} \times \frac{1}{2} \times \frac{1}{5} = 0.054 \tag{14}$$

Assume that the starting point of the hover training flight is $S_2\left(-\frac{\sqrt{2}}{2}R_0, \frac{\sqrt{2}}{2}R_0\right)$. Other parameters involved in the simulation are as follows in Table 1.

Among the above parameters, the military civil aircraft size uses the public data of the F16 and A380 passenger aircraft [4]. The vertical safety interval uses a safety interval of 10 km from our radar. The Monte Carlo method is used to select the random number corresponding to the probability density function of speed and slope, and the simulation of the lateral deviation of military aircraft and civil aviation aircraft with n = 1,000,000 times is carried out. The results are shown in Figs. 3 and 4.

It is assumed that the training airspace boundary is tangent to the circling path, that is, no interval margin is left in the training airspace. The lateral deviation value of the civil aircraft obtained from the simulation is subtracted from the lateral deviation of the military aircraft. Calculate the number of simulations where the difference is less than

Table 1. Lateral collision probability simulation parameter

Parameters	Value	Parameters	Value
E (S)	0.61	λ_{mz} (m)	24.1
E (O)	0.01	λ_{jx} (m)	15.6
V (m/s)	250	λ_{jy} (m)	9.45
γ (°)	45	λ_{jz} (m)	5.09
σ_{V}	15	$U_{\mathrm{s}}(\mathrm{m/s})$	3
σ_{γ}	2	$U_{\mathrm{o}}(\mathrm{m/s})$	497
$P_{\mathrm{z}}(0)$	0.5	V_{s}or $V_{\mathrm{o}}(\mathrm{m/s})$	5.23
L (km)	10	W_{s} or W_{o} (m/s)	0.58
λ_{mx} (m)	72.8	β	0.054
λ_{my} (m)	79.8		

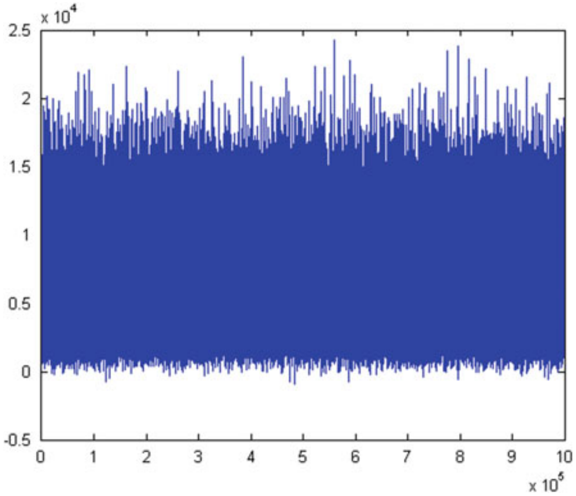


Fig. 3. Military aircraft lateral deviation

zero. From this, the frequency of the lateral separation loss of the military and civil aviation aircraft is GERh = 8.38 × 10^{-05}.

Considering the two cases of the same direction and reverse flight of military and civil aircraft, respectively, it is assumed that the aircraft with the same direction and reverse flight within 2L distance has $E(O)$ = 0.61, $E(S)$ = 0.01 pairs. Put GERh and the above parameters into Eq. 3. It is calculated that the collision probability of the same direction flight $N_{\mathrm{ay\_s}}$ is 2.58×10^{-08}, the collision probability of reverse flight $N_{\mathrm{ay\_r}}$ is 2.60 × 10^{-08}, and the total collision probability $N_{\mathrm{ay\_t}}$ is 5.18 × 10^{-08}. Take eight different circling starting points and calculate their collision probability, respectively (Table 2).

From S1 to S5, that is, the starting point of the circling clockwise from the negative X-axis direction to the positive X-axis direction, the collision probability is gradually

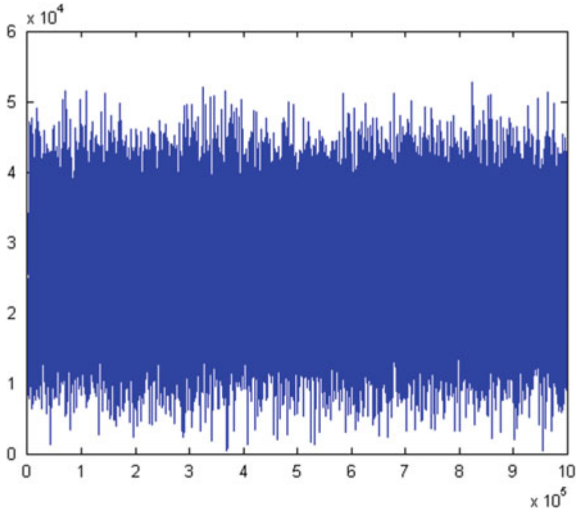


Fig. 4. Civil aircraft lateral deviation

Table 2. Military and civil aircraft collision probability without interval margin

Starting points	GERh	N_{ay_s} (Times/hour)	N_{ay_r} (Times/hour)	N_{ay_t} (Times/hour)
$S_1(-R_0, 0)$	9.56×10^{-05}	2.94×10^{-08}	2.96×10^{-08}	5.90×10^{-08}
$S_2\left(-\frac{\sqrt{2}}{2}R_0, \frac{\sqrt{2}}{2}R_0\right)$	8.38×10^{-05}	2.58×10^{-08}	2.60×10^{-08}	5.18×10^{-08}
$S_3(0, R_0)$	7.30×10^{-05}	2.25×10^{-08}	2.26×10^{-08}	4.51×10^{-08}
$S_4\left(\frac{\sqrt{2}}{2}R_0, \frac{\sqrt{2}}{2}R_0\right)$	6.34×10^{-05}	1.94×10^{-08}	1.95×10^{-08}	3.90×10^{-08}
$S_5(R_0, 0)$	5.45×10^{-05}	1.68×10^{-08}	1.69×10^{-08}	3.37×10^{-08}
$S_6\left(\frac{\sqrt{2}}{2}R_0, -\frac{\sqrt{2}}{2}R_0\right)$	1.05×10^{-04}	3.22×10^{-08}	3.24×10^{-08}	6.45×10^{-08}
$S_7(0, -R_0)$	1.25×10^{-04}	3.83×10^{-08}	3.86×10^{-08}	7.69×10^{-08}
$S_8\left(-\frac{\sqrt{2}}{2}R_0, -\frac{\sqrt{2}}{2}R_0\right)$	1.14×10^{-04}	3.51×10^{-08}	3.53×10^{-08}	7.05×10^{-08}

reduced. This is mainly because the military aircraft moves clockwise, the time to reach one side of the route is relatively short, and it is less affected by the omnidirectional wind. The deviation is mainly due to the pilot's operational error, and if the starting point of the turn is closer to the route, the less likely the military aircraft is to deviate from the training airspace due to pilot operational errors;

From *S*5 to *S*8, that is, clockwise from the positive *X*-axis direction to the negative *X*-axis direction, the collision probability gradually increases. The probability of collision is mainly affected by the pilot's operational error. But each point is affected by the omnidirectional wind. For example, points *S*3 and *S*7 at both positive and negative sides of the *Y*-axis are affected by the pilot's operational error. However, the path from the $S7$ point to the side of the route is longer, and it is also affected by the omnidirectional wind, so the probability of collision at the S7 point is high.

For the same circling starting point, when circling to the nearest position of the route, the collision probability of the military flight with the civil aviation aircraft in the same direction is lower than the collision probability of the reverse flight. The total collision probability of other circling starting points is larger than 5×10^{-9} except for all start points. Therefore, if the boundary of the circling path is regarded as the boundary of the airspace, there is a certain risk of collision. Therefore, we calculate the collision probability under various interval margin. When the ideal path and the airspace boundary have a safety margin of 7 km, the collision risk is shown in Table 3.

Table 3. Collision probability at 7 km safety margin

Starting points	GERh	N_{ay_s} (Times/hour)	N_{ay_r} (Times/hour)	N_{ay_t} (Times/hour)
$S_1(-R_0, 0)$	4.80×10^{-06}	1.48×10^{-09}	1.49×10^{-09}	2.96×10^{-09}
$S_2\left(-\frac{\sqrt{2}}{2}R_0, \frac{\sqrt{2}}{2}R_0\right)$	5.50×10^{-06}	1.69×10^{-09}	1.70×10^{-09}	3.40×10^{-09}
$S_3(0, R_0)$	5.30×10^{-06}	1.63×10^{-09}	1.64×10^{-09}	3.27×10^{-09}
$S_4\left(\frac{\sqrt{2}}{2}R_0, \frac{\sqrt{2}}{2}R_0\right)$	4.70×10^{-06}	1.44×10^{-09}	1.45×10^{-09}	2.89×10^{-09}
$S_5(R_0, 0)$	3.60×10^{-06}	1.11×10^{-09}	1.12×10^{-09}	2.22×10^{-09}
$S_6\left(\frac{\sqrt{2}}{2}R_0, -\frac{\sqrt{2}}{2}R_0\right)$	5.10×10^{-06}	1.57×10^{-09}	1.58×10^{-09}	3.15×10^{-09}
$S_7(0, -R_0)$	5.70×10^{-06}	1.75×10^{-09}	1.77×10^{-09}	3.52×10^{-09}
$S_8\left(-\frac{\sqrt{2}}{2}R_0, -\frac{\sqrt{2}}{2}R_0\right)$	5.40×10^{-06}	1.66×10^{-09}	1.67×10^{-09}	3.33×10^{-09}

6 Summary

This paper studies the collision risk assessment of military and civil aircraft. The Event model has been improved to assess the probability of collision between a military training aircraft and a civil aircraft. Taking the circling training flight as an example, the simulation study shows that the collision risk of the military and civil aircraft is related to the starting circling position and the circling direction of the military aircraft. When there is no interval margin in the training airspace, the collision risk of military and civil aircraft is larger than the standard requirement. According to the simulation results, the collision risk of military and civil aircraft can be reduced in the following ways without improving the navigation accuracy and the pilot's ability.

1. In order to reduce the risk of collision, the starting point of the circling training should be set to be close to the side of the civil air route and circling in the opposite direction of the route.
2. The heading of the circling flight shall be determined according to the heading of the civil aircraft at the level. When the military aircraft is closest to the air route, its heading should be the same as that of the civil aviation aircraft.

3. According to the most demanding conditions, the collision risk can be met when the ideal circling path maintains a 7 km margin with the airspace boundary close to the air route. In the other directions, since it is not adjacent to the air route, the interval margin does not need to be too large. According to the calculation of the interval margin of 7 km in the approaching route and 2 km in other directions, the circling training airspace size can be set to 22 km × 17 km.

References

1. Reich PG (1966) Analysis of long range air traffic systems: separation standards. J Inst Navig 19:331–347
2. Brooker P (2003) Lateral collision risk in air traffic systems: a "Post-Reich" event model. J Navig 56:399–409
3. Meng X, Zhang P et al (2010) Aircraft collision risk assessment at intersecting air routes. J Beijing Univ Aeronaut Astronaut 36:1–5 (in Chinese)
4. Cao X, Zhang Z (2015) Cross route collision risk assessment based on improved Event model. J Civ Aviat Univ China 33:1–4 (in Chinese)
5. Zhang X, Zhu D, Yang C (2014) Safety assessment of RNP4 Chengdu-Lasa parallel routes. Sci Technol Eng 14:307–310 (in Chinese)

Imbalanced Data Classification with Deep Support Vector Machines

Li Zhang, Wei Wang(✉), Mengjun Zhang, and Zhixiong Wang

Tianjin Key Laboratory of Wireless Mobile Communications and Power
Transmission, Tianjin Normal University, Tianjin 300387, China
weiwang@tjnu.edu.cn

Abstract. In recent years, deep learning has become increasingly popular in various fields. However, the performance of deep learning on imbalanced data has not been examined. The imbalanced data is a special problem in target detection and classification task, where the number of one class is less than the other classes. This paper focuses on evaluating the performance of the deep support vector machine (DSVM) algorithm in dealing with imbalanced human target detection datasets. Furthermore, we optimize the parameters of the DSVM algorithm to obtain better detection performance. It is compared with the stacked auto-encoder (SAE) and the support vector machine (SVM) algorithm. Finally, numerical experimental results show that the DSVM algorithm can effectively capture the minority class.

Keywords: Deep support vector machines · Imbalanced data · Human target detection

1 Introduction

The skewed distribution of data samples between different classes is a common phenomenon in many real-world classification issues, such as spam review detection [1], insurance fraud detection [2], web author identification [3], wilt disease classification [4] and multimedia concept detection [5]. In this paper, we focus on binary classes classification issues for imbalanced datasets, where the classes that contain a small number of instances are named the minority class, while another dominant instance space is named the majority class. The imbalanced datasets have reduced the performance of existing learning methods and posed a relatively new challenge to them for the hardness to learn the minority instance.

Confronted with the issue of imbalanced learning, plenty of approaches have been proposed to solve the imbalanced distribution. The existing methods can be mainly divided into two categories, one is the data processing level, and the other is the classification algorithms level. Sampling strategies are often used at the data processing level to provide a balanced class distribution. In [6], a new over-sampling technique called Density-Based Synthetic Minority Over-sampling Technique (DBSMOTE) was proposed, and the method was designed to over-sample an arbitrarily shaped cluster. He et al. [7] applied two ways for reducing the bias and shifting the classification decision boundary by the Adaptive Synthetic Sampling approach. In [8], two kinds of

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 87–95, 2020.
https://doi.org/10.1007/978-981-15-0187-6_10

over-sampling methods, namely border-SMOTE1 and border-SMOTE2 were given, and the method gets better True Positive (TP) rate and F-value. Although the data processing level has achieved some satisfactory results for imbalanced classification, they still have their drawbacks. It is easy to lose significant information or add trivial information, so as to affect the recognition accuracy of the minority.

At the same time, many machine learning methods have been focused on detailed techniques to cope with imbalanced data classification. Datta et al. [9] proposed the approach called near-Bayesian support vector machine (NBSVM) to adapt for cases with the skewed distribution misclassification. Support vector machine (SVM), as the basic classifiers of the ensemble committee, improved the accuracy of the hyperspectral remote sensing images imbalanced classification [10]. Yan et al. [5] used an extended bootstrapping and convolutional neural networks method to handle the skewed multimedia datasets. The results showed that the framework can work effectively and greatly reduce the running time. These papers provide us with a new idea of whether shallow models can be turned into depth architectures.

In this paper, the deep support vector machine (DSVM) algorithm has shown excellent performance in regression, classification and dimension reduction [11, 12]. Hence, we took into account the DSVM algorithm for imbalanced data classification. To the best of our knowledge, the DSVM algorithm is the first time to be applied to imbalanced human target detection. Furthermore, we fine-tune the DSVM algorithm to obtain promising performance of imbalanced human target detection.

The rest of this paper is organized as follows. Section 2 introduces the DSVM algorithm. In Sect. 3, we describe the experimental process and results in detail. It is also compared with SVM and the SAE algorithm. Numerical experimental results show that the DSVM algorithm performs better. In the end, Sect. 4 summarizes this paper and points out possible directions for future work.

2 The Deep Support Vector Machine Algorithm

Wiering et al. [11] proposed the DSVM algorithm, which replaces neurons as SVM and has depth models. The DSVM algorithm is the application of the deep learning model to SVM. First, it trains SVM in a standard way. Secondly, the kernel activation function of the support vector is used as input data into the SVMs of the hidden layer. Finally, the main SVM is trained to establish a nonlinear combination of the kernel activation function of the support vector. The module of a two-layer DSVM is shown in Fig. 1.

2.1 The DSVM Algorithm Model

We choose a classification dataset: $\{x_n \in \mathbb{R}^m, y_n \in \mathbb{R}^s\}_{n=1}^{N}$, where x_n is input vector and y_n is the scalar target output. The connection between the input vector x_n and the output target y_n is

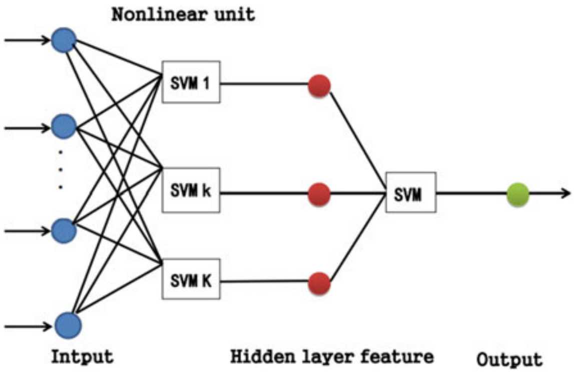


Fig. 1. Module of a two-layer DSVM

$$\begin{cases} h = (h_1, h_2, \ldots, h_K) \\ h_k = \sum_{n=1}^{N} \alpha_{k,n} \kappa(x_n, x) + b_k \end{cases} \tag{1}$$

where $\kappa(x_n, x)$ is a kernel function, the parameters $\alpha_{k,n}$ and b_k are obtained by the SVM optimization objective function. It is worth noting that $h_k (k = 1, 2, \ldots, K)$ is not a one dimensional by the structure of the algorithm and the specific formula and can be defined. Next, the hidden layer feature h is taken as the input to the next nonlinear unit (SVM unit), namely:

$$\begin{cases} y = \sum_{n=1}^{N} \beta_n \cdot \kappa(h_n, h) + c \\ h_n = \left[h_{1,n}, h_{2,n}, \ldots, h_{K,n}\right] \\ h = [h_1, h_2, \ldots, h_K] \end{cases} \tag{2}$$

where the parameters β_n and c need to be learned, h_n is the hidden layer feature of the input x_n, and h is the hidden layer feature of the input x.

2.2 Optimization Objective Function of the DSVM Algorithm

The optimization objective function is

$$\min_{\theta} J(\theta) = \frac{1}{N} \sum_{n=1}^{N} \|\hat{y}_n - y_n\|_2^2 + \lambda R(\theta) \tag{3}$$

where $\lambda R(\theta)$ is the regularization term symbol and

$$\begin{cases} \theta = (a, b; \beta, c) \\ \alpha = \{\alpha_{k,n}\}_{n,k=1}^{N,K}; b = \{b_k\}_{k=1}^{K} \end{cases} \tag{4}$$

2.3 The DSVM Algorithm Solving Process

The gradient descent method is used to realize the optimization of parameters. The core of the error propagation is to solve the partial derivative. If the network structure in Fig. 1. contains only one hidden layer, the error propagation item will be:

$$\delta = \frac{\partial J(\theta)}{\partial h} = \left(\frac{\partial J(\theta)}{\partial h_1}, \frac{\partial J(\theta)}{\partial h_2}, \ldots, \frac{\partial J(\theta)}{\partial h_K} \right) \tag{5}$$

Then, the chain rule is used to update the layer-by-layer parameters.

To adjust for the class imbalance, the process pipeline of this paper is shown in Fig. 2. Specifically, we filter the raw data. Then, the feature extraction and normalization processing are carried out, the purpose of which is to improve the accuracy of prediction and speed up the network learning. Next, the processed data is divided into training samples and test samples, in which the training samples are used to train the parameters in the model and test samples for testing. Finally, the sum of the correct test samples divided by the total number of test samples is evaluated as the output result.

3 Imbalanced Data Classification Experiment

In this section, we first construct the experimental system to obtain the human target detection datasets. Then, we introduce the data selection and processing. Finally, evaluate the performance of the DSVM algorithm on imbalanced human target detection datasets.

3.1 Radar Measurement System and Experimental Implementation Details

To evaluate the performance of the DSVM algorithm in imbalanced datasets application, we built a human target detection system to obtain data for verification, as shown in Fig. 3. In the experiment, we used the P410 MRM radar device. The device is a single-base station radar platform, with small size, low power consumption, affordable and other characteristics, can provide a central frequency of 4.3 GHz. The module of P410 MRM is shown in Fig. 4. The experimental environment belongs to the indoor, and the experimental wall is a brick wall. The experimental scene requires that the human target is 100 cm away from the brick wall. The radar equipment is 20 cm from the brick wall. The equipment is placed on a tripod with a height of 120 cm from the

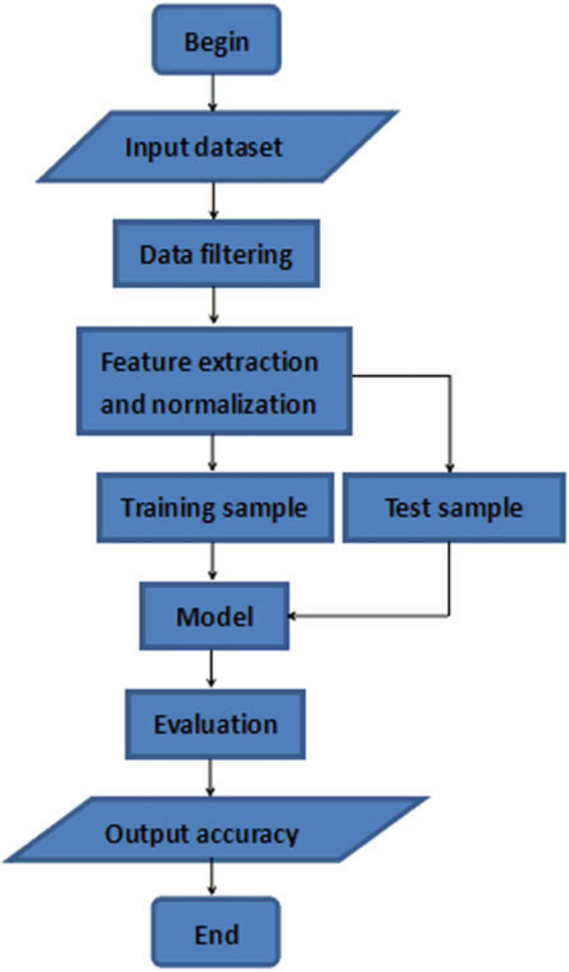


Fig. 2. Classification experiment flow chart

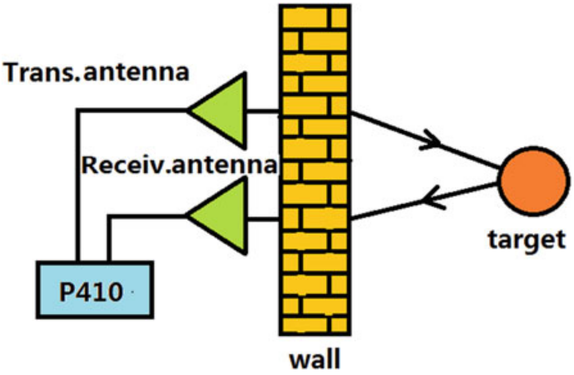


Fig. 3. Target detection system

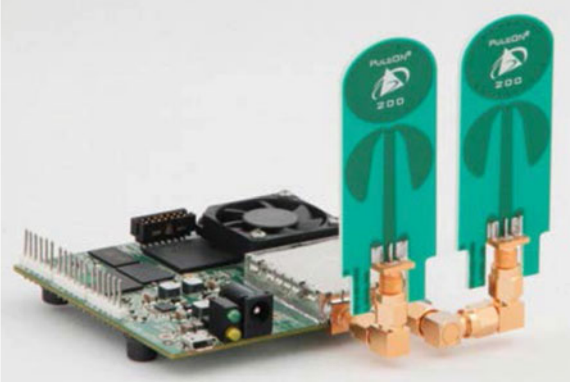


Fig. 4. Radar module: P410 MRM

ground. And the wall thickness is 24 cm. To ensure that the electromagnetic wave can be injected vertically into the wall, the receiving antenna and the transmitting antenna are set to face the wall.

In this case, experimental analysis of multistate human targets is carried out. The first group of experiments was unmanned state and the one-person rapid breathing state after the wall. The second group of experiments was the unmanned state and the two-person slowly walking state behind the wall. All human targets are required to face the radar system behind the wall.

3.2 Human Target Detection Dataset Descriptions

By constructing the experimental platform described above, we collected three groups data samples. In order to evaluate the performance of the DSVM algorithm for human target detection through the wall, we combined four groups of datasets, namely N200S20, N200Q20, N200S40 and N200Q40. Among them, the letter N represents unmanned behind the brick ball. The letter S stands for two-person slowly walking state behind the wall. The letter Q represents one-person rapid breathing state after the wall.

Specifically, the N200S20 dataset contains 220 samples where 200 samples are with nobody behind the brick wall and 20 samples are with two persons keeping the status of slow moving. The N200Q20 dataset contains 220 samples where 200 samples are with nobody behind the brick wall and 20 samples are with one-person rapid breathing state behind the wall. The N200S40 dataset and the N200Q40 dataset are similar to the above expression, except that the number of human states is changed to 40. After feature extraction, the number of feature attribute values for these four datasets is 34.

In this paper, the experimental data is normalized to meet the requirements of the experiment and reduce the amount of calculation. The training sample is a randomly selected 90% in each type of dataset. The selection of test samples is determined according to the human states in the training sample. For the training datasets, where the number of human states after the wall is 20, the test sample selects another dataset in that state, with a number of 10 and 20. For the training datasets, where the number of human states after the wall is 40, the test sample selects another dataset in that state, with a number of 10, 20 and 40. Therefore, the test samples for these four different datasets are S10, S20, S40 and Q10, Q20, Q40. Figure 5 shows the experimental example with the N200Q40 dataset. The datasets named Q10, Q20 and Q40 represent the test samples.

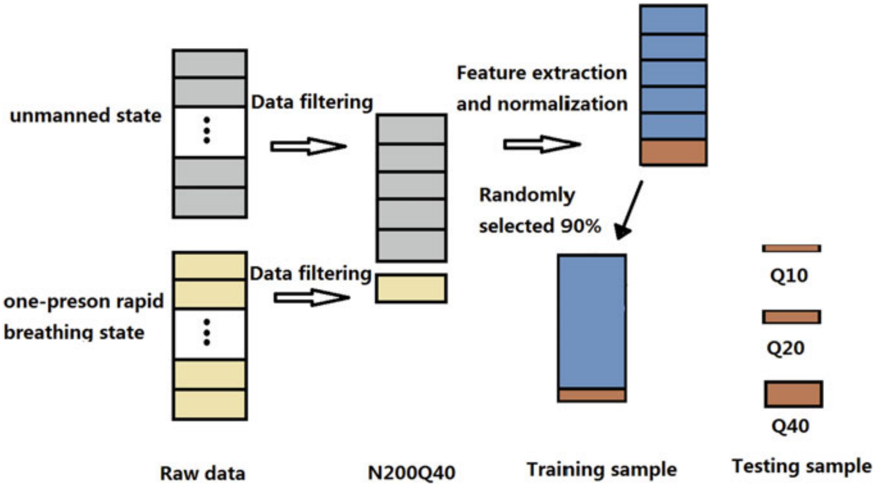


Fig. 5. Experimental example with the N200Q40 dataset

3.3 Experimental Analysis of the DSVM Algorithm

The DSVM used in this paper has a large architecture. Its complexity is linearly extended with the number of standard support vector machines, and the strong regularization of the main SVM prevents over-fitting. For the processing of hidden layer SVM, we use backpropagation-like technology to build a new dataset. And the target value of the hidden layer SVM needs to be limited between −1 and 1. In order for the hidden layer to extract different features, symmetric destruction is required.

In the experiment, we used radial base function (RBF) kernels in two layers of the DSVM because the experimental results obtained from other commonly used kernels were poor. There are two kernel functions: $k_1(x_n, x)$ is the kernel function of the hidden layer SVMs, and $k_2(x_n, x)$ is the kernel function of the output layer of DSVM. In the DSVM algorithm, alternating training between the main SVM and the hidden layer SVMs requires multiple periods of execution. We take the particle swarm optimization (PSO) algorithm to realize the re-optimization of the meta-ancestor parameters.

3.4 Algorithm Results Comparison

In order to evaluate the classification efficiency of the DSVM algorithm on the imbalanced datasets, we performed five classification evaluations in each set of instances on each dataset. Finally, the average result is taken as the classification accuracy of the target. The classification recognition rate is compared with SAE and SVM algorithm. The compared results are given in Table 1. Experimental results show that the DSVM algorithm improves the accuracy of the minority instances. The reason is that the output layer support vector machine has strong regularization ability, which makes the system difficult to over-fitting.

Table 1. Experimental results of DSVM, SAE and SVM under the same conditions

Data sets	Test sample	DSVM	SAE	SVM
		Accuracy (%)		
N200S20	S10	86.823	0	0
	S20	86.92	0	50
N200S40	S10	97.426	80	0
	S20	97.772	90	50
	S40	97.8825	95	82.5
N200Q20	Q10	90.782	0	0
	Q20	90.868	0	0
N200Q40	Q10	94.67	72.5	90
	Q20	94.818	80	90
	Q40	95.1005	85	95

4 Conclusion and Future Work

The DSVM algorithm is relatively flexible when adjusting the kernel function, and the selected RBF kernel function can achieve good results. Second, it is easier to implement with gradient ascent algorithm and backpropagation-like technology. Finally, the strong regularization capability of the output layer SVM makes it difficult for the classification system to overfit.

Considering these advantages, we applied it to the through-wall human target detection of imbalanced data sets and compared with the algorithm of SAE and SVM. Experimental results show that the DSVM algorithm can improve the classification performance of minority instances. However, the method has a high computational complexity.

Therefore, we should focus on improving the DSVM architecture, such as replacing the standard SVM with a least-squares support vector machine (LS-SVM) as a unit, trying new parameter optimization methods or ensemble learning to solve the skewed data distribution.

Acknowledgements. This paper is supported by Natural Youth Science Foundation of China (61501326), the National Natural Science Foundation of China (61731006).

References

1. Al Najada H, Zhu X (2014) ISRD: spam review detection with imbalanced data distributions. In: IEEE international conference on information reuse and integration, Redwood City, CA, USA, pp 553–560
2. Hassan AK, Abraham A (2016) Modeling insurance fraud detection using imbalanced data classification. In: Advances in nature and biologically inspired computing, pp 117–127
3. Vorobeva A (2016) Examining the performance of classification algorithms for imbalanced data sets in web author identification. In: Proceedings of the 18th conference of open innovations association FRUCT, pp 385–390
4. Mohd Pozi MS, Sulaiman MN, Mustapha N, Perumal T (2015) A new classification model for a class imbalanced data set using genetic programming and support vector machines: case study for wilt disease classification. Remote Sens Lett 6(7):568–577
5. Yan Y, Chen M, Shyu ML, Chen SC (2015) Deep learning for imbalanced multimedia data classification. In: 2015 IEEE international symposium on multimedia (ISM), pp 483–488
6. Bunkhumpornpat C, Sinapiromsaran K, Lursinsap C (2012) DBSMOTE: density-based synthetic minority over-sampling technique. Appl Intell 36(3):664–684
7. He H, Bai Y, Garcia EA, Li S (2008) ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence), pp 1322–1328
8. Han H, Wang WY, Mao BH (2005) Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: International conference on intelligent computing, pp 878–887
9. Datta S, Das S (2015) Near-Bayesian support vector machines for imbalanced data classification with equal or unequal misclassification costs. Neural Netw 70:39–52
10. Eeti LN, Buddhiraju KM (2018) Classification of hyperspectral remote sensing Images by an ensemble of support vector machines under imbalanced data. In: IGARSS 2018-2018 IEEE international geoscience and remote sensing symposium, pp 2659–2661
11. Wiering MA, Schutten M, Millea A, Meijster A, Schomaker LRB (2013) Deep support vector machines for regression problems. In: International workshop on advances in regularization, optimization, kernel methods, and support vector machines, pp 53–54
12. Wiering MA, Schomaker LRB (2014) Multi-layer support vector machines. In: Regularization optimization kernels and support vector machines. CRC Press, Boca Raton, pp 457–475

An Incident Identification Method Based on Improved RCNN

Han He, Haijun Zhang(✉), Sheng Lv, and Bingcai Chen

School of Computer Science and Technology, Xinjiang Normal University,
Urumqi 830054, China
zhjlp@163.com

Abstract. An emergency is a sudden and harmful event. It is of great significance to quickly identify the event and reduce the harm caused by the event. In this paper, the current advanced recurrent convolutional neural networks (RCNN) are utilized, but the traditional model cannot effectively identify the event, and the accuracy rate is not good enough. In order to solve this problem, the recurrent neural network and activation function part of the traditional model are improved, and through experimental comparison, the optimal model in the training model is selected. Finally, the accuracy of the model is 90%, the recall rate is 92.55%, and the F1 value, a metric that combines accuracy and recall, is 91.26%, which proves that the improved model has good effects.

Keywords: RCNN · LSTM · Event recognition · RELU · Deep learning

1 Introduction

China, a populous country, has experienced frequent emergencies. Since 2009, China has identified the detection of emergencies as one of the key research projects. Emergency [1] is an event that has no symptoms but causes harm to society and people, and it is divided into four types: natural disasters, accident disasters, public health events, and social security events [2]. The location, time, type, etc. of an emergency are often beyond people's imagination and generally occur when people are not prepared. In this case, it is precisely because of this urgency and uncertainty that China hopes to identify emergencies as early as possible through relevant research and take measures to minimize casualties.

In recent years, the technology of event recognition based on deep learning has attracted the attention of many scholars. In this paper, the recurrent convolutional neural network in deep learning is studied, and an improved model is proposed to solve the non-convergence of the model training process and to increase the recognition accuracy of the model.

The remainder of this paper is arranged as follows: the second section introduces the related works about event recognition; the third section describes the principle of the revised model on the traditional RCNN; the fourth part gives experiments and discussions on the proposed method. In the end, fifth section presents the conclusion and future works.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 96–104, 2020.
https://doi.org/10.1007/978-981-15-0187-6_11

2 Related Works

So far, there are two main methods of event recognition, one is the rule-based method to identify the event and the other is to extract the event through machine learning.

The rule-based identification method is to identify the type of event by manually defining rules for the text, creating a template and matching the text. Yankova and Boytcheva [3] and Rainer [4] use their knowledge and logical form representations in different fields to infer facts and template fills. Yangarber [5] takes advantage of regular expressions and an association map from syntactic to logical form for event extraction. However, this method requires a large amount of manpower and related experts to analyze them, and the accuracy of the extraction is not high.

Research scholars are also concerned about the problems brought by the rules. With the extensive use of machine learning, more and more people combine event recognition with machine learning [6]. For example, Ahn [7] has used MegaM and TiMBL, two kinds of machine learning methods have been used for event extraction and have achieved good results in English corpus; Comparing with independent learning classifiers, Vlachos [8] used structured predictive learning model, which based on searching. As the core of artificial intelligence, machine learning improves the performance of the algorithm itself through experience learning and solves the shortcomings of requiring a large amount of manpower in the rule-based way.

As a branch of machine learning, deep learning [9] is one of the most popular machine learnings. In 2014, Kim [10] proposed a convolutional neural network which achieved good results in text classification; Su et al. [11] combines two trained RNN models with a dictionary to obtain the best recognition results. In this paper, we will use the RCNN [12] that combines the CNN with the RNN, which not only contact the context, but also solve the shortcomings of data bias in the RNN.

3 Improved RCNN Incident Identification Method

3.1 The Principle of Traditional RCNN

The traditional RCNN model uses the structure of the bidirectional RNN + tanh activation function + pooling layer, as shown in Fig. 1.

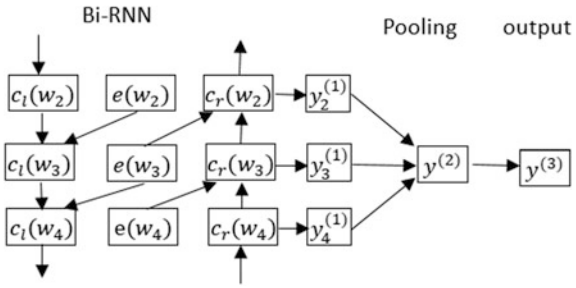


Fig. 1. RCNN working principle

In this paper, the input of a word is represented by the word and its context $c_l(w_i)$ represents the text to the left of the word w_i, and $c_r(w_i)$ represents the text to the right of the word w_i, then each word in the article is represented by x_i:

$$x_i = [c_l(w_i); e(w_2); c_r(w_i)] \tag{1}$$

Next, we will make a linear transformation of the input of the word and generate $y_i^{(1)}$ by the activation function to pass to the largest pooling layer to try to find the most important latent semantic factor, producing $y^{(2)}$.

$$y_i^{(1)} = \tanh\left(W^{(1)}x_i + b^{(1)}\right) \tag{2}$$

$$y^{(2)} = \max_{i=1}^{n} y_i^{(1)} \tag{3}$$

In the output layer, $y^{(2)}$ generated in the pooled layer is used as the input value of the output layer, then the output of each neuron in the output layer is expressed as:

$$y^{(3)} = f\left(W^{(3)}y^{(2)} + b^{(3)}\right) \tag{4}$$

where $y^{(2)}$ represents the input of the neuron, $W^{(3)}$ represents the weight, $b^{(3)}$ represents the offset value, and f represents the activation function. The output value $y^{(3)}$ is sent to the softmax classifier for probability distribution, and finally, the event recognition is completed. The formula is as follows:

$$p_i = \frac{\exp\left(y_i^{(3)}\right)}{\sum_{k=1}^{n} \exp\left(y_k^{(3)}\right)} \tag{5}$$

3.2 Model Improvement

Firstly, by training and testing the traditional RCNN, the accuracy is only 53.5%. This result is far from enough. The article will improve the model from the following two parts.

(1) Change of bidirectional RNN

As an improvement to the RNN, LSTM adds three control units to the traditional model: input gates, output gates, and forgetting gates. As the information enters the model, the units in the LSTM will judge the information, and the information that conforms to the rules will be left, and the non-compliant information will be forgotten. This method solves the long sequence dependency problem in the RNN model.

Therefore, this paper converts the bidirectional RNN of the traditional RCNN into a bidirectional LSTM, and constructs a structure of bidirectional LSTM + tanh activation function + pooling layer, but the accuracy of the model test is only 57%.

In order to solve this situation, this paper observes the model training through tensorboard.

As can be seen from Fig. 2, as the number of training increases, the training accuracy of the model can not reach the stable value, the accuracy of the training is fluctuating and the amplitude of the shock is large. This situation represents the appropriate model parameters that were not found during the training of the model, resulting in the model not converging. In order to solve this situation, the activation function will be changed.

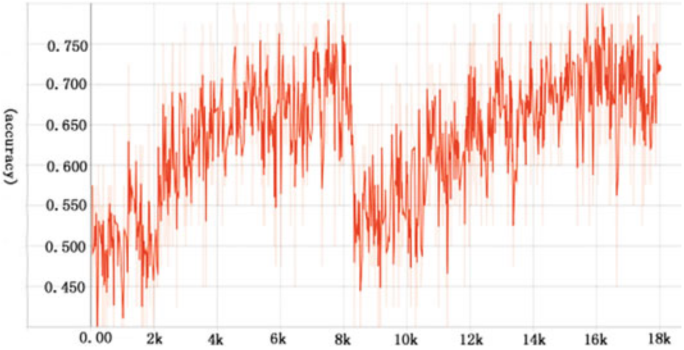


Fig. 2. Train situation of use Bi-LSTM + tanh + pooling

(2) Change of activation function

In 2003, neuroscientists discovered that during the process of dealing with a thing, the activated neurons actually accounted for only 1–4% of all neurons. Different neurons have different responsibilities and perform their duties. Due to the relu function that can simulate sparsity, this paper replaces the tanh activation function with the relu activation function after the bidirectional recurrent structure, which reduces the interdependence between parameters and slows down the problem that the model does not converge. We can observe the model training situation through tensorboard.

It can be seen from Fig. 3 that as the number of training increases, the accuracy of the model is continuously strengthened and the amplitude of the oscillation is continuously reduced, which solves the problem that the model does not converge.

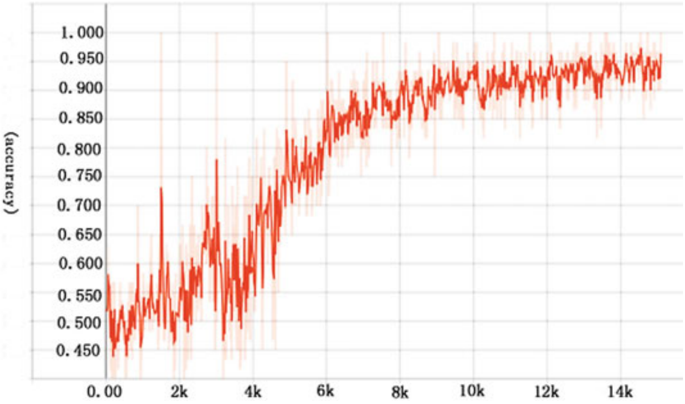


Fig. 3. Train situation of use Bi-LSTM + relu + pooling

This shows that the improved model is a good recognition model, and then the model will be tested and analyzed.

4 Experiment and Discussion

4.1 Data Set Construction and Pretreatment

The data mainly comes from the CEC corpus, the THUCNews dataset, and the headlines of today's headlines. First, the collected data needs to be manually classified according to the characteristics of the emergencies, and an emergency data set and a non-incident event data set are established.

The most difficult part of the manual classification process is the construction of the emergency set. In order to solve this problem, we construct a trigger vocabulary based on the types and characteristics of the events and help to construct the emergency data set by trigger words. The process is as in Fig. 4.

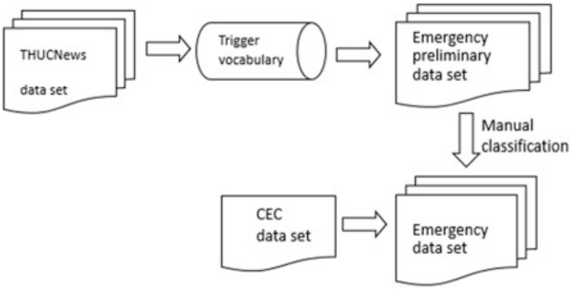


Fig. 4. Construction of emergency data sets

According to this method, a total of 6000 data were obtained, including 2000 for emergencies and 4000 for non-emergency events. All data must first be segmented, then the stop words are removed in the article, and finally, the words are converted into word vectors as algorithm input.

4.2 Evaluation Criterion

This paper adopts accuracy (A), recall (R), and F1 as comprehensive evaluation indicators. The calculation formula for each evaluation index is as follows:

$$A = \frac{\text{Correct number of classifications}}{\text{Actual number of classifications}} \times 100\% \tag{6}$$

$$R = \frac{\text{Correct positive sample classification}}{\text{Positive sample actual number}} \times 100\% \tag{7}$$

$$\mathrm{F1} = \frac{2 \times A \times R}{A + R} \times 100\% \tag{8}$$

4.3 Analysis and Discussion

In order to analyze the effectiveness of the model, the following three aspects will be compared and the model will be improved.

(1) Model performance verification

The traditional RCNN model is composed of bidirectional RNN + tanh + pooling. In the experiment, we represent this model by RT, and the paper improves it into a bidirectional LSTM + relu + pooling model, which is represented by LR.

The improved model of traditional RNN has LSTM model and GRU model. The activation function commonly uses tanh and relu activation functions. Then we construct different models, such as bidirectional RNN + relu + pooling, represented by RR; bidirectional LSTM + tanh + pooling, represented by LT; bidirectional GRU + tanh + pooling, represented by GT; bidirectional GRU + relu + pooling, represented by GR.

In the experiment, 2000 emergency data and 2000 non-incident data were used to train the model, and 200 data were used as the test set. The test was performed on different models. The results are as follows.

It can be seen from Table 1 that when the LR model is adopted, the recognition performance preferably obtains a higher accuracy. In the RCNN model, when the LSTM and relu activation functions are combined, the model can be trained to find the most suitable parameters.

Table 1. Different model performance verification results

Model	Accuracy (%)	Recall (%)	F1 (%)
RT	53.50	52.29	52.89
RR	57.50	63.64	60.41
GT	75.50	68.35	71.74
GR	77	70.45	73.60
LT	57	64	60.30
LR	83.50	77.24	80.25

(2) The impact of data set ratio on event recognition

This experiment used 4000 data as the training set, 200 data as the test set, and the ratio of the amount of emergency data to non-emergency data was 1:1.

There are two important indicators for the model test, namely TP and TN. TP stands for the numbers that test and true values are both positive. TN represents the numbers that test value is negative but true value is positive. In this test result, the number of TP is 95, but the number of TN is only 72, then it is suspected that the effect of the model is related to the ratio and number of training data sets. Because the amount of negative sample data for training is too small, the characteristics of the model for negative samples are not completely extracted. Then we increase the number of training for negative samples by 1000 each time. The test results are as follows.

Table 2 shows that when the data ratio is 2:3, the accuracy rate reaches 90%, and when the positive and negative ratios are too large, the feature extraction will be disordered, resulting in a decrease in accuracy.

Table 2. Comparison of different data ratios

Data ratio	Accuracy (%)	Recall (%)	F1 (%)
1:1	83.5	77.24	80.25
2:3	90	92.55	91.26
1:2	88.5	88.89	88.69

(3) The influence of word vector dimension on event recognition

The dimension of the word vector represents the characteristics of the word. Generally speaking, the more features, the more accurately distinguish the word from the word, but as the dimension increases, the relationship between words will also fade. Too high dimension will fade the relationship between words. If the dimension is too low, the word can not be distinguished, so the choice of the dimension of the word vector depends on your actual application scenario.

Here, the dimensions of the word vector are 50, 70, 90, 100, 110, 130, and 150. The accuracy, recall, and F1 values are as in Fig. 5.

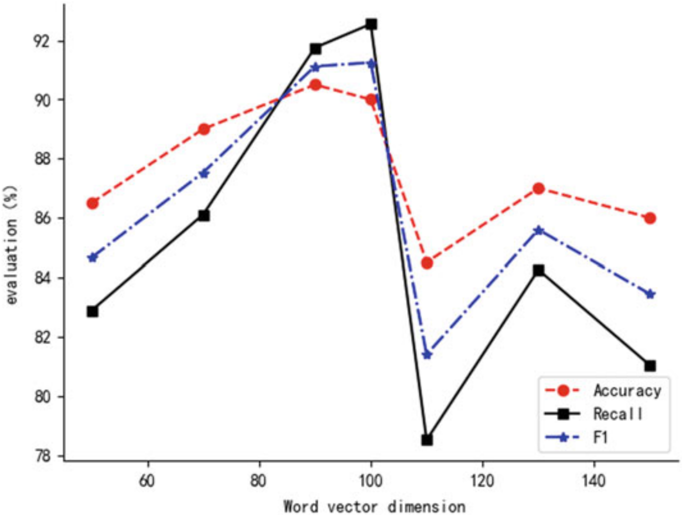


Fig. 5. Changes in the results of different word vector dimensions

As can be seen from the line graph above, when the word vector dimension is 100, the effect is the best, the accuracy rate is 90%, the recall rate is 92.55%, and the F1 value is 91.26%. When the dimension is too high, the generalization ability of the model to the feature data is reduced, resulting in a decline in model performance.

4.4 Comparison with Other Methods

In order to verify the validity of the model, this paper will compare with the more advanced models; the results are shown in Table 3.

Table 3. Experimental comparison results of different models

Model	Accuracy (%)	Recall (%)	F1 (%)
Bi-LSTM	78.6	76.63	77.6
CNN	86.5	88.42	87.45
RCNN	90	92.55	91.26

In this paper, the RCNN is used. From Table 3, the model has higher values in accuracy, recall, and F1 than other models, which proves that the model used in this paper has a good recognition effect.

5 Conclusion and Future Works

In this paper, the RCNN model is used to identify emergencies. Due to the shortcomings of traditional RCNN models in event recognition, this paper tries to improve and optimize the traditional models by combining different RNN models and activation functions. At the same time, we adjust the data scale and word vector dimension to get the best model.

The next step will accept a strategy to increase the speed of training, reduce the time of training, and combine with web crawlers to create an online emergency identification system to provide an external user with a continuous and reliable service.

Acknowledgements. This work is supported by Xinjiang Joint Fund of National Science Fund of China (U1703261). We are grateful to Shanghai University, Sina News for their provided open resources.

References

1. Phillips BD, Neal DM, Webb G (2016) Introduction to emergency management. CRC Press
2. Qin X, Zu-Jun M, Hua-Jun L (2008) Location-routing problem in emergency logistics for public emergencies. J Huazhong Univ Sci Technol
3. Yankova M, Boytcheva S (2003) Focusing on scenario recognition in information extraction. In: Tenth conference on European chapter of the association for computational linguistics. Association for Computational Linguistics
4. Rainer (2007) Modeling treatment processes using information extraction. Biochem Soc Trans 38(6):1581–1586
5. Yangarber R (2001) Scenario customization for information extraction. Dissertations & Theses
6. Abadi M, Barham P, Chen J et al (2016) TensorFlow: a system for large-scale machine learning
7. Ahn DD (2006) The stages of event extraction. Workshop on annotating & reasoning about time & events. Association for Computational Linguistics
8. Vlachos A, Craven M (2012) Biomedical event extraction from abstracts and full papers using search-based structured prediction. BMC Bioinform 13(11 suppl):S5
9. Lecun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
10. Kim Y (2014) Convolutional neural networks for sentence classification. Eprint Arxiv
11. Su B, Zhang X, Lu S et al (2015) Segmented handwritten text recognition with recurrent neural network classifiers. In: 2015 13th international conference on document analysis and recognition (ICDAR). IEEE Computer Society
12. Lai S, Xu L, Liu K et al (2015) Recurrent convolutional neural networks for text classification. In: Twenty-ninth AAAI conference on artificial intelligence

Latency Estimation of Big Data Processing Under the MapReduce Framework with Coupling Effects

Di Lin(✉), Lingshuang Cai, Xiaofeng Zhang, Xiao Zhang, and Jiazhi Huo

School of Information and Software Engineering,
University of Electronic Science and Technology of China, Chengdu, China
lindi@uestc.edu.cn

Abstract. MapReduce is a model of processing large-scaled data with parallel and distributed algorithms at a cluster, and it is composed of two stages: a map stage for filtering and sorting data and a reduce stage for the operation of summary. We develop a model with two connected queues: one upstream queue for the data flow to access the mappers and one downstream queue for the data flow to access the reducers. Also, we analyze the latency of processing a large scale of data using queueing models, in consideration of the coupling effects between these two queues for map and reduce, respectively. Our analysis results on various datasets and with various algorithms show that the MapReduce framework can almost linearly speed up with increasingly more processors, and adding mappers is usually more efficient than adding reducers to reduce the latency when processing a large-scaled dataset.

Keywords: MapReduce · Coupling effects · Latency estimation · Big data processing

1 Introduction

With the advances of mobile devices, sensing devices, social media, and Web technologies, the amount of data is dramatically increasing at an unexpected rate. In the era of big data, machine learning-based data analytics are viewed as the primary driver of the big data revolution by extracting the insights of data. A few algorithms have been adjusted to fit the machine learning algorithms for large datasets, such as the algorithms adapted to the paradigm of MapReduce [1].

Quite a few studies have addressed the challenges and solutions in the design of machine learning algorithms for big data processing under the MapReduce framework. Big data process violates the assumption of loading the entire data into memory or into a single file or disk at the training stage, and the operations of several machine learning algorithms rely on this assumption. The violation of

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 105–112, 2020.
https://doi.org/10.1007/978-981-15-0187-6_12

such an assumption when processing big data under the MapReduce framework refers to the curse of modularity [2]. Kune et al. in [3] adjust a few algorithms to fit the MapReduce framework, including Naive Bayes algorithms, k-means algorithms, neural network algorithms, support vector machine algorithms, and show that these algorithms can reduce the computation latency using the framework of MapReduce. However, a few algorithms with iterations cannot be easily transformed into parallel algorithms, and consequently, these algorithms may not be appropriate for the MapReduce paradigm in order to reduce the computation latency [4].

All the above-mentioned studies focus on computing the individual latencies at the map and reduce stages, and do not consider the coupling effects between these two stages. In this paper, we estimate the latency of big data processing system under the MapReduce framework by establishing a two-stage queueing model, and also present the impact of adding mappers or reducers on the latency. *To the best of our knowledge, this is the first study which studies the coupling effects in the MapReduce framework. The primary contributions of this paper include*: (i) modeling the coupling effects of a MapReduce framework and (ii) estimating the latency of data processing with different number of mappers and reducers.

2 Models of Data Flow

In this section, we firstly demonstrate the data flow through the map and reduce stages, and then, we estimate the latency of data processing within these two stages.

2.1 Queueing Models of Data Processing

As shown in Fig. 1, we use queueing models to analyze the latency in each stage. The model of characterizing the data process is composed of two queues: (1) The data flow from arriving at the mappers (i.e., the memory area of map servers) to leaving the mappers can be modeled as a $M/G/K_1/K_1 + Q$ queue, given K_1 mappers in the server cluster and Q positions in the buffering area of the mappers, e.g., the permanent storage areas of the mappers. In the following, to simplify the model, we assume $Q = \infty$ and use a $M/G/K_1/\infty$ queue to model the data flow arriving at and departing from the mappers since the size of buffering area at the map servers is rarely the limiting factor. (2) The data flow arriving at the reducers (i.e., the memory area of reduce servers) can be characterized as a $G/GI/K_2/K_2$ queue, in which the arrival process is general, the service time is independent and identically distributed in a general distribution, and the number of reducers is K_2 with no buffer capacity.

Model of Data Processing at the Map End Let λ denote the arrival data rate and μ_0 denote the average rate of service at the map end without considering the coupling effects with the reduce end. Due to the possibility of blocking the

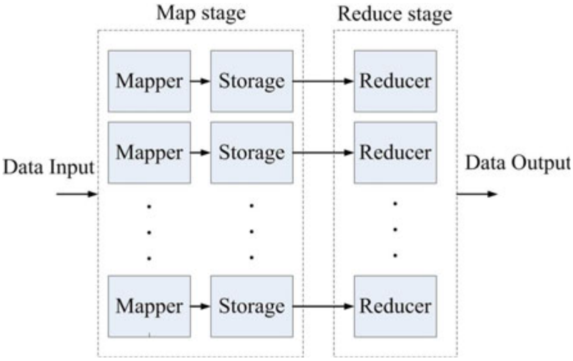


Fig. 1. The figure illustrates the model of MapReduce process.

data transfer from the map end to the reduce end, we can adjust the average service rate at the reduce end in the ED, μ, (represented by a $M/G/K_1/\infty$ queue) as

$$1/\mu = 1/\mu_0 + P_B \times \text{mean}(\min_j W_j) \tag{1}$$

where W_j $(j = 1, 2, \ldots, K_2)$ denotes the latency of data waiting at the map end until jth reducer is available, P_B denotes the probability of being blocked to transfer to reducers.

Equation (1) indicates that the average service time $1/\mu$ at the map end equals to the service time on the mappers and the waiting time to be transferred to the reducers.

Model of Data Processing at the Reduce End The data flow arriving and leaving the reducers can be modeled as a $G/GI/K_2/K_2$ queue. Mathematically, we can represent the blocking probability of P_B as [5]

$$P_B = \frac{\gamma\zeta e^{-\kappa\zeta/v}}{(1 - e^{-\kappa\zeta/v})\rho_R\sqrt{K_2}} \tag{2}$$

where K_2 denotes the number of available reducers, μ_R denotes the average service rate at the reducers, and the other parameters in Eq. (2) can be represented as:

$$\rho_R = \frac{R_T}{K_2\mu_R}, \qquad \zeta = \sqrt{K_2}(1 - \rho_R),$$

$$\kappa = \sqrt{K_2}, \qquad v = \frac{1 + C_a^2}{2},$$

$$\gamma = \left[1 + \zeta\Phi(\zeta)/\varphi(\zeta)\right]^{-1},$$

where C_a^2 denotes the squared coefficient of variation (SCV) of the service time at the reduce end, $\Phi(\cdot)$ and $\varphi(\cdot)$ denote the cumulative distribution function (CDF) and probability density function (PDF) of a standard normal distribution, respectively.

2.2 Steady-State Conditions of Queuing Models

As shown in Fig. 1, the data flow through the map and reduce ends can be represented as two coupled queues. In the following, we investigate the steady-state conditions of both queueing models. Given the data rate λ, we can easily show that the necessary and sufficient steady-state condition of the first queue (i.e., the queue at the map stage) is $\lambda \leq K_1\mu$, and also we can show that the condition of the second queue (i.e., the queue at the reduce stage) is $R_T \leq K_2\mu_R$. However, we cannot determine whether these two queues are capable of achieving the steady state because of the unknown μ, which depends on both the service rate at the mappers μ_0 and the service rate at the reducers. Also, μ is interdependent with P_B (shown in Eq. (1)), and neither of them can be mathematically expressed in a closed form. In the following, we investigate the computation of μ to achieve a necessary and sufficient steady-state condition.

Lemma 1. *The following equation has at least one feasible solution*

$$y_1(\mu) + y_2(\mu) - 1/\mu = y_0, \tag{3}$$

any solution $\hat{\mu}$ *satisfies* $\hat{\mu} \in [\mu_{\min}, \mu_{\max}]$*, given* $\mu_{\min} = 1/(1/\mu_0 + \mathit{mean}(\min_j W_j)), \mu_{\max} = \mu_0, y_1(\mu) = 1/\mu, y_2(\mu) = 1/\mu - P_b \times \mathit{mean}(\min_i W_i)$*, and* $y_0 = 1/\mu_0$.

Proof. We proof it by contradiction. Set $y(\mu) = y_1(\mu) + y_2(\mu) - 1/\mu$. By Eq. (1), we can get $y_2(\hat{\mu}) \leq y(\hat{\mu}) \leq y_1(\hat{\mu})$. If Eq. (3) has no feasible solution, then $y_0 = y_2(\hat{\mu}) < y(\hat{\mu})$ at $\hat{\mu} = \mu_{\min}$, otherwise $\mu_{\min}$ is the feasible solution. Since $y(\mu)$ is continuous and we assume no feasible solution in the range of $[\mu_{\min}, \mu_{\max}]$, we can deduce that $y(\hat{\mu}) > y_0$ for $\hat{\mu} \in [\mu_{\min}, \mu_{\max}]$. At $\hat{\mu} = \mu_{\max}$, we can achieve that $y(\hat{\mu}) > y_0 = y_1(\hat{\mu})$, which is contradicted with $y(\hat{\mu}) \leq y_1(\hat{\mu})$ by Eq. (1).

Beyond the proof, we present that at least one solution exists, which is intuitively demonstrated in circle in Fig. 2, since $y_2(\mu) \leq y(\mu) \leq y_1(\mu)$, $y_2(\mu_{\min}) = y_0$, and $y_1(\mu_{\max}) = y_0$.

If a unique solution $\hat{\mu}$ to Eq. (3) exists, we can easily show the necessary and sufficient steady-state condition as $\lambda \leq K_1\hat{\mu}$ for the first queue and $\min\{\lambda, K_1\hat{\mu}\} \leq K_2\mu_R$ for the second queue (shown in Theorem 2). If multiple solutions to Eq. (3) exist, we cannot achieve the necessary and sufficient steady-state condition. Instead, we can find the tightest sufficient (not necessary) condition. Denote that $\hat{\mu}_{\min}$ and $\hat{\mu}_{\max}$ as the minimal and maximal feasible solutions, respectively. We can easily show that the tightest sufficient (not necessary) steady-state condition: $\lambda \leq K_1\hat{\mu}_{\min}$ for the first queue and $\min\{\lambda, K_1\hat{\mu}_{\min}\}+ \leq K_2\mu_R$ for the second queue. All the above-mentioned discussions can be summarized in Theorems 1 and 2, respectively.

Theorem 1. *The necessary and sufficient steady-state condition of the coupled queues for map and reduce stages exists if Eq. (3) has a unique solution* $\hat{\mu}$*. The necessary and sufficient steady-state condition can be shown as* $\lambda \leq K_1\hat{\mu}$ *for the first queue and* $\min\{\lambda, K_1\hat{\mu}\} \leq K_2\mu_R$ *for the second queue.*

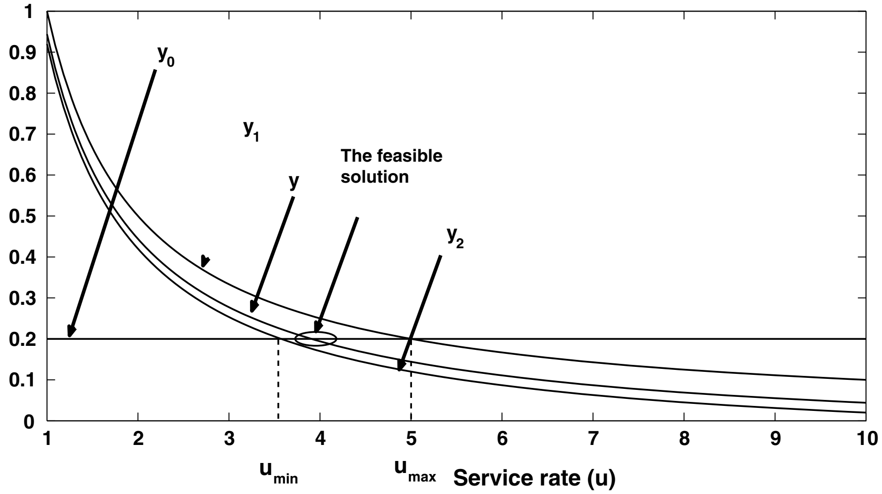


Fig. 2. Intuitively demonstrating the proof.

Theorem 2. *If Eq. (3) has multiple solutions, the tightest sufficient (not necessary) steady-state condition of the coupled queues for map and reduce stages is:* $\lambda \leq K_1\hat{\mu}_{\min}$ *for the first queue and* $\min\{\lambda, K_1\hat{\mu}_{\min}\}+ \leq K_2\mu_R$ *for the second queue.*

3 Estimation of the Latency of Queueing Models

In this section, we address the estimation of the latency at the map and reduce stages. The latency $T^C_{tot} = T^M_W + T^M_S + T^R_W + T^R_S$ is composed of four parts: (1) the waiting time to be served on the mappers T^M_W; (2) the service time on the mappers T^M_S; (3) the waiting time to be transferred to reducers T^R_W; and (4) the service time on the reducers T^R_S. The average service time on the mappers can be denoted as $T^M_S = 1/\mu_0$, and the average service time on the reducers can be denoted as $T^R_S = 1/\mu_R$. In the following, we focus on the latency of T^M_W and T^R_W.

3.1 Estimation of the Waiting Time to Be Served on the Mappers

In the following, we investigate how to compute the average waiting time to be served on the mappers, and the process of data flow at the map end can be modeled as a $M/G/K_1/\infty$ queue. The average waiting time of this queue can be denoted as [5]

$$T^M_W = \left[1/\mu + \frac{P_Q/\mu}{K_1 - \lambda/\mu}\right] \frac{1 + C^2_a}{2} \tag{4}$$

where C^2_a denotes the SCV of the service time at the map end, λ denotes the arrival data rate, μ denotes the mean service rate for data processing at the map end, P_Q denotes the probability as

$$P_Q = \frac{(K_1\rho)^{K_1}}{K_1!(1-\rho)}\left[\sum\nolimits_{p=1}^{K_1-1}\frac{(K_1\rho)^p}{p!} + \sum\nolimits_{p=K_1}^{\infty}\frac{(K_1\rho)^p}{K_1!K_1^{p-K_1}}\right]^{-1} \tag{5}$$

where $\rho = \frac{\lambda}{\mu K_1}$.

Equation (5) indicates that T_W^M depends on μ, and Eq. (1) shows that the computation of μ depends on mean($\min_i W_i$), which will be detailed in the following Sect. 3.2.

3.2 Estimation of the Waiting Time to Be Transferred to Reducers

In the following, we study how to compute the average waiting time to be transferred to reducers, i.e., mean($\min_i W_i$). Denote $F_i(z)$ as the cumulative distribution function (CDF) of W_i ($i = 1, \ldots, K_2$). Then the CDF of $\min_i W_i$ (denoted as $F_{\min}(z)$) can be expressed as

$$F_{\min}(z) = 1 - \prod_{i=1}^{K_2}[1 - F_i(z)] \tag{6}$$

Based on Eq. (6), we can achieve the mean of $\min_i W_i$ as

$$\text{mean}(\min_i W_i) = \int z dF_{\min}(z) \tag{7}$$

By substituting Eq. (6) into Eq. (7), we can compute the average waiting time to be transferred to reducers, i.e., mean($\min_i W_i$).

4 Simulation Results

In this section, we employ multiple classification algorithms under the MapReduce framework on the real data of a few public datasets [6], including the Amazon reviews dataset, MovieLens reviews dataset, Yahoo music user ratings dataset, Youtube comedy slam preference dataset, and Thomson Reuters text research collection dataset. The basic information of the above-mentioned datasets is summarized in Table 1.

Table 1. Information of datasets for classification algorithms

Dataset	Samples
Amazon reviews	82,000,000
MovieLens reviews	22,000,000
Yahoo music user rating	10,000,000
Youtube comedy slam preference	1,138,562
Thomson Reuters text research	1,800,370

Table 2. Latency of various classification algorithms

Dataset	Latency [s]											
	wlr			nb			log			svm		
	S	C	N	S	C	N	S	C	N	S	C	N
Am	20.7	20.3	19.2	19.8	19.6	18.3	20.2	19.8	18.6	18.9	18.6	17.4
Mo	6.1	6.0	4.9	5.6	5.5	4.5	5.9	5.8	4.8	5.0	5.0	4.0
Ya	3.1	3.0	2.4	2.8	2.7	2.2	2.9	2.9	2.3	2.5	2.4	1.7
Yo	0.6	0.6	0.4	0.5	0.5	0.3	0.6	0.6	0.3	0.5	0.5	0.3
Th	1.4	1.4	1.0	1.4	1.4	1.1	1.4	1.4	1.1	1.2	1.2	0.8

4.1 Latency of Various Classification Algorithms

The latency of various classification algorithms on various datasets is shown in Table 2, in which 'Am' refers to the Amazon reviews dataset, 'Mo' refers to MovieLens reviews dataset, 'Ya' refers to Yahoo music user ratings dataset, 'Yo' refers to Youtube comedy slam preference dataset, 'Th' refers to Thomson Reuters text research collection dataset. Also, 'wlr' refers to the weighted linear regression algorithm, 'nb' refers to the Naive Bayes algorithm, 'log' refers to the logistic regression algorithm, and 'svm' refers to the support vector machine algorithm. 'S' refers to the simulation results where we establish a cluster of mappers and a cluster of reducers and compute the running time in the real lab environment; 'C' refers to the numerical results in view of the coupling effects (T_{tot}^{C}); 'N' refers to the numerical results without considering the coupling effects, i.e., the total latency of individual map and reduce stages ($T_{tot}^{N} = T_{W}^{M} + T_{S}^{M} + T_{S}^{R}$) [7]. As shown in Table 2, the estimation by our algorithms in view of coupling effects is closer to the latency in the simulation than the estimation without considering coupling effects in [7].

4.2 Latency with Different Number of Mappers and Reducers

Table 2 shows the latency with only one mapper and one reducer. In the following, we investigate the latency with various numbers of mappers and reducers. As shown in Fig. 3, the latency by the weighted linear regression algorithm and the support vector machine algorithm almost linearly decrease with the number of mappers and the number of reducers. Also, the latency can decrease to 1/10 when adding 10× mappers, while the latency is only around 1/3 when adding 10× reducers. Through adding more mappers, we can dramatically reduce the latency. This result is in line with [7], which demonstrates that the processing algorithms at the mappers have a higher computation complexity than those at the reduces.

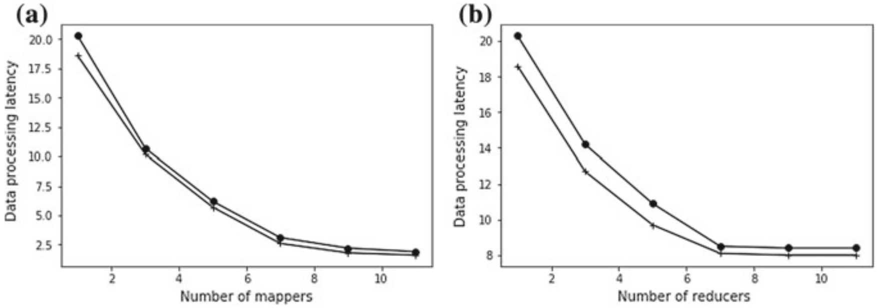


Fig. 3. The Figure illustrates the latency with different number of mappers. Line with 'o' represents the weighted linear regression algorithm, and line with '+' represents the support vector machine algorithm.

5 Conclusion

In this paper, we establish a two-stage queueing model to mimic the MapReduce process in consideration of the coupling effects between the map and reduce stages. Also, with this model, we estimate the latency of data processing under the MapReduce framework with multiple classification algorithms on various datasets. The results show that our proposed algorithm in consideration of the coupling effects outperforms the algorithms without considering the coupling effects in the estimation of data processing latency. Also, the results show that the bottleneck of data processing is usually at the mapping stage, and adding mappers is more efficient than adding the reducers to reduce the latency of data processing.

References

1. Grolinger K, Hayes M, Higashino WA, Heureux A, Allison DS, Capretz MAM (2014) Challenges for MapReduce in big data. In: IEEE World Congress Services (SERVICES), pp 182–189
2. Najafabadi MM, Villanustre F, Khoshgoftaar TM, Seliya N, Wald R, Muharemagic E (2015) Deep learning applications and challenges in big data analytics. J. Big Data 2(1):1
3. Kune R, Konugurthi PK, Agarwal A, Chillarige RR, Buyya R (2016) The anatomy of big data computing. Softw Pract Experience 46(1):79–105
4. Jagadish HV et al (2014) Big data and its technical challenges. Commun ACM 57(7):86–94
5. Lin D, Patrick J, Labeau F (2014) Estimating the waiting time of multi-priority emergency patients with downstream blocking. Health Care Manage Sci 17(1):88–99
6. Wissner-Gross A (2016) Datasets over algorithms. https://www.Edge.com. Retrieved 8 Jan 2016
7. Schölkopf B, Platt J, Hofmann T (2007) Map-reduce for machine learning on multicore. In: Advances in neural information processing systems. Proceedings of the 2006 conference. MIT Press, pp 281–288

Zone-Based Resource Allocation Strategy for Heterogeneous Spark Clusters

Yao Qin, Yu Tang(✉), Xun Zhu, Chuanxiang Yan, Chenyao Wu, and Di Lin

University of Electronic Science and Technology of China, No. 4, Section 2,
North Jianshe Road, Chengdu, People's Republic of China
yutang@uestc.edu.cn

Abstract. As a primary big data processing framework, Spark can support memory computing to improve the computation efficiency. However, Spark cannot handle the situation of a heterogeneous cluster, in which the nodes have different structures. Specifically, a primary problem in Spark is that the resource allocation strategy based on the number of homogeneous processor cores cannot adapt to the heterogeneous cluster environment. To solve the above-mentioned problem, we propose a zone-based resource allocation strategy based on heterogeneous Spark cluster (ZbRAS) and implement such a strategy to improve the efficiency of Spark. We compare the proposed strategy with the native resource allocation strategy of Spark, and the comparison results show that our proposed strategy can significantly enhance the execution speed of Spark jobs in a heterogeneous cluster.

Keywords: Spark · Heterogeneity · Scheduling

1 Introduction

In recent years, Spark [1] and Hadoop [2] have emerged as an efficient framework that is being extensively deployed to support a variety of big data applications. And modern computer clusters supporting big data platforms such as Spark are more and more heterogeneous with the continuous iterative updating of the hardware in the cluster. The main challenge faced by researchers and IT practitioners in optimizing Spark framework is how to make the best use of the computation capacity in a heterogeneous cluster environment to deal with the exponentially growing data volumes by an economically viable fashion.

Heterogeneous clusters mean that computers in a cluster have different hardware configurations, resulting in different computing performance of compute nodes in the cluster. Therefore, scheduling optimization for heterogeneous clusters is an important direction in distributed computing. However, in heterogeneous clusters, the computing resource cannot be used reasonably, and the job with higher priority cannot obtain higher performance computing resource.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 113–121, 2020.
https://doi.org/10.1007/978-981-15-0187-6_13

In order to make full use of the heterogeneity of the cluster and solve the fluctuations and differences in job performance caused by different hardware configurations in heterogeneous clusters, cluster heterogeneity should be fully considered when assigning Executors to the application in the process of Spark resource allocation [3].

2 System Model

Zone-based resource allocation strategy (ZbRAS) enhances Spark for heterogeneous clusters by scheduling tasks based on zone division module and resource allocation module and can improve the execution speed of the job by assigning high-performance computing resources to the job.

2.1 The Definition of Zone

We have analyzed the existence of optimization space for Spark scheduling on heterogeneous clusters. The following cluster consists of three different types of computers. The three types of computers have the same number of CPU cores and the same memory size. The CPU performance of the Type-1 nodes is the highest, and the CPU performance of the Type-2 and Type-3 nodes is second. Figure 1 shows the general task schedule in Spark. The task to be executed is randomly assigned to the node in the cluster. Obviously, the default scheduling strategy does not fully utilize the highest performing Type-1 nodes in the cluster.

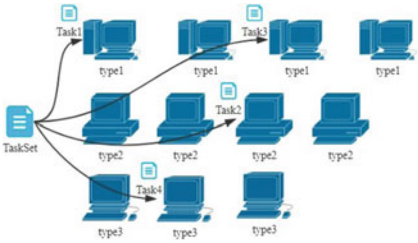


Fig. 1. Scheduling in native Spark.

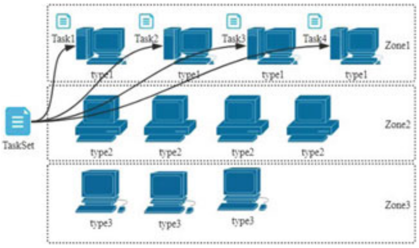


Fig. 2. Scheduling in Spark with ZbRAS.

To reflect the similarities and differences between computers in a Spark cluster, we introduce a new definition—Zone. Different zones represent computer clusters of different performance and also represent computing resources of different priorities. With sufficient computing resources, users can schedule high-priority jobs to run in high-performance zones and low-priority jobs to run in low-performance zones. This strategy makes it possible to use computers in the cluster more reasonably.

Spark is a big data computing framework which divides cluster resources by CPU cores and can complete jobs' running in a limited time. Therefore, in

Spark, the heterogeneity of different computers is reflected in the time difference between worker [4] in different computer performing the same Spark job. In addition, since distributed computing requires data locality, the computers of high data locality have the higher concurrency, the faster the job runs, and the amount of computer concurrency is determined by the number of CPU cores. In conclusion, the heterogeneity of computers in Spark is reflected in the difference in the execution power of the CPU core and the number of CPU cores. Define the heterogeneity of the computer in the Spark cluster by CPU cores (cpu_cores) and computing power ($worker\_capacity$), see Eq. 1.

$$Heterogeneity = (cpu\_cores, worker\_capacity) \tag{1}$$

$$Zone = (Heterogeneity, \sum_{i=1}^{m} Worker\_i) \tag{2}$$

$$Spark\_cluster = (Heterogeneity, \sum_{j=1}^{n} Zone\_j) \tag{3}$$

Then, define the zone as a grouping of computers with the same $Heterogeneity$ in the cluster, see Eq. 2. After the zone division, the Spark cluster is composed of zones; see Eq. 3.

After the zone is introduced, in Spark, the task can be assigned to the specified zone to obtain the computing performance brought by the heterogeneity of the zone according to the zone information. Figure 2 shows the task schedule in Spark after the zone is introduced.

By dividing the cluster into zones, users can prioritize jobs based on their real-time requirements. Jobs with higher priorities run in high-performance zones, and jobs with lower priorities run in low-performance zones. It means that cluster resources have space for scheduling optimization, and scheduling optimization for heterogeneity can improve the performance of Spark.

2.2 Zone Division Strategy

To enable Spark to allocate and schedule resources based on the hardware performance of each node in a heterogeneous cluster, the first problem to be solved is how to make Spark aware of the heterogeneity of the cluster and obtain the parameter $worker\_capacity$ in Eq. 1. Computers are made up of complex hardwares. Different types of computers have a variety of hardware configurations, so it is difficult to calculate the performance of various computers only through hardware configurations. For the performance of the computer in the Spark framework, we can test the results of the Spark task execution by executing the Spark program and score the performance of the computer to obtain the $worker\_capacity$ of each worker in the Spark job.

The conventional method is to create a benchmark program. The benchmark is a Spark job that runs only on one computer worker. We choose the classic WordCount program, which reads text files and performs word frequency statistics on text data. The job result $worker\_result$ is the execution time of the program, and time and speed are inversely proportional. Therefore, the reciprocal of *worker_result* reflects the computing power of the worker, and the heterogeneity difference between different workers is reflected in different *worker_capacity*. *worker_result* is calculated by running the benchmark program *benchmark_job* in the worker. After the introduction of benchmark, the definition of zone can also be expressed as a grouping of computers in the cluster with the same benchmark performance and the same number of CPU cores.

2.3 Zone-Based Resource Allocation Strategy

Algorithm 1 Zone-based Resource Allocation Strategy

```
Input: spark.zone.level,spark.cores
Output: allocated computing resources
function AllocateResource(spark.zone.level, spark.cores)
    reqCores ← spark.cores
    zoneLevel ← spark.zone.level
    targetZone ← getById(zoneLevel)
    for worker in targetZone.workerList do
        if $reqCores >= worker.coreNum - 1$ then
            allocateResources(app,worker)
            $reqCores ← reqCores - worker.coreNum$
        end if
        if $reqCores <= 0$ then
            return
        end if
    end for
    zoneList.filter(zoneLevel)
    zoneQueue ← zoneList.sortByCpuCapacity().sortByCpuNums()
    while $reqCores > 0$ and !zoneQueue.isEmpty() do
        zone ← zoneQueue.pop()
        for worker in zone do
            if $reqCores >= worker.coreNum - 1$ then
                allocateResources(app,worker)
                $reqCores ← reqCores - worker.coreNum$
            end if
        end for
    end while
    return
end function
```

The above section defines zones and the heterogeneity between zones in a heterogeneous cluster. This part will design an allocation strategy of computing resource that utilizes cluster heterogeneity to avoid heterogeneity affecting job execution, improve job execution speed and perform zones scheduling according to user requirements and make rational use of cluster computing resources.

The following describes the zone-based resource allocation strategy in the form of pseudocode, see Algorithm 1. The resource allocation strategy relies on two input parameters, the configuration item cores.max represents the number of cores that the user wants to start for the job, and the configuration item zone.level represents the target area that the user wants to schedule for the job, which reflects the priority score of the job.

Spark processes on different nodes communicate through remote procedure calls [5] (RPC). In order to keep the functionality and structure of Spark intact, and to ensure the elegance and clarity of the code, we chose to set up a new component, the Schedule Center. The Schedule Center is responsible for scheduling optimization functions and providing services to Spark.

3 Simulation

In this section, we present the evaluation of the zone-based resource allocation strategy using both real deployment on a heterogeneous cluster. The cluster used in the experiment consisted of four different models of computers with different hardware configurations. The computer configuration is shown in Table 1.

Table 1. Experimental hardware configuration environment

Computer model	Processor model	Memory size	Cores number
Lenovo T530	i5-3210M	4G*1+8G*1	4 cores
Asus X550V	i5-3230M	4G*1	4 cores
Lenovo ER202	CeleronE3200	4G*1	2 cores
TF Z01	i7-3632QM	4G*2	8 cores

3.1 Experiment I

The purpose of Experiment I is to divide the clusters built in the laboratory according to the zone division rules formulated in the previous section and prepare for resource allocation and job scheduling.

The experiment uses the benchmark application *WordCount*, which is from the well-known Hadoop HiBench Benchmark Suite [6], to test the running speed of Spark jobs on three types of nodes in the cluster, divides of heterogeneous clusters into different zones based on the experimental results and then configures the data related to the zone division into the zone division module and the resource allocation module of the Schedule Center.

The benchmark application is configured as follows. The Spark WordCount application is run on each computer in stand-alone mode. Each computer starts an Executor, and the amount of data processed is 1GB. The configuration of Experiment I is shown in Table 2. And Table 3 shows the results of running the benchmark application on each of the eight worker computers in the cluster.

Table 2. Hardware configuration of Experiment I

group_no	computer_model	Character	Number	cores_number
1	TF Z01	Worker	1	8 cores
2	Lenovo T530	Worker	3	4 cores
3	Lenovo ER202	Worker	4	2 cores
4	Asus X550V	Driver	1	4 cores

Table 3. Result of Experiment I.

No.	computer_1 (s)	computer_2 (s)	computer_3 (s)	computer_4 (s)	average_time (s)
group_1	12.846				12.846
group_2	26.518	27.156	25.984		26.553
group_3	54.194	54.642	56.158	55.110	55.026

It can be seen that the difference in performance between the three types of computers in running benchmark is obvious, so the cluster can be divided into three zones, namely zone 1 to zone 3. After calculation, the configuration of the three zones is shown in Table 4.

Now, we can configure the Schedule Center based on the experimental results. And it is possible to conduct zone scheduling experiments and test the effect of zone scheduling on the optimization of native Spark.

Table 4. Result of Experiment I: Zone division

zone_no	zone_1	zone_2	zone_3
computer_model	TF Z01	Lenovo T530	Lenovo ER202
*wordcount*_time (s)	12.846	26.553	55.026
*benchmark*_score	0.03241911	0.01768284	0.00968279
zone_score	4.28	2.07	1.0

3.2 Experiment II

The purpose of Experiment II is to verify that zone-based resource allocation enables Spark to take full advantage of the cluster's high-performance nodes, thereby increasing the speed of the job. At the same time, users can set different priorities for jobs based on the real-time requirements of the job and assign jobs to the specified zone to run. The cluster used in Experiment II consisted of nine computers. See Table 5 for configuration information of the experiment.

Table 5. Hardware configuration of Experiment II

zone_no	character	computers_no
1	Worker	1
2	Worker	2, 3, 4
3	Worker driver	5, 6, 7, 8

Table 6. Configuration of Experiment II

test_no	Native Spark required cores	Spark with ZbRAS job propriety
1	12 cores	2
2	8 cores	2
3	8 cores	3
4	16 cores	1

Table 7. Result of Test 1

	Native Spark	ZbRAS	Optimization rate
Time	33.41 s	28.01 s	16.2%
Node	4, 5, 6, 7, 8	2, 3, 4	

Table 8. Result of Test 2

	Native Spark	ZbRAS	Optimization rate
Time	46.69 s	37.11 s	20.5%
Node	4, 5, 6, 7	2, 3	

After Experiment I to carry out zone division, Experiment II can be carried out on zone scheduling to test the degree of optimization of zone scheduling to Spark. Experiment II will use zone scheduling and native Spark scheduling to test and compare the scheduling results and running time of the two. The purpose of zone scheduling is to select the optimal computing resources for high-priority jobs when the computing resources in the cluster are sufficient.

Experiment II needs to compare the native Spark scheduling with zone scheduling. In the native Spark, we need to configure the number of CPU cores for the job. In the improved version of Spark with ZbRAS, we need to configure the zone of the job scheduling, that is, the job priority and the number of CPU cores. The test plan for Experiment II is shown in Table 6.

The experimental results of Test 1 are shown in Table 7. In Test 1, the native Spark started five nodes in order to start 12 core computing resources, and the zone scheduling started three nodes which are of Type-2. The two schedules allocated the same number of CPUs, but there are two types of computers in the native Spark scheduling. Therefore, the zone scheduling result is better than the native Spark scheduling. Table 8 shows the experimental results of Test 2. In Test 2, since the number of cores required for the job changed from 12 to 8, the native Spark started four nodes which are of Type-2, and the zone schedule started two nodes which are of Type-2. The results of Test 2 fully reflected the performance difference between two models of the computer. The experimental results of Test 3 are shown in Table 9. In Test 3, the zone scheduling dispatches the job to zone 3, which is the same scheduling result as the native Spark schedule. The experimental results of Test 4 are shown in Table 10. In Test 4, in order to use 16 CPU cores, the native Spark started six nodes and was not randomly assigned to the best performing node 1. But the zone scheduling started node 1 in zone 1 according to the user's configuration with ZbRAS. The

Table 9. Result of Test 3

	Native Spark	ZbRAS	Optimization rate
Time	46.69 s	47.95 s	
Node	4, 5, 6, 7	4, 5, 6, 7	

Table 10. Result of Test 4

	Native Spark	ZbRAS	Optimization rate
Time	22.91 s	19.45 s	15.1%
Node	2, 4, 5, 6, 7, 8	1, 2, 3	

strategy continues to start in zone 2, starting node 2 and node 3 for a total of three Executors. The result of the final zone scheduling is much better than the native Spark scheduling.

In conclusion, Test 1 and Test 2 reflect that ZbRAS has made full use of the heterogeneity of the cluster and improved the execution speed of the job by assigning high-performance computing resources to the job. Test 3 reflects that ZbRAS can schedule the job to the specified area according to the user set job priority. Test 4 reflects that ZbRAS will obtain computing resources in a lower-level zone when the resources in the required zone are insufficient. Overall, it shows that after zone scheduling, resource utilization is more reasonable, and the execution speed of the application is increased by an average of 17%.

4 Conclusion

Overall, the experimental results show that, in heterogeneous Spark cluster, the partition scheduling algorithm *ZbRAS* enables Spark to allocate resources based on the heterogeneity of the cluster, gives the user the ability to allocate more detailed resources for the job and makes the task scheduling more reasonable and the task assignment more uniform, and the execution speed of Spark jobs is significantly improved.

Acknowledgements. This work is jointly supported by the National Natural Science Foundation of China (No. 61601082, No. 61471100, No. 61701503, No. 61750110527).

References

1. Armbrust M, Das T, Davidson A, Ghodsi A, Or A, Rosen J, Stoica I, Wendell P, Xin R, Zaharia M (2015) Scaling spark in the real world: performance and usability. Proc VLDB Endowment 8(12):1840–1843
2. Apache Hadoop (2011) http://hadoop.apache.org
3. Gao H, Yang Z, Bhimani J, Wang T, Wang J, Sheng B, Mi N (2017) Autopath: harnessing parallel execution paths for efficient resource allocation in multi-stage big data frameworks. In: 2017 26th international conference on computer communication and networks (ICCCN). IEEE, pp 1–9
4. Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I (2010) Spark: cluster computing with working sets. HotCloud 10(10–10):95

5. Birrell AD, Nelson BJ (1984) Implementing remote procedure calls. ACM Trans Comput Syst (TOCS) 2(1):39–59
6. Huang S, Huang J, Liu Y, Yi L, Dai J (2010) Hibench: a representative and comprehensive hadoop benchmark suite. In: Proceedings of the ICDE workshops, pp 41–51

Battle Prediction System in StarCraft Combined with Topographic Considerations

ChengZhen Meng, Yu Tang(✉), ChenYao Wu, and Di Lin

University of Electronic Science and Technology of China, Shahe Campus:
No. 4, Section 2, North Jianshe Road, Chengdu 610054, Sichuan,
People's Republic of China
yutang@uestc.edu.cn

Abstract. This paper focuses on the prediction of combat outcomes in a local battle during a game of StarCraft. Through the analysis of the initial state of the two armies and considering the influence of the terrain in StarCraft on the combat effectiveness of both sides, the concept of the terrain correction factor is introduced to establish a mathematical model. Secondly, using SparCraft is to simulate battles and generate data sets. Finally, the maximum a posteriori probability estimation (MAP) is used to train the previously established data set to complete the parameter estimation of the mathematical model.

Keywords: SparCraft · MAP · Bayesian formula

1 Overview

With the continuous development of machine learning technology, artificial intelligence has received more and more attention from society, and people have more and more demand for high-performance artificial intelligence. As a result, many artificial intelligence-related competitions have emerged. Real-time strategy games (RTS), because of their unique tactical nature and a well-defined complex confrontation system they represented, which can be divided into many interesting subproblems, have become a hot area for testing artificial intelligence.

With regard to the field of artificial intelligence in StarCraft, there are many directions to be studied, from pathfinding to image recognition, strategic decision making, etc., and the core point is to judge the outcome of small-scale battles in a game, accurately predicting the outcome of the battle will be regard as a very important reference for AI's subsequent tactical decisions and actions. In StarCraft, the factors involved in a battle are mainly the size and composition of the armed forces on both sides of the war. The main purpose is to improve the model-related parameters through the mathematical composition of the two sides, through mathematical modeling, and combined with Bayesian classifier algorithm, and finally achieve the purpose of more accurate prediction of the battle results. Namely given the size and composition of Army A and Army B, it is possible to answer more accurately: Who will win? [1].

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 122–130, 2020.
https://doi.org/10.1007/978-981-15-0187-6_14

2 Related Technology Introduction

2.1 Maximum Posterior Probability Estimate [2, 3]

In Bayesian statistics, the "maximum posterior probability estimate" is the mode of the posterior probability distribution. Point estimates of quantities that are not directly observable in the experimental data can be obtained using the maximum a posteriori probability estimate. It is closely related to the classical method of maximum likelihood estimation, but it uses an augmented optimization goal to further consider the prior probability distribution of the estimated quantity. Therefore, the maximum posterior probability estimate can be regarded as the maximum likelihood estimate of regularization.

Although the use of posteriori distribution was shared by the maximum a posteriori estimate and the Bayesian statistic, MAP is generally not considered a Bayesian method, the maximum posterior estimate is a point estimate; whereas, the Bayesian method is characterized by using these distributions to summarize the data and get inferences. The Bayesian method attempts to calculate the posterior mean or median and the posterior interval, not the posterior model. This is especially true when the posterior distribution does not have a simple analytical form: In this case, the posterior distribution can be modeled using Markov chain Monte Carlo techniques, but it is difficult to find the optimization of its modulus.

2.2 StarCraft Combat Simulation System SparCraft

SparCraft is an open-source abstract StarCraft combat simulation package for Windows and Linux. It can be used to create stand-alone combat simulations or import into existing BWAPI-based StarCraft robots to provide additional AI functionality.

SparCraft is built very fast and uses a frame fast-forward system to bypass game frames, during which no combat decisions are made (i.e., during attack cooling, moving, etc.). With this system, SparCraft can easily perform tens of millions of unit actions per second.

For training and testing purposes, we generated data sets using the StarCraft combat simulator (developed by SparCraft, UAlberta). The simulator allows for the creation and execution of battles based on deterministic scripts (or decision agents such as confrontational search algorithms). We chose to use simulation data instead of data from real games, mainly because the simulator allowed us to generate large amounts of data and avoid unnecessary interference with data results due to different player game styles and operating techniques [4, 5].

3 Mathematical Modeling

In StarCraft, the victory of a battle depends largely on a series of parameters such as damage, range of attack, hit rate, armor value, and speed of each unit in both armies. Therefore, in the model, we need to build a set of characteristic attributes for each unit, and the system calculates the result based on the attribute values of each unit in an army. Another advantage of such a model is that it can be judged that the attribute has the greatest impact on the outcome of the victory, that is, the main factors that can clarify the victory of a battle in a vaguer way [4].

In addition, there is an important terrain in StarCraft. The difference between high and low terrain, the one standing in the high ground often has greater advantages, including better visibility, wider attack range, better hash formation, etc. Therefore, it is one-sided to judge the outcome of a battle by considering only the attributes of the unit itself. Therefore, based on the above ideas, we should further model. That is, based on the above model, a terrain parameter of the highland and lowland is added as a correction factor.

For ease of calculation, the above mathematical modeling analysis should not be performed simultaneously. The situation should be analyzed first when both sides are on the ground, and after the results are obtained, the situation will be modeled when one of the parties is at a high level and the other is at a low level.

3.1 Core Ideas [6]

A battle process in StarCraft can be simplified as a unit in an army that outputs firepower to enemy forces during their own survival [4, 7], thereby destroying the enemy. Therefore, a relatively straightforward calculation model for combat wins and losses is to compare the output values of all units in the military during the survival period. If the one with the larger output value wins, Formula (3.1) is calculated as follows:

$$\text{Result} = \sum_{a \in A} \text{Lp}(a)\text{Atk}(a) - \sum_{b \in B} \text{Lp}(b)\text{Atk}(b) \tag{3.1}$$

Among them, A and B, respectively, represent the total collection of units of Army A and Army B of the warring parties. Lp(a) represents the sum of the damage per life caused by a unit of a, and Atk(a) represents the sum of the attack values per second of a unit. The central idea of winning a battle is to minimize the damage it takes in a battle and increase its output in a battle.

The above formula is obviously too simple, and it ignores the complex combat system in StarCraft. But, we can use the general idea of this formula to further expand it.

3.2 Ideas to Improve [8]

StarCraft's combat system is more complex, and each unit has its own unique attributes, including but not limited to: attack power, health, armor value, attack frequency, moving speed, attack range, and so on. Therefore, we can extend Lp and Atk in the formula in 3.1—expand Lp into offensive eigenvalues and expand Atk into defensive eigenvalues. The former includes attack power, attack range, attack frequency, etc., while the latter includes health, armor, speed, etc.

At the same time, we also noticed that there is a strong confrontation between specific attack eigenvalues and defensive eigenvalues—such as attack power and health, attack range and moving speed. These eigenvalues can be grouped according to relevance, such as attack power and health value grouped together for separate modeling operations.

In summary, we can use another formula to improve the formula in 3.1, using O_A, O_B to represent the offensive eigenvalues of Army A and Army B (attack point), and D_A, D_B represent the defensive eigenvalues of Army A and Army B (defend point).

The improved model structure is shown in Fig. 1.

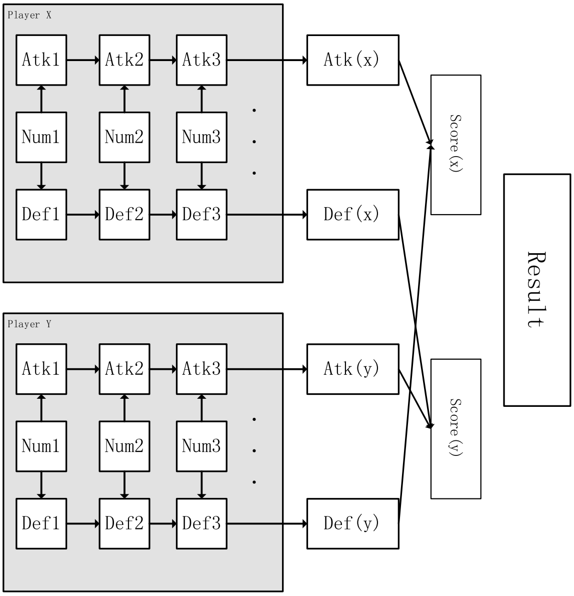


Fig. 1. Mathematical model for predicting battle outcomes

For a unit consisting of a total of *M* units, each node is constructed with one node, each node is divided into two parts, one part is the unit number count Num, the other part is composed of *K* units, and each unit consists of two eigenvalues are composed—the attack point (at the node Atk) and the defensive value (the node Def in the figure). For example, a single unit can be {harm value, health value}, or {attack Range, maneuverability}, and the two variables Atk and Def in the unit are one-dimensional Gaussian variables. The specific data structure is shown in Fig. 1.

Then, for each tuple, the number of the respective species (Num value) is combined, and an arithmetic summation operation is performed. For example, for the damage/health feature group, the Atk values of all *M* types of arms are multiplied by the Num value and then summation.

Do the same for the Def value, you can get two values: Atk and Def, for the two engagement forces *x* and *y*, we can have the total of four values—Atk_x, Def_x and Atk_y, Def_y; these four values are calculated as follows in (3.2):

$$\text{Result}_i = \left(\frac{\text{Atk}_{xi}}{\text{Def}_{yi}} - \frac{\text{Atk}_{yi}}{\text{Def}_{xi}} \right) \tag{3.2}$$

The predicted results for the damage/health feature set are thus obtained. For the other feature groups, the same operation can be performed to obtain a total of K result values.

A total of K eigenvalue groups can be normalized as shown in 3.3 to obtain the final prediction result.

$$\text{Dig} = \frac{1}{1 + \text{e}^{-\sum_{i=1}^{K} \text{Result}_i}} \tag{3.3}$$

Then, the final value Dig is a number between [0, 1], indicating the probability that player X defeats player Y to win.

So far, we have come up with a mathematical model of the outcome of the battle without considering the terrain of both sides. Let us consider how the mathematical model after adding topographical factors should be constructed.

First of all, it is necessary to be clear that the topographical factors can only occur when the two sides are in different terrains (one is in the high ground and one is in the low ground). When the two sides are in the same terrain, the above mathematical model can be used to meet the demand. Terrain factors will have an impact on the outcome of the battle if and only if one of them is on unfavorable terrain.

We assume that player Y is at a high level, and player X is at a low level. From a mathematical point of view, the terrain difference will provide a reduction to the player X's army, weakening the player's ability to fight X units. In addition, different units have different adaptability to the terrain. Some units are less affected by the terrain. Therefore, we can consider that for any types of the units i of the player X, a correction factor δ is added to each of the K units, indicating the reduction of Atk/Def in the unit. From this, we can modify Formulas (3.2)–(3.4) as follows:

$$\text{Result}_i = \left(\frac{\text{Atk}_{xi} * \delta_{xi}}{\text{Def}_{yi}} - \frac{\text{Atk}_{yi}}{\text{Def}_{xi} * \delta_{xi}} \right) \tag{3.4}$$

Thus, the terrain factor can be separated from the combat value of each unit. The follow-up training process can be divided into two steps:

1. A training set is established for the situation where both players are in the same terrain state. Training is carried out for the modeling of (3.2). By doing this, the values of Atk and Def of each unit can be calculated.
2. The configuration of the arms in step 1 is unchanged, one of them is placed in a special terrain, establishing a new training set, and training is performed for the model (3.4). At this time, the values of Atk and Def have been determined, and only the terrain correction factor δ of respective arms needs to be re-determined.

3.3 Training Model Establishment

According to the rules set out in 3.2, we should conduct two trainings in succession. First, determine the Atk and Def values for each unit regardless of the terrain factor.

Let Obs $= (O_1, O_2, O_3, \ldots, O_c)$ denote a training set with a sample size of C, which is a data set of C battles conducted through SparCraft. The i-th element O_i in Obs represents the winner of the i-th battle (i.e., X or Y).

Assume the two-dimensional matrix Vec $= \left(\text{node}_i^j\right)$, where node_i^j represents the j-th feature value of the i-th unit. Then, Vec is the characteristic matrix composed of the eigenvalues of all M arms. The initial value is the basic value of each unit in the game. For example, the damage characteristic value is the attack power of the unit in the game, and the vital sign value is the life value of the unit in the game.

Based on the above two data sets Obs and Vec, we can combine the model proposed in 3.2 and propose the Bayesian Formula (3.5).

$$P(\text{Obs}, \text{Vec}) = P(\text{Vec}) \prod_i P(O_i | \text{Vec}) \tag{3.5}$$

Based on this, we can estimate the parameters of the parameter set Vec by the maximum posterior probability estimation (MAP). It should be noted that the distribution of Ack and Def values between individual arms can be independent of each other without considering the group augmentation skills. Therefore, for $P(O_i|\text{Vec})$, we can do the split like (3.6).

$$P(O_i|\text{Vec}) = P\left(O_i|\text{node}_1^1, \text{node}_1^2, \ldots\right) = \prod_{i,j} P\left(O_i|\text{node}_i^j\right) \tag{3.6}$$

In order to facilitate the calculation, we use the method of training in batches, and for each eigenvalue, it is regarded as a random variable satisfying $N(\mu, \sigma)$. Initially, we consider μ as the initial value of node_i^j. Then, the posterior probability distribution function can be obtained by formula (3.7).

$$\theta = \frac{f(\text{Obs}|\text{Vec})f(\text{Vec})}{\int_\theta f(x|\text{Vec}')f(\text{Vec}')d\theta'} \# \tag{3.7}$$

When the maximum value is obtained, the value of Vec is the desired value.

Then, it is considered to add terrain factors to the model. Let $\vartheta = \left(\delta_i^j\right)$, where δ_i^j represents the terrain correction factor corresponding to the j-th feature value of the i-th arm. Equation 3.5 can be rewritten as (3.8).

$$P(\text{Obs}, \vartheta) = P(\text{Vec}) \prod_i P(O_i|\vartheta) \tag{3.8}$$

4 Experimental Results

4.1 Data set Acquisition

In order to develop the speedometer, we will only consider the Army of each party for the time being, regardless of the air force situation of the three races, and then consider expanding the Air Force part after the follow-up algorithm is implemented and verified [4, 9].

SparCraft requires the following information to run a simulation:

1. player's race and AI information
2. initial strength and distribution of both sides
3. map information.

In order to improve the simulation speed and reduce the interference factor, I decided to plan the above information as follows:

1. player race and AI information:

Test data should include war between three races and ethnic civil wars, a total of six types of combat.

In order to avoid the simulation error caused by the different types of AI, the AI of all the battles here adopts the same AI of the nearest attack principle.

Only consider 1v1, the rest of the situation will not be considered.

2. Initial strength and distribution of both sides:

Limit the initial strength of both sides to no more than 100.

The number of each type of arms on both sides is random. For example, if there are ten types of arms in the Terran, the number of each type of arms is completely random, but the sum of the ten types of arms should be 100. The position of each unit is randomly distributed.

3. map information:

In order to avoid interference from other factors, a flat map of $1280 * 720$ is used here, and the two armies are distributed on the left and right sides [10].

After the above training set is completed, it is also necessary to place one of them in the high ground to fight and collect the battle results under the same conditions.

4.2 Training Result

We compared this method with other standard classifiers and divided the results into two groups for comparison. One group was regarded as the control group in which both sides were placed on the same terrain, and the other group was regarded as the experimental group where one of the two sides were placed on unfavorable terrain. Each group recorded a result with a sample size of 50, 100, 200, 500, and 800 in the test set. The results are shown in Fig. 2.

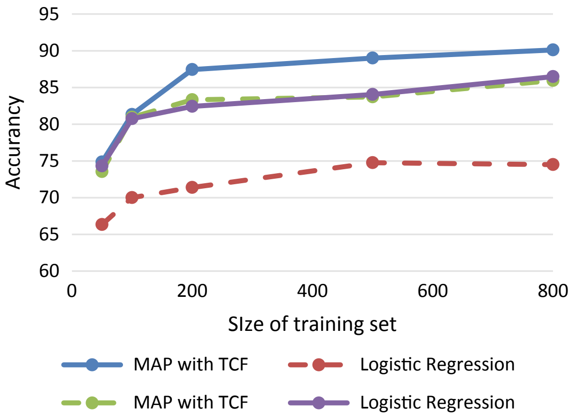


Fig. 2. Accuracy of prediction in different situations

The dotted line in the figure is the test result of the experimental group, and the solid line part is the test result of the control group. It can be seen that the addition of terrain does have a certain impact on the prediction accuracy of the model, especially when the sample size of the test set increases, it has an error of about 4% on the prediction accuracy, but compared to other classifications. In terms of the algorithm, adding a correction factor can greatly reduce the impact of terrain on the accuracy of the prediction results.

Acknowledge. This work is jointly supported by the National Natural Science Foundation of China (No. 61601082, No. 61471100, No. 61701503, No. 61750110527).

References

1. Ravari YN, Bakkes S, Spronck P (2016) Starcraft winner prediction. In: Twelfth artificial intelligence and interactive digital entertainment conference
2. Graepel T, Minka T, Herbrich RTS (2007) A Bayesian skill rating system. Adv Neural Inf Process Syst 19:569–576
3. Minka TP (2001) A family of algorithms for approximate Bayesian inference. Massachusetts Institute of Technology
4. Sánchez-Ruiz AA (2015) Predicting the outcome of small battles in starcraft. ICCBR (Workshops), pp 33–42
5. Stanescu MA, Barriga N, Buro M (2015) Using Lanchester attrition laws for combat prediction in StarCraft. In: Eleventh artificial intelligence and interactive digital entertainment conference
6. Uriarte A, Ontanón S (2015) Automatic learning of combat models for RTS games. In: Eleventh artificial intelligence and interactive digital entertainment conference

7. Stanescu M, Hernandez SP, Erickson G et al (2013) Predicting army combat outcomes in StarCraft. In: Ninth artificial intelligence and interactive digital entertainment conference
8. Uriarte A, Ontanón S (2018) Combat models for RTS games. IEEE Trans Games 10(1): 29–41
9. Wender S, Watson I (2012) Applying reinforcement learning to small scale combat in the real-time strategy game StarCraft: Broodwar. In: 2012 IEEE conference on computational intelligence and games (CIG). IEEE, pp 402–408
10. Sánchez-Ruiz AA, Miranda M (2017) A machine learning approach to predict the winner in StarCraft based on influence maps. Entertain Comput 19:29–41

Task Scheduling Strategy for Heterogeneous Spark Clusters

Yu Liang, Yu Tang(✉), Xun Zhu, Xiaoyuan Guo, Chenyao Wu, and Di Lin

University of Electronic Science and Technology of China, Shahe Campus: No. 4, Section 2, North Jianshe Road, 610054 Chengdu, Sichuan, People's Republic of China
yutang@uestc.edu.cn

Abstract. As a primary data processing and computing framework, Spark can support memory computing, interactive computing, and querying in a huge amount of data. Also, it can provide data mining, machine learning, stream computing, and the other services. However, the strategy of allocating resources among isomorphic processors cannot adapt to heterogeneous cluster environment due to its lack of load-based task scheduling. Therefore, we propose a dynamic load scheduling algorithm for heterogeneous Spark clusters by regularly collecting load information from each of the cluster node. Such an algorithm can dramatically reduce the allocation of load to the nodes which are already heavily loaded and in turn allocate more task to the idle nodes, thereby speeding up the process of job allocation in Spark. The experimental results show that the proposed algorithm can dramatically improve the computation efficiency by dynamically loading among the nodes in a heterogeneous cluster.

Keywords: Spark platform · Heterogeneous cluster · Load balancing · Task scheduling

1 Introduction

The low-latency processing of massive data is a new technical challenge in the era of big data. Spark [1] is a big data processing engine for in-memory computing, which enables Spark to provide faster data processing. Spark performs lazy computing [2] by manipulating elastic distributed datasets and adopts a simpler API to support more flexible computing operations, demonstrating better performance than Hadoop [3] in every respect.

Computers in a computer cluster have different hardware configurations, which make them different in performance in Spark jobs. The development of cloud computing and the use of data centers make clusters more heterogeneous [4]. The rise of machine learning has made clusters with computers with mixed CPU [5] and GPU [6] architectures, making clusters heterogeneous.

Because the computing power of each node in a heterogeneous cluster [7] is inconsistent, the same task allocation on different nodes will have different impact on the node load. The default Spark task scheduling method does not consider cluster

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 131–138, 2020.
https://doi.org/10.1007/978-981-15-0187-6_15

heterogeneity and lack of load-based task scheduling strategies. Therefore, this paper proposes a load scheduling algorithm that can consider cluster node heterogeneity and node resource dynamics.

2 Research Background

2.1 Spark Scheduling Strategy

The scheduling model in Spark refers to the process in which Spark writes the submitted jobs and data, how it is parsed, split and transformed by Spark, and how it is distributed to each computing node in the cluster for distributed computing and results. During the scheduling process, Spark's scheduling components and models are involved, as well as the job scheduling algorithms provided by Spark. Spark's job scheduling strategy is divided into two types: FIFO [8] (first in, first out) and FAIR (fair scheduling).

2.2 Task Scheduling Process

When scheduling Task to Executor, if you can fully understand the load of Spark cluster and introduce load balancing algorithm [8], you can make full use of low-load nodes, make the load of nodes in the cluster more even. More rational use of cluster computing resources can speed up the operation of the entire job.

In the above task scheduling process, there is no load-based scheduling process. There are many factors that cause this situation. There are two main points, namely resource reuse and container technology.

Whether a physical cluster deploys multiple computing architectures or uses container technology for resource isolation, it means that the operation of Spark in a production environment may share computing resources with other computing services. The operational load of other computing frameworks will definitely affect the Spark running tasks. If you can schedule tasks based on node load, Spark can still maintain high-performance computing services in complex hardware and software environments.

3 Spark Load Balancing Task Scheduling Strategy

3.1 Spark Load Definition

The process of Spark assigning Executors to Tasks is random. Introducing a load balancing algorithm here will optimize this process and speed up the progress of the job.

In order to sort according to the load, a calculation method of the node load is needed to judge the load level of each node. Because Spark is a computationally intensive and in-memory computing framework, the main computing resources used are CPU and memory. The CPU utilization, system task queue length, and memory utilization are usually selected to describe the load of the Spark program on the computer. The computer load can be expressed as Eq. (1).

$$\text{worker\_load} = \frac{(\text{cpu\_utilization} + \text{ memory\_utilization} + \text{ task\_competition})}{3} \tag{1}$$

Through the parameters, it can be calculated that the total time cpu_time of the effective utilization of the CPU is Eq. (2).

$$\text{cpu\_time} = \text{user} + \text{system} + \text{nice} + \text{idle} + \text{iowait} + \text{irq} + \text{softirq} \tag{2}$$

For the calculation method, see Eq. (3).

$$\text{cpu\_utilization} = 1 - \frac{\text{idle2} - \text{idle1}}{\text{cpu\_time2} - \text{cpu\_time1}} \tag{3}$$

How to calculate the memory utilization, see Eq. (4).

$$\text{memory\_utilization} = \frac{\text{mem\_total} - \text{mem\_free}}{\text{mem\_total}} \tag{4}$$

Whenever there is a Task that needs to be assigned an Executor, the minimum load Executor in the load-ordered queue is selected to be dequeued. If the Executor resource meets the running requirements of the Task, the Executor is assigned to the Task. The load update method is shown in Eq. (5).

$$\text{worker\_load} = (3 \times \text{old\_load} + \text{ new\_load})/4 \tag{5}$$

The preset load in the schedule is shown in Eq. (6).

$$\text{worker\_load} = \frac{\text{spark.task.cpus}}{\text{worker.cores}} \tag{6}$$

3.2 Load Balancing Task Scheduling

Algorithm 1 describes dynamic load-based task scheduling in the form of pseudocode.

Algorithm 1 Dynamic load task scheduling algorithm
Input: node load queue load_worker_queue, to be scheduled task set TaskSets
Output: Dispatch Task to Executor Steps:

```
(1) updateLoad(load_worker_queue)
(2) for(worker in load_worker_queue){worker.load/=zone.cpu_capacity}
(3) sortedByLoad(load_worker_queue)
```

```
sorted_executor_queue = sortedByWorker(load_worker_queue)
for(taskset in tasksets){
  for(LOCALITY in data_locality if !taskset.isEmpty){
    for(count in 1 to 2){
      for(executor in sorted_executor_queue if !taskset.isEmpty){
        while(canAllocateTask(executor)){
          if(count == 1 && executor.worker.load > spark.load.threshold){
            continue} task = taskAtLocality(taskset, LOCALITY, executor)
          taskset.remove(task)
          allocateTask(task, executor)
          executor.worker.load + = spark.task.cpus/executor.worker.cores
          executor.cores − = spark.task.cpus
          If(executor.cores < spark.task.cpus){sortedByLoad(load_worker_queue)
          sorted_executor_queue = sortedByWorker(load_worker_queue)}}}}}
}
```

4 Experiment and Result Analysis

4.1 Experimental Design

The purpose of the experiment is to verify that the task scheduling of the dynamic load schedules the task to the low-load node, makes the distribution in the task re-cluster as uniform as possible, and accelerates the job operation with the low-load node.

The experimental cluster provides four worker CPUs with four CPU cores. The maximum task concurrency of the program is 12. In the experiment, it is necessary to test whether the load scheduling has the ability to avoid high-load nodes and select the low-load Executor to perform tasks. The test plan is shown in Table 1.

Table 1. Experimental test program configuration table

Program name	Log handler		
Computer model	Use	Number of computers	Number of processor cores
Lenovo T530	Worker	4	4
ASUS X550V	Driver	1	4

The experimental node sequentially presets the load values shown in Table 2 to perform experiments. The reason why tests 1, 2 and tests 3, 4 choose two different tasks is because the degree of concurrency of 1, 2 is exactly the number of all CPU cores of the three Executors. In this case, Spark will take up all the cores for calculation. The concurrency of 9 is less than 12, which means that load scheduling can fill up to 4 Tasks on some Executors and only 4 Tasks on other Executors, depending on the load status of each computer.

Table 2. Experimental test plan table

Test number	Node 1 (%)	Node 2 (%)	Node 3 (%)	Node 4 (%)	Maximum task concurrency	Processing data volume (M)
1	0	0	0	0	12	384
2	0	0	90	0	12	384
3	0	0	0	0	9	288
4	90	30	20	90	9	288

4.2 Experimental Results and Analysis

Each set of tests has undergone multiple experiments and selected typical experimental cases for analysis. The "average time of the task" is the average execution time of the Task that takes the first stage of the job. This data is intended to illustrate the impact of the execution time of a single task under different load conditions. "Maximum task concurrency" refers to how many tasks are in parallel when the first stage is executed on the specified computer. This data is intended to illustrate the task load on the node. By analyzing the Task run and concurrency of the first stage, you can specifically analyze the relationship between node load and Task execution.

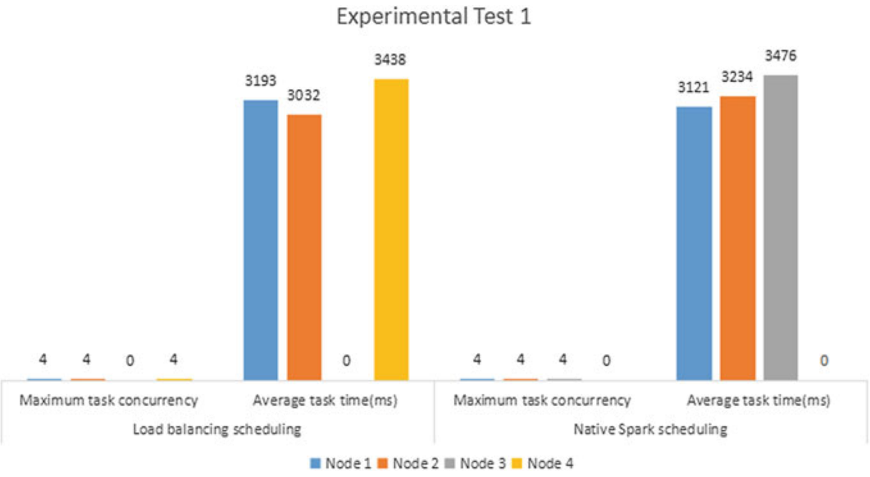


Fig. 1. Experimental test 1

Through experimental tests (Fig. 1), the load balancing scheduling operation time is 22.419 s, and the Spark scheduling operation time is 23.61 s. In test 1 (Fig. 1), the initial load of all nodes is 0%. The native Spark scheduling and load balancing scheduling in the above table yielded basically consistent runtime results. Although the two nodes select different nodes, the initial load on each node is empty, and the time performance of each running node is basically the same, indicating that the load scheduling and the original Spark scheduling can achieve the same result without any difference in load.

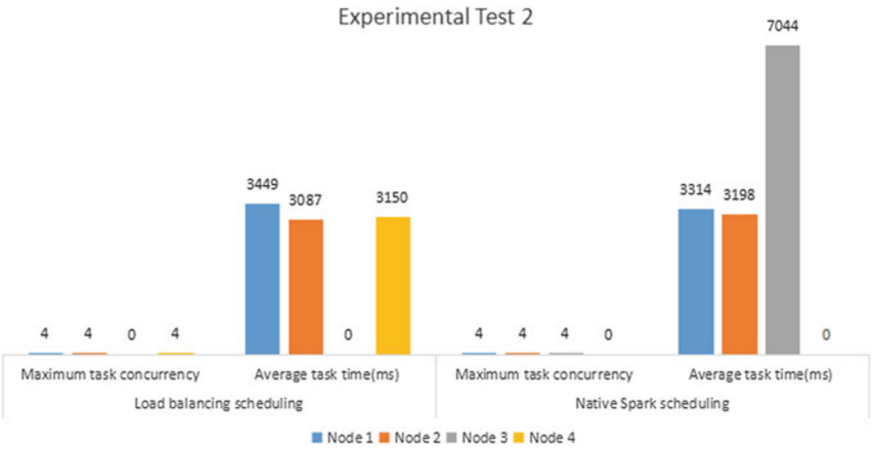


Fig. 2. Experimental test 2

Through experimental tests (Fig. 2), the load balancing scheduling operation time is 23.562 s, the Spark scheduling operation time is 30.17 s, and the actual optimization percentage is 22%. It shows that during the execution of the task, the load scheduling avoids the node 3 with high load, but the native Spark scheduling cannot do it, and the node 3 drags the execution progress of the native Spark.

Through experimental tests (Fig. 3), the load balancing scheduling operation time is 21.637 s, and the Spark scheduling operation time is 23.054 s. In test 3, although the nodes are all empty, the load balancing is faster than the running time of the native Spark schedule.

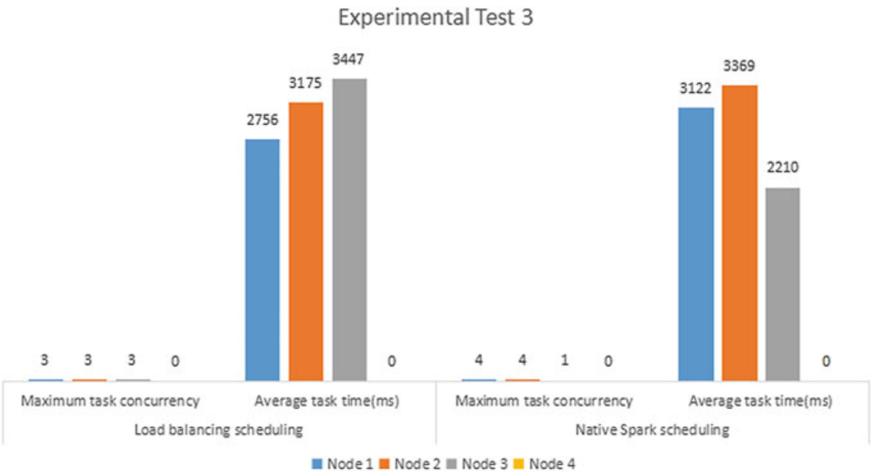


Fig. 3. Experimental test 3

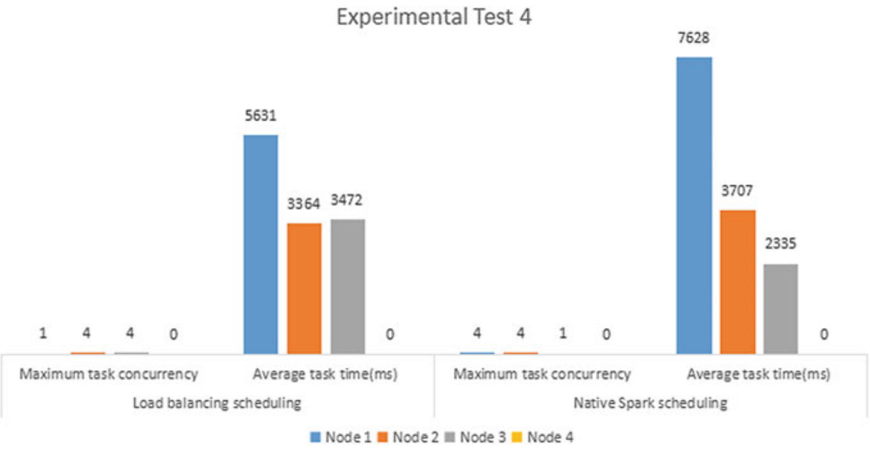


Fig. 4. Experimental test 4

Through testing (Fig. 4), the load balancing scheduling operation time is 25.922 s, the Spark scheduling operation time is 32.163 s, and the actual optimization percentage is 19%. In Test 4, each node has different workloads. Load balancing scheduling allocates different tasks on different nodes, and the result is better than native Spark scheduling.

The performance of the above four tests shows that dynamic load scheduling can avoid high-load nodes, allocate tasks according to the load on each node, and accelerate the execution speed of the job.

5 Conclusion

In a production environment, Spark may run under complex load environments due to resource isolation and resource reuse. Spark's task scheduling is not based on node load scheduling, which makes Spark unable to get the best performance under complex cluster load conditions. Based on the above problem, the dynamic load-based Spark task scheduling algorithm optimizes the task scheduling by using the dynamic load status of each node of the cluster to improve the overall running speed of the operation. The experimental results show that the dynamic load scheduling algorithm of heterogeneous Spark clusters gives users the ability to perform more detailed resource allocation for jobs. The task scheduling is more reasonable, the task assignment is more uniform, and the job execution speed is significantly improved.

Acknowledge. This work is jointly supported by the National Natural Science Foundation of China (No. 61601082, No. 61471100, No. 61701503, No. 61750110527)

References

1. Zaharia M, Chowdhury M, Franklin MJ et al (2010) Spark: cluster computing with working sets. In: Usenix conference on hot topics in cloud computing
2. Parsian MA (2015) Data algorithms: recipes for scaling up with Hadoop and Spark
3. Shvachko K, Kuang H, Radia S et al (2010) The Hadoop distributed file system. In: 2010 IEEE 26th symposium on mass storage systems and technologies (MSST). IEEE
4. Zhu X, He C, Li K et al (2012) Adaptive energy-efficient scheduling for real-time tasks on DVS-enabled heterogeneous clusters. J. Parallel Distrib. Comput. 72(6):751–763
5. Lee VW, Kim C, Chhugani J et al, Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. International
6. Nickolls J, Dally WJ (2010) The GPU computing era. IEEE Micro 30(2):56–69
7. Topcuouglu H, Hariri S, Wu MY (2002) Performance-effective and low-complexity task scheduling for heterogeneous computing. IEEE Trans Parallel Distrib Syst 13(3):260–274
8. Cardellini V, Colajanni M, Yu PS (1999) Dynamic load balancing on web-server systems. Internet Comput IEEE 3(3):28–39

Research on Multi-priority Task Scheduling Algorithms for Mobile Edge Computing

Yanrong Zhu, Yu Tang(✉), Chenyao Wu, and Di Lin

University of Electronic Science and Technology of China, Shahe Campus:
No. 4, Section 2, North Jianshe Road, Chengdu 610054, Sichuan,
People's Republic of China
yutang@uestc.edu.cn

Abstract. In this paper, priority-based mobile edge computing technology is used to decide the task migration problem on telemedicine devices. Tasks can be adaptively assigned to mobile edge servers or processed locally according to the current network situation and specific task processing environment, thus improving the efficiency and quality of telemedicine services. The main work of this paper is in three aspects. Firstly, priority is set for urgent tasks, and non-preemptive priority queues are set on the server side of MEC, so that the average stay time of each priority can be calculated according to the current situation of priority tasks queuing. Secondly, in the process of task transmission to the server, the channel resources are allocated adaptively by priority-based twice filtering strategy, and the final task migration decision is obtained by auction algorithm. Thirdly, compared with the existing mobile edge computing task migration model, the priority-based task migration model greatly guarantees the real-time and high quality of telemedicine.

Keywords: Mobile edge computing · Telemedicine · Task migration decision · Tasks priority · Auction algorithm

1 Introduction

With the development of computer technology and communication technology, telemedicine, which integrates medicine, computer technology, and communication technology, has gained more and more attention and application in the global health industry [1]. In telemedicine, doctors can predict the severity of patients' injuries through the Internet, so as to inform the medical staff in ambulances of the surgical pretreatment that should be carried out, thus greatly improving the survival rate of seriously injured patients. However, the development of telemedicine also has some difficulties because of the requirements for high-definition video and latency [8].

Mobile edge computing is one of the core standards of 5G, especially suitable for computing intensive, time-sensitive tasks, such as augmented reality, Internet of Vehicles, and other scenarios. With the advent of 5G era, mobile edge computing server (MEC server) will be deployed around the cell. Users will transmit information to the base station through wireless channels. The base station will connect with MEC

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 139–147, 2020.
https://doi.org/10.1007/978-981-15-0187-6_16

server through high-speed links and upload tasks to MEC server for processing [6]. The architecture model for mobile edge computing is shown in Fig. 1.

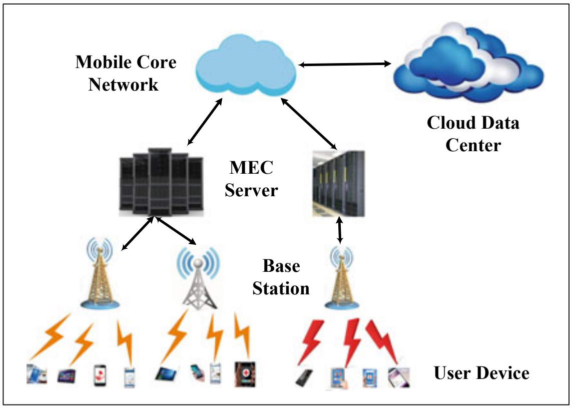


Fig. 1. Architecture of MEC

1.1 Related Work

In the field of MEC task migration, the main problems focus on the optimization of energy consumption, delay, and mobility of devices. Jiang et al. [3] proposed an energy-aware model to optimize the execution energy of mobile devices. Liu et al. [5] proposed the decision of task migration which made by game theory method. Seung-Woo et al. [7] constructed a spatial random network model with random node distribution and parallel computing. Zhang et al. [9] proposed an energy-aware unloading scheme to study the trade-off between energy consumption and delay. Zhang et al. [10] presented a green and low-latency mobile perception hierarchical framework, which reduces the energy cost of smart devices.

1.2 Motivation and Contributions

However, these models basically ignore the priority of task processing. In telemedicine, how to satisfy data transmission and processing services such as high-definition video in the shortest time so as to ensure the safety of patients' lives is a problem worthy of further study. This paper defines priority according to the urgency of mobile device tasks and provides priority registration mechanism and audit mechanism to ensure the authenticity and credibility of priority. Firstly, priority is set for urgent tasks, and non-preemptive priority queues are set on the server side of MEC according to the degree of urgency of tasks. Secondly, the channel resources are allocated adaptively by priority-based twice filtering strategy. Thirdly, compared with the existing MEC task migration model, our task migration model greatly guarantees the real-time and high quality of telemedicine.

2 Establishing Task Migration Model

There are two ways to perform tasks on the mobile end. One is to process directly at the mobile end, and the other is to put tasks on MEC server for processing and return the results to the local area. If the local processing time of a task is less than or equal to the time when it is transferred to the remote, then it is processed locally. Otherwise, the task is sent to remote processing. The decision is expressed by $\xi_{k,s}$:

$$\xi_{k,s} = \begin{cases} 0, & \text{task}\, k\, \text{don't translate by channels} \\ 1, & \text{task}\, k\, \text{computing in cloud by channels} \end{cases} \tag{2.1}$$

Suppose that the set of user devices is $\mathrm{DN} = \{1, 2, 3, \ldots, |\mathrm{DN}|\}$, the set of transmission channels is $\mathrm{CN} = \{1, 2, 3, \ldots, |\mathrm{CN}|\}$, the priority set is $\mathrm{PN} = \{1, 2, 3, \ldots, |\mathrm{PN}|\}$. For a task k, CW_k represents the workload (CPU cycles), DS_k represents the data size of the task, OD_k represents the output size after processing and PN_k represents the priority of the task. Task k can be represented by quadruples: $\Psi_k = \{\mathrm{CW}_k, \mathrm{DS}_k, \mathrm{OD}_k, \mathrm{PN}_k\}$.

2.1 Task Processing Time Locally

Let f_k^l represents the rate at which task k executes on the local device (CPU cycles/per sec), so the time required for task k to execute locally can be expressed as:

$$T_k^l = \frac{\mathrm{CW}_k}{f_k^l} \tag{2.2}$$

2.2 Task Processing Time in MEC Server

2.2.1 Transmission Delay of Tasks in Remote Processing

When a task is processed on the MEC server, the data transmission rate of task k transmitted through the channel s can be expressed as [3]:

$$\boldsymbol{R_{k,s}}(\xi) = \boldsymbol{w_s log_2}\left(1 + \frac{\boldsymbol{P_k^{tr} G_k}}{\boldsymbol{\sigma}^2 + \sum_{\boldsymbol{h \in DN\{i\}}:\xi_{\mathbf{h},s}=1,\boldsymbol{h \neq k}} \boldsymbol{P_h^{tr}} \cdot \boldsymbol{G_h}}\right) \tag{2.3}$$

where σ^2 is the transmission noise and G_k is the channel gain of the k-th device. In this case, we assume that in an open environment, $G_k = l^{-a}$, where l is the distance between the device and the base station, a is the path fading factor, and l^{-a} is the fading power with distance. Therefore, the transmission delay of task k through channel s is:

$$t_{k,s}^{\mathrm{tr}} = \frac{\mathrm{DS}_k}{R_{k,s}} \tag{2.4}$$

2.2.2 Task Stay Time on MEC Server

When task k is transmitted to MEC server through channel s for processing, queuing will begin when the number of tasks is increasing. Therefore, the duration of a task on the MEC server is the sum of queuing delay and processing delay.

$$T_k^{\text{queue}} = t_k^w + t_k^c \tag{2.5}$$

In the non-preemptive priority queuing model [4], we assume that the model is $M/M/N/\infty$. Task arrival obeys Poisson distribution with parameter λ_j, and service time obeys negative exponential distribution with parameter μ. The proportion of arrival rate can be obtained by detection program. The number of MEC servers is |SN|. Because of the extra memory space, the storage capacity can be regarded as infinite, and the total number of priorities is |PN|. The specific queuing system model is shown in Fig. 2.

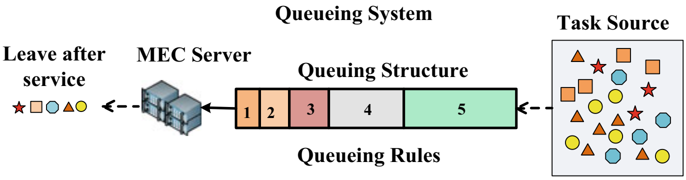


Fig. 2. Queue model based on multi-priority

In this paper, we use emergent (level-1), urgent (level-2), semi-urgent (level-3), nonurgent (level-4) to classify the severity of injury and plus the urgency of ordinary tasks (level-5). We divide the priority into 1–5 levels. If we make λ_j to represent the arrival rate of the j-th priority, then the total arrival rate λ can be expressed as:

$$\lambda = \sum_{i=1}^{|\text{PN}|} \lambda_i \tag{2.6}$$

From this, we can calculate the average stay time of each priority task as $T_{\text{pri}}^{\text{queue}}$:

$$\boldsymbol{T_{pri}^{queue}} = \frac{1}{\boldsymbol{HJ_{pri}J_{pri-1}}} + \frac{1}{\boldsymbol{\mu}},\ \forall \boldsymbol{pri} \in \boldsymbol{PN} \tag{2.7}$$

$$\boldsymbol{H} = \boldsymbol{SN}!* \frac{\boldsymbol{SN} * \boldsymbol{\mu} - \boldsymbol{\lambda}}{\left(\frac{\boldsymbol{\lambda}}{\boldsymbol{\mu}}\right)^{\boldsymbol{SN}}} * \sum_{\boldsymbol{j}=0}^{\boldsymbol{SN}-1} \frac{\left(\frac{\boldsymbol{\lambda}}{\boldsymbol{\mu}}\right)^{\boldsymbol{j}}}{\boldsymbol{j}!} + \boldsymbol{SN} \cdot \boldsymbol{\mu} \tag{2.8}$$

$$\boldsymbol{J_{pri}} = \begin{cases} 1, & \boldsymbol{pri} = 0 \\ 1 - \frac{\sum_{\boldsymbol{j}=1}^{\boldsymbol{pr_i}} \boldsymbol{\lambda_j}}{\boldsymbol{SN \cdot \mu}}, & \boldsymbol{pri} \geq 1 \Cap \text{pri} \leq |\text{PN}| \end{cases} \tag{2.9}$$

St.

$$\sum_{i=1}^{pr_i} \lambda_i < \mathrm{SN} \cdot \mu, \mathrm{pri} \in \mathrm{PN} \tag{2.10}$$

2.2.3 Total Task Delay on MEC Server

The total time includes the transmission delay of the uploaded task $t_{k,s}^{\mathrm{tr}}$, queuing delay t_k^w and processing delay t_k^c, and the delay of the calculated results t_k^r (can be neglected). Thus, the total latency $T_{k,s}^{\mathrm{off}}$ required for processing on the MEC server is [2]:

$$T_{k,s}^{\mathrm{off}} = t_{k,s}^{\mathrm{tr}} + t_k^w + t_k^c + t_k^r = t_{k,s}^{\mathrm{tr}} + T_{\mathrm{pri}}^{\mathrm{queue}} + t_k^r \cong \frac{\mathrm{DS}_k}{R_{k,s}} + T_{\mathrm{pri}}^{\mathrm{queue}}. \tag{2.11}$$

2.3 Task Migration Decision Algorithm

2.3.1 First Filtering Algorithm

The time consumed to restrict the processing of a task on the MEC server consists of two parts, one is the stay delay, and the other is the transmission delay [7]. If the time of remote processing of a task in the exclusive channel is longer than that of the task in the local processing, it will be processed directly in the local processing. This is the first filtering strategy. Thus, the initial filtering strategy can be expressed as:

$$\mathcal{F} = \begin{cases} k \to \mathbb{Z}_l, & T_k^l \le T_{k,s}^{\mathrm{off},0} \\ k \to \mathbb{Z}_r, & T_k^l > T_{k,s}^{\mathrm{off},0} \end{cases} \tag{2.12}$$

When the task is in an exclusive channel, the transmission rate is:

$$R_{k,s}^0 = w_s \log_2\left(1 + \frac{P_k^{\mathrm{tr}} G_k}{\sigma^2}\right) \tag{2.13}$$

The specific algorithm description is shown in Table 1.

2.3.2 Second Filtering Algorithm Based on Channel Transport Priority

For the remaining equipment decision-making, the auction algorithm is used to allocate the channel, and the channel resources are obtained first with high priority. If a task can be remotely processed for less than local time, and the remaining channel resources can be obtained, then the task will be remotely processed. A schematic diagram of channel allocation is shown in Fig. 3. The second filtering algorithm is shown in Table 2.

3 Simulation Result

This paper mainly compares with the following experiments:

Table 1. First filtering algorithm description

```
the first filtering algorithm
Input $\mathbb{Z}$; Output $\mathbb{Z}_l$,$\mathbb{Z}_r$;Set $\mathbb{Z}^{temp} = \mathbb{Z}$, $\mathbb{Z}_l = \mathbb{Z}_r = \emptyset$, $\xi = \emptyset$;
While $\mathbb{Z}^{temp} \neq \emptyset$ do
  Select the device k, where k$\in \mathbb{Z}^{temp}$;
  for channel s=1 to |SN| do  update $R_{k,s}^{0}$ end for
  select the s where s=argmax$\{R_{k,s}^{0}\}$
  calculate  $T_{k,s}^{off,0} = \frac{DS_k}{R_{k,s}^{0}} + T_{pri}^{queue}$ and $T_k^l = \frac{CW_k}{f_k^l}$
  if $T_{k,s}^{off,0}$ >= $T_k^l$
      $\mathbb{Z}_l$ += $k$ , $\xi_{k,s} = 0$ , $\forall s \in |CN|$
  else $\mathbb{Z}_r$ += $k$ ; end if
  $\mathbb{Z}^{temp} = \mathbb{Z}^{temp} \backslash$k
end while
```

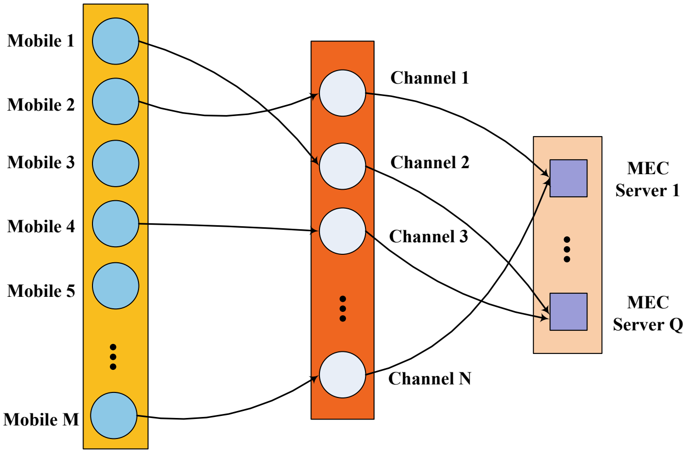


Fig. 3. Priority-based channel allocation

(a) Local execution of all tasks;
(b) All tasks are executed on the MEC server and are based on task priority;
(c) All tasks are executed on the MEC server and are not based on task priority.

3.1 Average Execution Time of Different Algorithms with High Priority

Firstly, we compare the average execution time of various algorithms for high-priority tasks as the number of tasks increases. As can be seen from Fig. 4, the average execution time of high priority increases slowly. Algorithms (a), the average execution time of high priority is larger than that of priority-based task migration model from the beginning. Algorithms (b), although the high-priority delay increases, it is not very

Table 2. Second filtering algorithm description

Priority-based Channel Auction Algorithms

```
Input $\mathbb{Z}_r$; Output: $\xi_{k,s}$; Set $\mathbb{Z}' = \mathbb{Z}_r$, $bd = \emptyset$;
While $\mathbb{Z}' \neq \emptyset$ do
    Select the device k, where k 's priority = argmin{PN},k$\in \mathbb{Z}'$;
    for channel s=1 to SN do update $R_{k,s}$ and $T_{k,s}^{off}$;
      if $g_s - \Sigma_{r=1,r\neq k}^{DN} \xi_{r,s} \cdot P_k^{tr} \cdot G_k > 0$ $then$ calculate the bid density $bd_s$
            based on the 2-tuple <$b_s, c_s$>,where $b_s = T_{k,s}^{off}$,$c_s = P_k^{tr} G_{k,s}$
      $bd_s = \frac{\left(g_s - \Sigma_{r=1,r\neq k}^{DN} \xi_{r,s} \cdot P_r^{tr} \cdot G_r\right) \cdot T_{k,s}^{off}}{\sqrt{g_s - \Sigma_{r=1,r\neq k}^{DN} \xi_{r,s} \cdot P_r^{tr} \cdot G_r}}$ ,$g_s - \Sigma_{r=1,r\neq k}^{DN} \xi_{r,s} \cdot P_r^{tr} \cdot G_r > 0$
      else set $\xi_{k,s} = 0$; end if; end for
    While bd $\neq \emptyset$ do
        Select channel h where h=argmin{bd$_s$}
        if $T_{k,h}^{off} < T_k^l$ then
            let $\xi_{k,h} = 1$; $g_h = g_h - \Sigma_{r=1,r\neq k}^{DN} \xi_{r,h} \cdot P_r^{tr} \cdot G_r$;
            for $bd_j$ in bd do let $\xi_{k,j} = 0$;bd=bd\bd$_j$ ; end for
            break;
        else let $\xi_{k,h} = 0$; end if
        bd=bd\bd$_h$
    end while
$\mathbb{Z}' = \mathbb{Z}' \backslash k$
end while
```

drastic. Algorithm (c), the tasks are not prioritized and all are executed on the MEC server, resulting in high-priority tasks waiting for a long time. Our algorithm is dynamically selected through the current network situation, which has the best performance effect.

Notice the intersection of algorithm (a) and algorithms (b) and (c), and when the number of devices reaches a certain upper limit, all tasks take longer to execute remotely than tasks execute locally. From the formula, we can easily analyze the reasons.

$$T_k^l = t_{k,s}^{\mathrm{tr}} \Rightarrow \frac{\mathrm{CW}_k}{f_k^l} - \frac{\mathrm{CW}_k}{f_k^c} = \mathrm{CW}_k * \frac{f_k^c - f_k^l}{f_k^l f_k^c} = \frac{\mathrm{DS}_k}{R_{k,s}} + t_k^w \tag{3.1}$$

3.2 Average Execution Time for Different Priorities

Secondly, we investigate the changes in the average execution time of tasks under different priorities, as shown in Fig. 5. Algorithm (a), the processing time of priority tasks is almost the same and must be greater than our algorithm. Algorithm (b), for those with high priority the time consumed is less than executed locally, and it takes

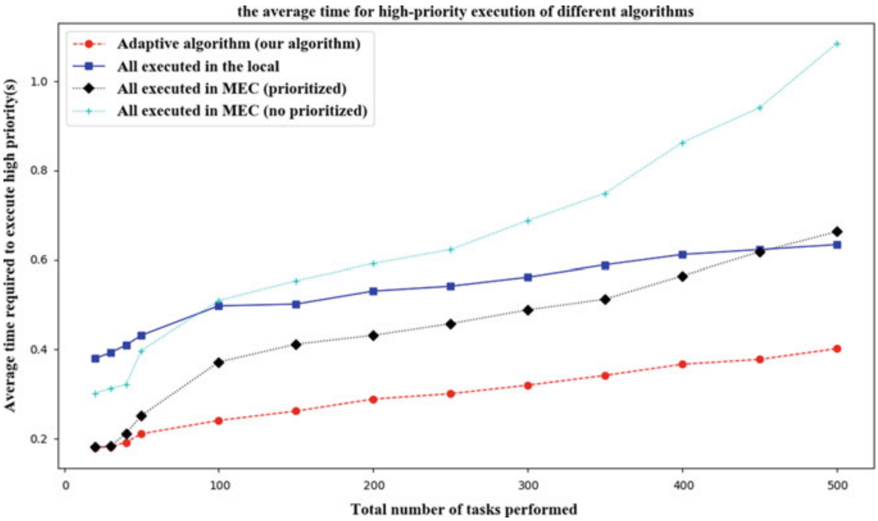


Fig. 4. Average execution time of high-priority tasks with increasing number of tasks

longer than the adaptive algorithm. For ordinary tasks, the execution time is higher than executed locally. Our algorithm adopts priority mechanism and dynamic assignment algorithm, which fully guarantees the real-time performance of high-priority tasks.

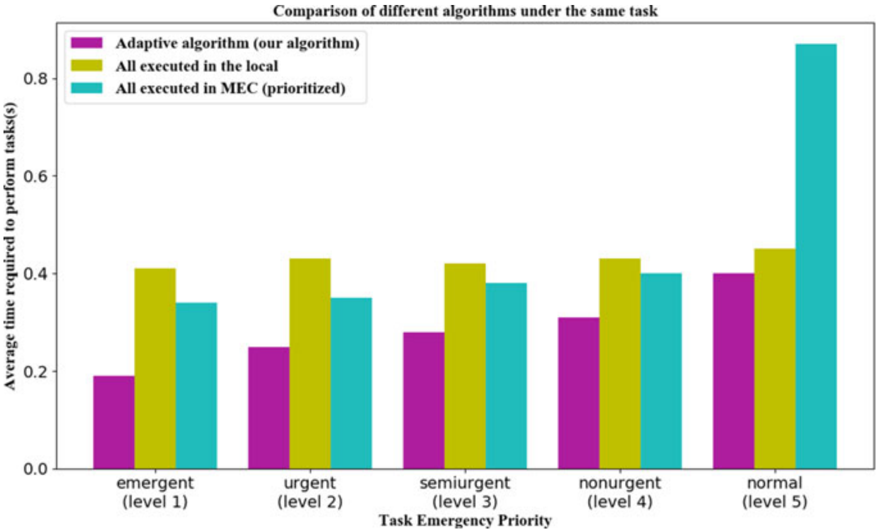


Fig. 5. Average time required for algorithms to execute under different task priorities

4 Conclusion

The priority-based mobile edge computing task migration model greatly improves the processing efficiency and quality of emergency tasks. Compared with the general migration model without task priority, it has great advantages and can be applied in many fields and industries such as telemedicine. At the same time, there are still some areas to be improved, such as bringing the energy model of ordinary tasks into consideration, setting the capacity of MEC servers to a limited situation, considering the mobility management of mobile device, and so on.

Acknowledgements. This work is jointly supported by the National Natural Science Foundation of China (No. 61601082, No. 61471100, No. 61701503, No. 61750110527).

References

1. Bourne PE, Lee KC, Ma JD et al (2011) Telemedicine, genomics and personalized medicine: synergies and challenges. Curr Pharmacogen Pers Med (Former Curr Pharmacogen) 9(1)
2. Chen X, Jiao L, Li W et al (2015) Efficient multi-user computation offloading for mobile-edge cloud computing. IEEE/ACM Trans Netw 24(5):2795–2808
3. Jiang F, Zhang X, Peng J et al (2018) An energy-aware task offloading mechanism in multiuser mobile-edge cloud computing. Mob Inf Syst 2018(4):1–12
4. Kamoun F (2008) Performance analysis of a non-preemptive priority queuing system subjected to a correlated Markovian interruption process. Comput Oper Res 35(12):3969–3988
5. Liu CF, Bennis M, Poor HV (2017) Latency and reliability-aware task offloading and resource allocation for mobile edge computing
6. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun Surv Tutor 99:1-1
7. Seung-Woo K, Kaifeng H, Kaibin H (2018) Wireless networks for mobile edge computing: spatial modeling and latency analysis. IEEE Trans Wirel Commun 1-1
8. Wootton R (1997) Telemedicine: the current state of the art. Minim Invasive Ther Allied Technol 6(5–6):393–403
9. Zhang J, Hu X, Ning Z et al (2017) Energy-latency trade-off for energy-aware offloading in mobile edge computing networks. IEEE Internet Things J 1-1
10. Zhang K, Leng S, He Y et al (2018) Mobile edge computing and networking for green and low-latency internet of things. IEEE Commun Mag 56(5):39–45

Microblog Rumor Detection Based on Comment Sentiment and CNN-LSTM

Sheng Lv, Haijun Zhang(✉), Han He, and Bingcai Chen

School of Computer Science and Technology, Xinjiang Normal University,
Urumqi 830054, China
zhjlp@163.com

Abstract. Traditional rumor detection methods, such as feature engineering, are difficult and time-consuming. Moreover, the user page structure of Sina Weibo includes not only the content text, but also a large amount of comment information, among which the sentimental characteristics of comment are difficult to learn by neural network. In order to solve these problems, a rumor detection method based on comment sentiment and CNN-LSTM is proposed, and long short-term memory (LSTM) is connected to the pooling layer and full connection layer of convolutional neural network (CNN). Meanwhile, comment sentiment is added to rumor detection model as an important feature. The effectiveness of this method is verified by experiments.

Keywords: Sentimental analysis · Deep learning · Rumor detection · Word vector

1 Introduction

With the rise of social networks, Sina Weibo has become the largest social network information platform in China. According to the third quarter of Sina Weibo 2017, the monthly active users exceed 376 million, the daily active users exceed 165 million, and hundreds of millions of microblogs are published every day. According to China's new media development report, 60% of Internet rumors first appeared in Sina Weibo platforms, which has become the largest media for the dissemination of false information. Sina Weibo provides information services such as publishing, sharing, and learning, and users can get and share information only by moving their fingers. Therefore, the harm and impact of Sina Weibo rumors are even greater. Rumor detection can serve the task of rumor clearance, such as prevention and supervision, but at present, we still rely on manual reporting to dispel rumors. How to extract rumor features and design an effective rumor detection method become more and more significant.

This paper proposes a rumor detection method based on comment sentiment and CNN-LSTM. The remainder of this paper is arranged as follows: The second section introduces the related work of rumor detection. In the third section, the rumor detection method based on comment sentiment and CNN-LSTM is introduced. The fourth part gives experiments and discussions. Finally, the conclusions and future works are presented.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 148–156, 2020.
https://doi.org/10.1007/978-981-15-0187-6_17

2 Related Works

Nowadays, most researchers regard rumor detection as text classification [1] and design different features in the framework of classification, mainly considering the features of text content, communication structure, and credibility.

The first method detects rumors based on text features of microblog. Qazvinian et al. [2] detect rumors on Bayesian classifier by selecting Twitter text features, user features, and related character label features. Sun et al. [3] have studied the major rumors on the microblog platform. They extract features from the microblog text, microblog video pictures, and microblog users and use machine learning algorithms such as Bayesian and decision tree to carry out experiments.

The second method detects rumors based on the characteristics of communication structure. Takahashi et al. [4] collected comments from users on Twitter after the tsunami in Japan and found that we can find features from the growth of rumor microblogs and forwarding rate. Nourbakhsh et al. [5] collected hundreds of rumors and studied their semantic features. At the same time, the influence of different user roles on rumor communication structure was studied. Mendoza et al. [6] studied the structure of rumors and user behavior under the topic of “Chile earthquake” on Twitter. They found that rumors are more intense and faster than normal news discussions.

The third method detects rumors based on credibility. Gupta et al. [7] studied hot events and found that there was more false information about hot events on Twitter. Suzuki et al. [8] designed a method to evaluate the credibility of Twitter information and judged the credibility of Twitter information by the retention rate of Twitter content forwarding. If the original Twitter message remains almost unchanged after forwarding, the credibility will be higher; on the contrary, it will be lower.

However, up to now, the traditional machine learning method [9] relies on feature engineering, which is time-consuming and laborious. Common in-depth learning model also has good performance, but it cannot learn deeper features, such as the sentimental characteristics of Weibo. In order to solve the above problems, combining CNN and LSTM to extract the basic features of rumors and utilizing the sentimental analysis technology to mine sentimental features can detect rumor effectively [10].

3 Rumor Detection Based on Comment Sentiment and CNN-LSTM

3.1 CNN-LSTM

The CNN-LSTM is a fusion model, which combines the advantages of the two models. The corresponding vectors are generated by word vector technology after text data pretreatment. First, the convolution layer of CNN [11] is used to extract local features, then the pooling layer is used to reduce the parameter dimension, then the LSTM [12] layer is entered, and then the full connection layer and Softmax layer are used to realize the classification output. The structure of the CNN-LSTM model is shown in Fig. 1.

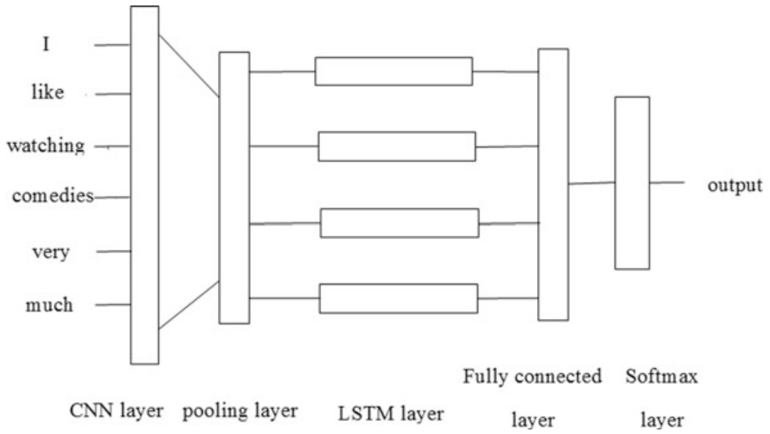


Fig. 1. CNN-LSTM model structure

3.2 Excavating Sentimental Features of Comments

With the liberalization of microblog language, it is suitable to use simple and effective methods of deep learning to classify, that is, to construct an LSTM-based sentimental classifier for sentimental analysis of comments.

LSTM sentimental classifier is used to mark the sentimental tendencies of comment data in rumor microblog and non-rumor microblog in corpus. The final results are shown in Table 1.

Table 1. Sentimental markers of comments

	Positive (%)	Negative (%)
Rumor comments	33.9	66.1
Non-rumor comments	48.3	51.7
All comments	41.1	58.9

According to the sentimental tendency of the comment, the sentimental intensity of the whole comment is calculated, and the formula is as follows:

$$\text{text}_{\text{sentiment}} = \frac{n_{\text{pos}} - n_{\text{neg}}}{N} \tag{1}$$

Among them, n represents the number of positive (negative) comments, and N represents the total number of comments. When the sentimental value is between $[-1, 1]$, when it is greater than 0, it indicates that the overall sentimental of microblog comments is positive, and vice versa, negative. The greater the absolute value, the stronger the positive (negative) sentiment of microblog comments.

3.3 Incorporating Sentimental Polarity into CNN-LSTM

The specific workflow of CNN-LSTM rumor detection model integrating sentimental polarity is as follows:

(1) Construct microblog corpus, and the data set is divided into rumor microblog and normal microblog.

(2) Cleaning data; filtering garbage comments and word segmentation; and training word vector model to vectorize microblog text.

(3) Sentimental analysis of microblog text. The formula for calculating the sentimental variance of microblog comments is as follows:

$$\text{sentiment}_{\text{variance}} = \frac{1}{n}\sum_{i=1}^{n}\left(\text{text}_{\text{sentiment}} - s\right)^2 \tag{2}$$

(4) Constructing the sentimental feature set of microblog comments and the sentimental difference is calculated according to the sentimental intensity and variance of microblog comments. The formula for calculating sentimental differences is as follows:

$$\text{sentiment}_{\text{diff}} = \alpha|\text{text}_{\text{sentiment}} - a| + \beta|\text{sentiment}_{\text{variance}} - b| \tag{3}$$

(5) Training a rumor detection model based on CNN-LSTM, and predicting the text as the probability of rumors (and non-rumors).

(6) When the sentimental difference and the predicted results of CNN-LSTM are substituted into the rumor calculation formula, if the calculation results are larger than the threshold value, then microblog is a rumor; otherwise, it is not a rumor. $\text{Rumor}_{\text{predict}}$ represents rumor detection value, λ and γ are weighting parameters, and the values are 0.4 and 0.6. Then the rumor calculation formula is as follows:

$$\text{rumor}_{\text{predict}} = \lambda * \text{sentiment}_{\text{diff}} + \gamma * \text{cnn} - \text{lstm}_{\text{predict}} \tag{4}$$

Rumor detection threshold is set to K. If $\text{rumor}_{\text{predict}} > K$, it is judged as a rumor; otherwise, it is judged as a non-rumor.

The model framework is shown in Fig. 2.

4 Experiments and Analysis

4.1 Data Set and Spam Comment Filtering

Directly crawl the fake Weibo information published by Weibo Community Management Center, and collect the name of the rumor microblog reporter, the name of the publisher, and the comments and so on. Then crawl the same number of normal microblogs to ensure the quality of information by limiting the information published by at least blue V users. Finally, 10,229 rumors and 10,200 non-rumors were collected as rumor training sets, coae2014 task 4 sentimental data set as sentimental training sets.

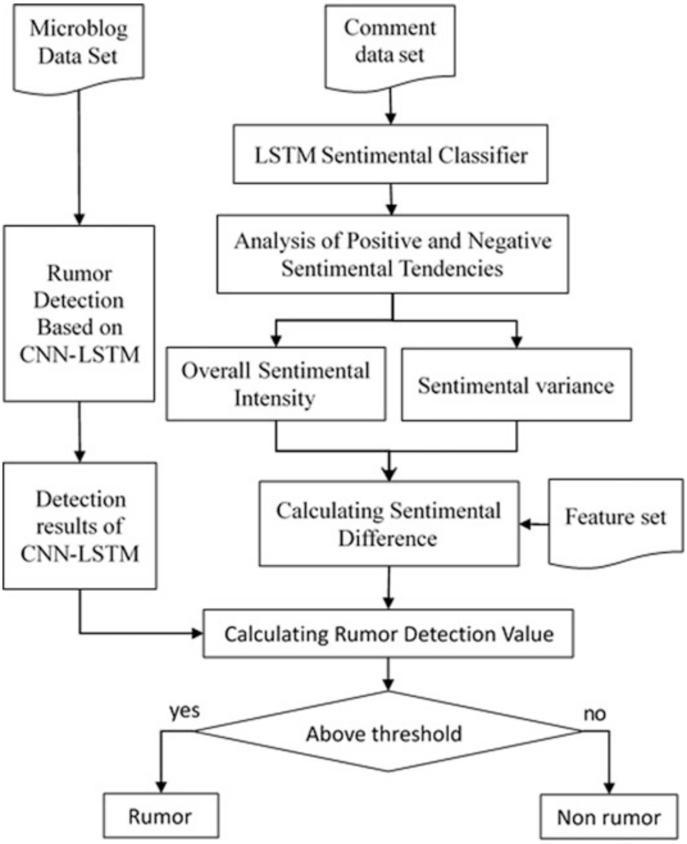

Microblog Data Set
Comment data set
LSTM Sentimental Classifier
Analysis of Positive and Negative Sentimental Tendencies
Rumor Detection Based on CNN-LSTM
Overall Sentimental Intensity
Sentimental variance
Detection results of CNN-LSTM
Calculating Sentimental Difference
Feature set
Calculating Rumor Detection Value
yes
Above threshold
no
Rumor
Non rumor

Fig. 2. CNN-LSTM rumor detection model fusing sentimental polarity

The test set contains 300 suspected rumor microblogs and 500 related comments for each suspected rumor microblog.

Spam comment generally has a low similarity with the original microblog in content and is often a declarative sentence, with a high proportion of noun. Therefore, it can be filtered according to the characteristics of similarity and noun proportion.

4.2 Experimental Result

When the rumor detection threshold is 0.3, the result of rumor detection is shown in Table 2.

The experimental results of the evaluation criteria are shown in Table 3.

Table 2. Rumor detection results

	Rumor	Non-rumor
Classified as rumors	182	10
Classified as non-rumors	12	96

Table 3. Experimental results of various evaluation indicators of sentimental CNN-LSTM model

Model	Recall	F1	Accuracy
Sentimental CNN-LSTM	0.9381	0.9430	0.9266

4.3 Comparative Analysis

4.3.1 Effect of Rumor Detection Threshold on Experimental Results

In [0.05, 0.8] interval, rumor detection threshold are divided into 16 grades. The experimental results are shown in Fig. 3.

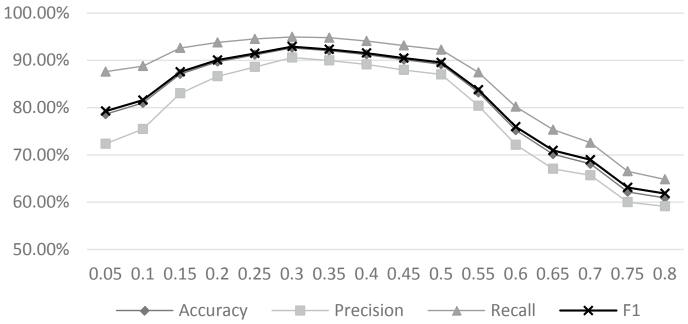


Fig. 3. Effect of rumor detection threshold on experimental indicators

If the threshold of rumor detection is too high, the model will easily misjudge rumors as true, and while the threshold of rumor detection is too low, the model will easily misjudge truth as rumors. When the threshold of rumor detection is 0.3, the accuracy of the model reaches 92.66%.

4.3.2 The Effect of Adding Sentimental Characteristics on the Experiment

The Glove model was used, and the word vector was 250 dimensions. The experimental results of CNN-LSTM are shown in Fig. 4.

After adding sentimental characteristics, rumor detection threshold is 0.3. The model comparison results are shown in Fig. 5.

Figure 5 shows that the accuracy of rumor detection increases by 1.28%, F1 value increases by 2.62%, and recall rate decreases slightly when sentimental features are

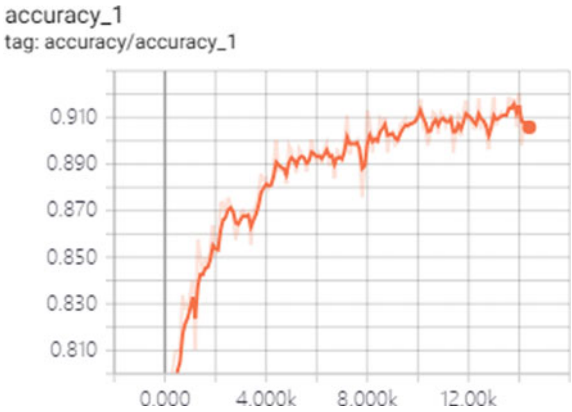


Fig. 4. Change of accuracy in CNN-LSTM training

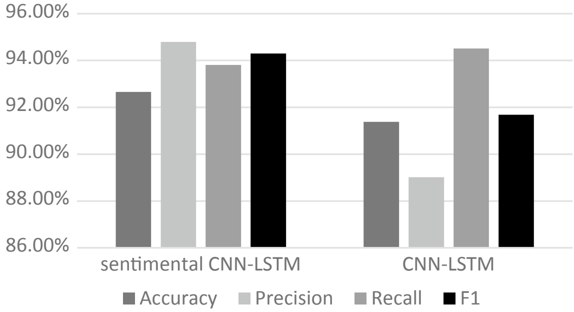


Fig. 5. Influence of sentimental features on model performance

added to the model. The experimental data show that adding comment sentimental features can improve the rumor detection effect of the model.

4.3.3 Experimental Comparison of Sentimental CNN-LSTM with Other Models

This method is compared to traditional machine learning methods such as Naive Bayes, SVM classifier model, and common deep learning model. The word vector model is Glove, the word vector dimension is 250, and the rumor detection threshold is 0.3. The experimental results are shown in Table 4.

From Table 4, we can see that the sentimental CNN-LSTM constructed in this paper has the best effect in the comparative experiments of the above seven models. The method based on deep learning is obviously better than traditional machine learning method, and the performance of CNN-LSTM in deep learning method is slightly better than other models. The validity of the sentiment CNN-LSTM proposed in this paper is proved by comparative experiments.

Table 4. Comparison of evaluation indicators among different models

Model	Recall	F1	Accuracy
Naive Bayes	0.7995	0.7701	0.7631
SVM	0.8252	0.7986	0.7922
GRU	0.9047	0.8837	0.8796
CNN	0.9262	0.8893	0.8854
LSTM	0.9209	0.8817	0.8775
CNN-LSTM	0.9451	0.9168	0.9138
Sentimental CNN-LSTM	0.9381	0.9430	0.9266

5 Conclusion and Future Works

Based on convolution neural network and cyclic neural network, this paper constructs a CNN-LSTM, which not only retains the local feature extraction ability of CNN, but also has the “memory” ability of LSTM. Because the rumors in Weibo have the characteristics of language exaggeration and obvious sentiment, this paper focuses on exploring the sentimental characteristics of microblog comments and proposes a CNN-LSTM integrating sentimental polarity. The experimental results show that the accuracy of the model reaches 92.66%, which is better than the common neural network model and machine learning model, and proves the validity of the model. However, the method of this paper is just suitable for specific areas. In future work, we will consider how to design a new model to adapt to rumor information detection in more areas.

Acknowledgements. This work is supported by Xinjiang Joint Fund of National Science Fund of China (U1703261). We thank Tsinghua University, Sina Weibo platform, and Sogou Laboratory for their open resources.

References

1. Yang Y, Liu X (1999) A re-examination of text categorization methods. In: International Acm Sigir conference on research & development in information retrieval. ACM
2. Qazvinian V, Rosengren E, Radev DR et al (2011) Rumor has it: identifying misinformation in microblogs. In: Conference on empirical methods in natural language processing, EMNLP 2011, 27–31 July 2011, John Mcintyre Conference Centre, Edinburgh, UK, A Meeting of Sigdat, A Special Interest Group of the ACL. DBLP, pp 1589–1599
3. Sun S, Liu H, He J, Du X (2013) Detecting event rumors on sina weibo automatically. In: Web technologies and applications. Springer, pp 120–131
4. Takahashi T, Igata N (2012) Rumor detection on twitter. In: Joint, international conference on soft computing and intelligent systems, pp 452–457
5. Nourbakhsh A, Liu X, Shah S et al (2015) Newsworthy rumor events: a case study of twitter. In: IEEE international conference on data mining workshop. IEEE, pp 27–32
6. Mendoza M, Poblete B (2011) Twitter under crisis: can we trust what we RT?. In: Proceedings of the first workshop on social media analytics, pp 71–79

7. Gupta A, Kumaraguru P (2012) Credibility ranking of tweets during high impact events, pp 2–8
8. Suzuki Y (2011) A credibility assessment for message streams on microblogs. In: 2010 international conference on P2P, parallel, grid, cloud and internet computing. IEEE
9. Dayani R, Chhabra N, Kadian T et al (2016) Rumor detection in twitter: an analysis in retrospect. In: IEEE international conference on advanced networks & telecommuncations systems. IEEE
10. Li D, Qian J (2016) Text sentiment analysis based on long short-term memory. In: IEEE international conference on computer communication & the internet. IEEE
11. Kim Y (2014) Convolutional neural networks for sentence classification. Eprint Arxiv
12. Roy A, Basak K, Ekbal A et al (2018) A deep ensemble framework for fake news detection and classification

A Guideline for Object Detection Using Convolutional Neural Networks

Xingguo Zhang, Guoyue Chen(✉), Kazuki Saruta, and Yuki Terata

Akita Prefectural University, 84-4 Aza Ebinokuchi Tsuchiya, Yurihonjo City
015-0055, Japan
chen@akita-pu.ac.jp

Abstract. The main purpose of object detection is to detect and locate specific targets from images. The traditional detection methods are usually complex and require prior knowledge of the detection target. In this paper, we will introduce how to use convolutional neural networks to perform object detection from image. This is one of the important areas of computer vision. In order to build up to object detection, we first learn about how we can get the object localization or landmark by a neural network. And then I will give the detail of sliding windows detection algorithm and introduce how to use the convolutional implementation of sliding windows to speed up the process. Then we will introduce the transfer learning and how to prepare your own learning data for training networks.

Keywords: Object detection · Bounding box · Transfer learning

1 Introduction

The main purpose of object detection is to detect and locate specific targets from images. The traditional detection model usually represents the target object by handcraft features and then predicts the category and location by classifier [1]. These methods are usually intuitive and easy to understand. However, the design of these methods is often complex and requires prior knowledge of the detection target. In addition, it is highly dependent on specific tasks and has poor portability. Once the detection target changes significantly, it is necessary to redesign the algorithm.

In recent years, with the improvement of hardware and algorithm, convolutional neural networks (ConvNet) have achieved great success in image classification, which led researchers to study its effects in other areas of computer vision. The early algorithm based on deep learning is generally divided into three steps, selecting the object candidate region (proposal), extracting the features of the candidate region, and finally putting the extracted features into the classifier to predict the object category. In 2014, Girshick designed the R-CNN model [2] based on ConvNet. The mean average precision (mAP) of this model in the object detection task of PASCAL VOC [3] was 62.4%, which was nearly 20% higher than the traditional algorithm. He et al. proposed Spatial Pyramid Pooling Net (SPP net) [4], which only performed convolution operation on the whole picture once and added pyramid pooling layer after convolution layer, so as to fix the feature map to the required size. This greatly saves time, reducing

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 157–164, 2020.
https://doi.org/10.1007/978-981-15-0187-6_18

the processing time of single-frame image to 2.5 s. However, SPP net failed to optimize the hard disk storage space of R-CNN. To address the issue, Girshic proposed the Fast R-CNN [5].

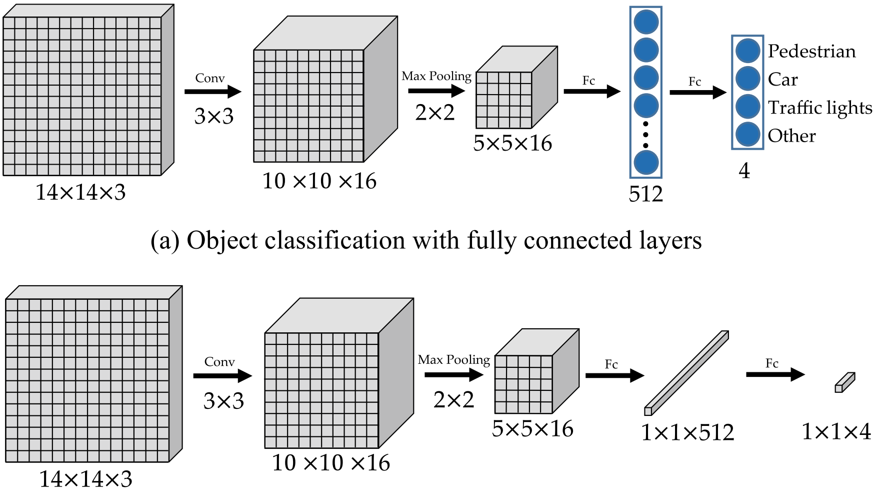
Fast R-CNN proposed a multitask loss function, which adds the loss of target positioning to the traditional loss function to correct the position information. After that, Ren et al. designed the structure of Faster R-CNN [6] and used the ConvNet to generate candidate regions directly. Faster R-CNN realizes end-to-end training and realizes real-time detection. On this basis, some improved methods such as region-based fully convolutional networks (R-FCN) [7], Mask R-CNN [8] have been proposed successively. However, most of these methods are based on the three-step strategy of candidate region selection, feature extraction, and classification.

After that, some researchers proposed the regression-based object detection method, which was represented by YOLO [9, 10], SSD [11], etc. These methods only use one ConvNet for detection and use the regression method to correct the object location, making the detection speed much faster than the candidate region-based detection methods such as Faster R-CNN. However, their disadvantages are large localization error, which is mainly because it is very difficult to get an accurate location by regression directly in the absence of proposal regions. In the training, there will be a wide range jitter of the bounding box and the losses function is hard to converge. In addition, the detection performance often becomes poor when multiple adjacent small objects appear. The rest of this paper, I will focus on the key techniques used in object detection using ConvNet.

2 Convolutional Implementation of Sliding Windows

In this section, we will introduce to build a convolutional neural network to implementation of sliding windows. The traditional sliding window method is very slow due to a lot of repeated convolution computation. In this part, let us see how we can improve the processing speed by using the convolutional implementation of sliding windows.

Firstly, let us see how we can transform fully connected layers into convolutional layers. For illustrative purposes, let us see a simple network as shown in Fig. 1a. Our object detection algorithm inputs 14 × 14 × 3 images, then 16 of 3 × 3 filters are used to map the inputs images from 14 × 14 × 3 to 10 × 10 × 16, then does a 2 × 2 max pooling to reduce it to 4 × 4 × 16, then has a fully connected layer to connect 512 units, and then finally, outputs y using a softmax layer. Here, we assume that there are four categories, which are pedestrian, car, traffic light, and background. Then y have four units, corresponding to the crossed probabilities of the four classes that the softmax unit is classifying among. And the four classes could be pedestrian, car, traffic lights, or background.



(a) Object classification with fully connected layers

(b) Object classification without fully connected layers

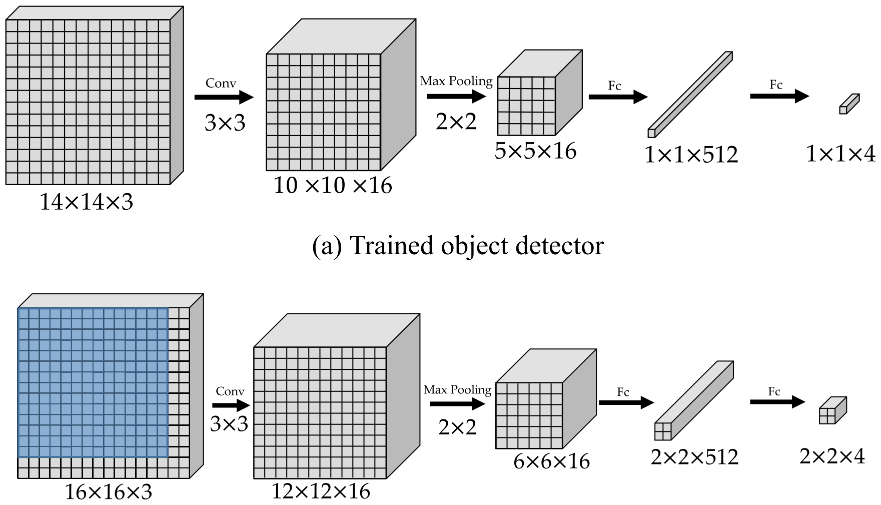
Fig. 1. Convolutional implementation for fully connected layers

Now, let us show you how these fully connected layers can be turned into convolutional layers. As shown in Fig. 1b, the first few layers of the ConvNet have the same structure. After that, to implement the convolutional layer, we use 512 of 5 × 5 × 16 filters to do the convolution, and the output dimension is going to be 1 × 1 × 512. In mathematically, this is the same as a fully connected layer, because each of these 512 nodes has a filter of dimension 5 × 5 × 16, and so each of those 512 values is some arbitrary linear function of these 5 × 5 × 16 activations from the previous layer.

Finally, we are going to use four of 1 × 1 × 512 filters by a softmax activation to get a 1 × 1 × 4 volume as the output of this network. So, this shows how you can take these fully connected layers and implement those using convolutional layers, and these fully connected layers are now implemented as 1 × 1 × 512 and 1 × 1 × 4 volumes.

When the size of the test image we inputted changed to 16 × 16 × 3, assume our trained detection window is still 14 × 14 (shown in Fig. 2a). So in the original sliding windows algorithm, you might want to input the first 14 × 14 regions into a ConvNet and run that once to generate a classification 0 or 1. Then slide the window to the right by a stride = 2 pixels to get the second rectangular area, and run the whole ConvNet for this window to get another label 0 or 1. Then repeat this process by slide the window until get the output of lower right window. You will find for this small input image, we run this ConvNet from above four times in order to get four labels. But we can see that many of the operations by these four ConvNet are highly duplicated. So what the convolutional implementation of sliding windows does is it allows these four forward passes of the ConvNet to share a lot of computation.

As shown in Fig. 2b, you can take the ConvNet and just run it by the same 5 × 5 × 16 filters with same parameters, and you can get a 12 × 12 × 16 output volume, and then do the max pool same as before, get a 6 × 6 × 16 output, run



(a) Trained object detector

(b) Object detection

Fig. 2. Object detection by convolutional implementation

through your same 400 of 5 × 5 filters to get a 2 × 2 × 400 volume output. So, now instead of a 1 × 1 × 400 volume, you can get a 2 × 2 × 400 volume. Do that one more time, now you are left with a 2 × 2 × 4 output volume instead of 1 × 1 × 4. So, in this final output layer, which turns out the upper left 1 × 1 × 4 subset gives you the result of running in the upper left corner 14 × 14 regions of input image, the upper right 1 × 1 × 4 volume gives you the upper right result and same argument for the lower left and right 1 × 1 × 4 volume. If you step through all the steps of the calculation, you will find that the value of each cell is the same as if you cut out each region and input it to ConvNet.

So, what this convolutional implementation does is you need not run forward propagation on four subsets of the input image independently. We only need to do forward propagation once for the entire image and share a lot of the computation in the regions of the image that is common. You can refer to [12] to get more details about this part.

3 Transfer Learning

When your network is ready, prepare training data for learning. The process of optimizing the neural network is to find the value of each layer of parameters that minimize the output of the loss function. But if you do a little bit of calculation, you will find that there are huge parameters in a neural network that needs to be trained, i. e, a medium-sized vgg-16 network [13] has at least 1.5 × 10^7 parameters.

If you have a new network that needs to be trained from scratch, then you have to prepare a lot of training samples, such as a database like ImageNet [14], MS COCO

[15], etc. And, you probably need a high-performance GPU computing cluster, and sometimes, this training takes several weeks or even months to make the output of the loss function converge.

So, how can we train neural networks to develop our own image processing applications without such a GPU computing cluster or lots of training samples? One solution is transfer learning. Transfer learning is to transfer the model parameters that have been learned from other places to the new model to help train the new model.

Due to the explosive growth of research related to neural network in recent years, people often associate transfer learning with training of neural network. These two concepts are unrelated at the beginning. Transfer learning is a branch of machine learning, and many transfer methods did not need to use neural network. But now, people find that deep neural network model has strong transferability. Because DNN is a hierarchical representation of data obtained through pretrain and then classified with high-level semantic classification. The low-level semantic features (such as texture, edge, color information, etc.) are at the bottom of the model. Such features are actually invariable in different classification tasks, and the real difference is the high-level features.

If you apply the same network architecture to implement a vision task, you can usually download weights that someone else has trained on the same network as initialization, rather than training it from random values. And use transfer learning to sort of transfer knowledge from some of these very large public data sets to our own problem. Usually, this transfer process can greatly reduce the convergence time of the model.

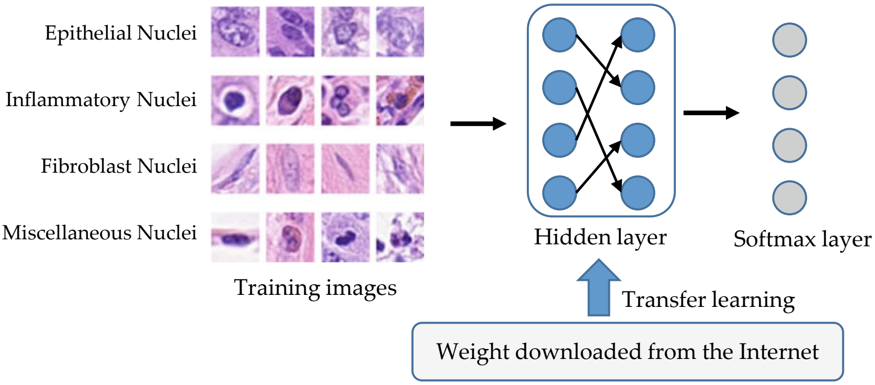


Fig. 3. A sample of transfer learning

Let us see an example of medical image recognition, as shown in Fig. 3. If we need building a cell detector to recognize colon cancer histology images, we were prepared to recognize four clinically meaningful cells: epithelial nuclei, inflammatory nuclei, fibroblasts, and miscellaneous nuclei. And we assume that each cropped image patch contains only one cell. So, we have a classification problem with four classes. But our training set is small. Then we would better download some open-source implementation of a neural network and download not just the code, but also the weights of this network. There are a lot of networks; you can download that have been trained on, i.e., the ImageNet dataset, which has 1000 different classes. The network typically contains a softmax layer used as the output layer, which contains 1000 units. You usually need to modify the softmax layer and create your own softmax unit that outputs epithelial, inflammatory, fibroblasts, or miscellaneous.

Since our training dataset is small, we would better freeze the first few layers of the neural network and then just train the parameters associated with finial several layers. For most machine vision tasks, the first few layers of the network are usually used to extract common fundamental features of input images, such as edges, gradients, etc. And by using the pretrained weights, we can get very good performance even with a small data sets. Many current deep learning frameworks support using parameters to specify training or freezing the weights associated with a particular layer. This method is usually used to solve the problem of overfitting caused by insufficient training data. But if you have enough data, you could use the downloaded weights just as initialization to replace random initialization. And then you can do gradient descent training, updating all the weights in all the layers to train the whole network.

This is the transfer learning for the training of ConvNet. In practice, because many network weights are publicly available on the internet, you can easily download these open source weights that someone else has spent weeks working with a large computing device to initialize your network model. This will greatly improve your work efficiency. You can refer to [16] for more details.

4 Generate Training Dataset

When your network is built, you need to provide the image it will use to train a new detection classifier. Most deep neural network models require at least hundreds of images to train a detection classifier. Many computer vision researchers are willing to share their data online, i.e., ImageNet. And other researchers can also train their algorithms on these databases

To train a robust classifier, the training images should have random objects in the image along with the desired objects and should have a variety of backgrounds and lighting conditions. There should be some images where the desired object is partially obscured, overlapped with something else, or only halfway in the picture. Make sure the images are not too large. The larger the images are, the longer it will take to train the classifier. As my advice, the images size should be less than 200 KB each, and their resolution should not be more than 1280 × 720.

In practice, after you have all the pictures you need, we usually use 80% of them as the training images and 20% of them as testing images. Make sure there are a variety of pictures in both directories of the training and testing data.

With all the pictures gathered, it is time to label the desired objects in every picture. LabelImg [17] is a tool for labeling images, and its GitHub page has very clear instructions on how to install and use it. Once you have labeled each image, there will be generated one *.xml file for each image in the images directory. These will be used to train the new object detection classifier.

5 Conclusion

Object detection has been developing rapidly in the field of computer vision and has repeatedly created amazing achievements. The ConvNet has a strong versatility and portability, and it is widely used in robot, autonomous driving, medical assistance, etc. This chapter focuses on several core problems in object detection: object localization, sliding windows object detection, and transfer learning. In addition, we also introduced how to classify and locate cells in medical image by ConvNet.

In summary, the success of ConvNet in recent years mainly depends on three pillars: data, model, and calculation power. A large amount of manually annotated data makes it possible to conduct supervised training. A deeper and larger model improves the recognition ability of the neural network. The combination with GPU and the rapid development of computer hardware makes large-scale training become time-saving and effective. However, the research on ConvNet is just beginning, and many aspects need further study.

At present, ConvNet needs training samples of tens of thousands or even millions of levels, and the training process for such a large number of samples is also extremely long. And in many areas, it is very expensive to obtain large numbers of precisely labeled samples. How to generate a good neural network from a small amount of data will be the future research direction. How to get satisfactory performance by train a small amount of dataset will be the future research direction.

In addition, the trend is that the deeper the network, the better the performance of ConvNet, and some networks even reach thousands of layers. However, as the network deepens, overfitting and gradient disappeared become more serious. Although the research such as residual network [18], etc., is devoted to solving such problems, the large model also limits the application of ConvNet on common devices, especially mobile devices. The ConvNet needs to optimize the structural design to find more efficient neurons and structural units.

References

1. Wang X, Han TX, Yan S (2009) An HOG-LBP human detector with partial occlusion handling. In: 2009 IEEE 12th international conference on computer vision, Sept 2009, pp 32–39
2. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition, pp 580–587
3. Everingham M, Gool L, Williams CKI, Winn J, Zisserman A (2009) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
4. He K, Zhang X, Ren S, Sun J (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans Pattern Anal Mach Intell 346–361

5. Girshick R (2015) Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
6. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 6:1137–1149
7. Dai J, Li Y, He K, Sun J (2016) R-FCN: object detection via region-based fully convolutional networks. Adv Neural Inf Process Syst 379–387
8. He K, Gkioxari G, Dollar P, Girshick R (2017) Mask R-CNN. In: Proceedings of the IEEE international conference on computer vision, pp 2980–2988
9. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 779–788
10. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: IEEE conference on computer vision and pattern recognition, CVPR 2017, pp 7263–7271
11. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) SSD: single shot multibox detector. In: European conference on computer vision, pp 21–37
12. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y (2013) OverFeat: integrated recognition, localization and detection using convolutional networks. arXiv Prepr. arXiv
13. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition
14. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) ImageNet: a large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition
15. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft COCO: common objects in context. In: Lecture Notes in Computer science (including subseries Lecture Notes in Artificial intelligence and lecture notes in bioinformatics)
16. Yosinski J, Clune J, Bengio Y, Lipson H (2014) How transferable are features in deep neural networks? Adv Neural Inf Process Syst 3320–3328
17. Tzutalin (2015) LabelImg. https://github.com/tzutalin/labelImg
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778

Research on Prediction Model of Gas Emission Based on Lasso Penalty Regression Algorithm

Qian Chen[1](✉) and Lianbing Huang[2]

[1] China Coal Research Institute, Beijing 100013, China
365039211@qq.com
[2] Institute of Manned Space System Engineering, Beijing 100094, China

Abstract. Researches show that the amount of mine gas emission is influenced by many factors, including the buried depth of coal seams, coal thickness, gas content, CH_4 concentration, daily output, coal seam distance, permeability, volatile yield, air volume, etc. Its high-dimensional characteristics could easily lead to *dimension disaster*. In order to eliminate the collinearity of attributes and avoid the over-fitting of functions, Lasso algorithm is used to reduce the dimension of variables. After low-redundancy feature subset is obtained, the best performance model is selected by 10-fold cross-validation method. Finally, the gas emission is predicted and analyzed based on public data from coal mine. The results show that the prediction model based on Lasso has higher accuracy and better generalization performance than principal component analysis prediction model,and the accurate prediction of gas emission can be realized more effectively.

Keywords: Gas · The amount of emission · Feature selection · Penalty regression · Prediction model

1 Introduction

In recent years, with the number of safety accidents and death toll have decreased year by year, China's coal mine safety situation has improved, but the security situation is still grim, since coal mine accidents occur frequently and major accidents happen from time to time. According to incomplete statistics, there are 1945 coal mine accidents from 2013 to 2017, and the death toll was 3771. Among them, the number of gas accidents and deaths accounted for 11.21% and 30.17%, respectively, becoming the second largest safety accident after the roof disaster accident. Therefore, gas is still one of the main factors causing coal mine safety accidents.

Many scholars have done a large number of researches on the prediction model of gas emission for coal mine. Qi and Xia [1] used the different-source prediction method to predict the mine gas emission. Li [2] used the grey system theory to study the mine gas emission. Using the l_1 regularized outlier isolation and regression method (LOIRE) with the TLBO optimization algorithm, Hu et al. [3] established the TLBO-LOIRE optimization prediction model to calculate and analyze the relevant influencing factors and predicted the gas emission from the coal mining face.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 165–172, 2020.
https://doi.org/10.1007/978-981-15-0187-6_19

The gas emission from the mining face is complicated and affected by many factors including gas content, gas permeability, pressure, and buried depth. The high-dimensional characteristics of these influencing factors are likely to lead to *dimensional disaster* and *over-exposure*, which will influence the accuracy of prediction. Thus, dimension, accuracy, and representativeness of the factors affecting the amount of gas emission are crucial to the prediction. In this article, a feature selection method based on Lasso algorithm is proposed. On the basis of the original feature space, an optimal feature subset is selected by eliminating irrelevant and redundant features with better readability; meantime, the feature meaning of the original dataset has not changed. The main features of the influence factors of gas emission screened from data perspective are used to establish the prediction model, which can solve the multi-collinearity problem between variables so that the changing law of gas emission from the mining face can be accurately tracked.

2 Principle of Lasso Algorithm

The least absolute shrinkage and selection operator (Lasso) is a regularized sparse model with penalty, first proposed by statistician Tibshirani in 1996. In order to provide effective algorithm support for Lasso, Efron et al. [4] proposed the least angle regression (LARS) algorithm. Friedman proposed algorithm which can solve calculation problem of Lasso as well, but glmnet ignores the difference between the two adjacent regression coefficients, making the volatility of the estimated value large. Zou and Trevor [5] proposed the elastic net method, which adds a two-norm constraint on the basis of LARS to solve the over-fitting problem of high-dimensional small sample.

The basic idea of Lasso regression is to perform L1 norm constraint on regression coefficients, so that the regression coefficients of some independent variables can be automatically compressed to zero since the residual sum of square is minimized. In other words, based on the least square estimation of the traditional linear regression method, it adds the penalty term of the absolute value situation to select variables and obtain an interpretable model.

For multiple linear regression models:

$$y = \alpha + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \varepsilon, \tag{1}$$

The Lasso estimates for the constant term and the regression coefficient are as follows:

$$(\hat{\alpha}, \hat{\beta}) = \arg\min \sum_{i=1}^{n} \left(y_i - \alpha_i - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \quad \text{s.t.} \sum_{j=1}^{p} |\beta_j| \le t, (\mathrm{t} \ge 0) \tag{2}$$

Lasso regression uses the correlation among data to compress the regression coefficient β_i which has little influence on y by controlling the parameter t. The smaller the t is, the higher the degree of compression, so that more regression coefficients will be compressed to 0 and the corresponding variables will be removed from the model to achieve the purpose of selecting variables. $\hat{\beta}_j^0$ represents least squares estimation of β_j and compression will happen once $t < \sum_{j=1}^{p} \left| \hat{\beta}_j^0 \right|$, with some coefficients changing toward 0.

3 Prediction Model Based on Lasso

In order to build the model, the original data is normalized at first, and then, the collinearity between properties is judged. If collinearity exists, Lasso is used for dimensionality reduction. Otherwise, multiple regression analysis is performed. Whole process is shown in Fig. 1.

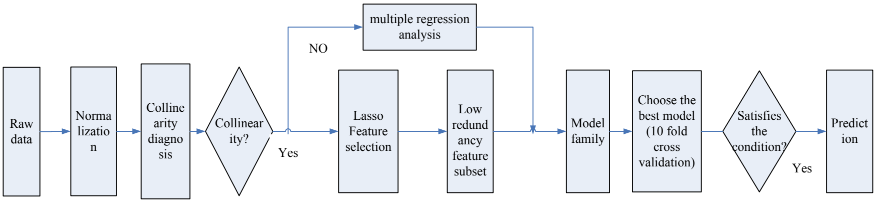


Fig. 1. Modeling process of Lasso penalty regression algorithm

1. Normalization of data.

When using Lasso regression, in order to eliminate the influence of dimensions of different indexes, it is necessary to standardize and normalize the observation data.

2. Feature selection by Lasso.

After calculation through Lasso, when the parameter regression result corresponding to influencing factor is “0”, the feature is discarded. Otherwise, the feature is included in the candidate feature set. Thereby, a sparse solution is obtained for dimensionality reduction.

3. Generation of model family.

After obtaining a subset of low-redundancy features, the model family is created by using feature subsets to obtain regression coefficients corresponding to different alpha values.

4. Model optimization.

This article uses the 10-fold tutorial verification method to select the model, and specific steps are as follows: ① Disturb the order of the training set randomly. ② Divide the disturbed training set into 10 parts. ③ Start with the first parameter in the parameter set and select one parameter each time without repeating. ④ From the first copy, take one copy each time as a test set, and the rest as a training set. ⑤ Use training sets and selected parameters for model training. ⑥ Forecast test sets with trained models. ⑦ Calculate the average recognition rate for all prediction results. ⑧ Select the test set parameter that gets the highest recognition rate. Using cross-validation methods, good learning parameters can be verified based on the characteristics of the data itself.

4 Application and Analysis of Model

4.1 Selection of Factors Affecting Gas Emission in Working Face

There are many influencing factors affecting the mine gas emission. In this paper, nine factors measured within 48 months in the mining face of a coal mine in Huaibei Province are adopted as original variables, including coal seam depth (X_1), gas permeability (X_2), coal seam thickness (X_3), gas content (X_4), CH_4 concentration (X_5), air volume (X_6), daily production of coal (X_7), coal seam distance (X_8), and volatile yield (X_9). Table 1 shows 62 sets of data including the amount of gas emission (Y) and its influence factors in working face. The first 58 sets of data are used as training sets and the last 4 sets are used to test the prediction effect.

Table 1. Datasets of gas emission quantity and influencing factors

No.	X_1 (m)	X_2 (m/d)	X_3 (m)	X_4 (m^3/t)	X_5 (%)	X_6 (m^3/min)	X_7 (t)	X_8 (m)	X_9 (%)	Y (m^3/t)
1	228	4.1	1.9	4.99	0.22	589	786	10	32	1.38
2	229	6.1	2.5	6.01	0.2	360	786	9	30	1.32
3	230	4.9	1.5	5.27	0.2	405	421	9	30	2.77
4	235	7.2	1.2	4.1	0.2	417.1	1436	10	31	1.84
5	236	6.1	1.1	2.96	0.2	447.2	1517	8	38	0.85
6	237	7.6	3.3	3.7	0.2	477	1644	10	38	1.81
7	238	7.91	1.9	2.01	0.2	480	1616	10	38	0.86
8	239	8.11	1.8	3.11	0.2	462	1694	10	38	0.79
9	240	7.12	1.9	2.14	0.2	462	1661	7	39	1.52
10	241	9.1	1.9	3.25	0.2	480	1661	8	38	1.83
11	242	6.99	2	2.91	0.2	540	1553	7	38	1
12	243	6.72	1.1	2.16	0.2	720	1570	7	39	1.32
13	244	7.81	1.2	3.19	0.2	660	1500	8	37	1.27
14	245	7.37	1.6	4.1	0.24	648	1619	8	34	1.38
15	246	8.26	1.6	2.31	0.25	598.5	1581	9	38	1.36
16	247	8.46	1	2.46	0.2	452.4	1574	6	37	0.83
…	…	…	…	…	…	…	…	…	…	…

4.2 Correlation Analysis

This article uses a visual approach named parallel graphs and associated heat maps to show the relationship between factors and target (gas emission).

After normalizing the data in Table 1, the correlation visualization is displayed in Figs. 2 and 3. In parallel graph, where the properties are similar, the colors of the polylines will be close and they will be centralized, indicating that a fairly accurate model can be built. In the associated heat map, red represents strong correlation and blue represents weak correlation. It can be seen from the heat map and parallel graph that there is a strong correlation between the attributes, which is suitable for the Lasso algorithm for feature screening.

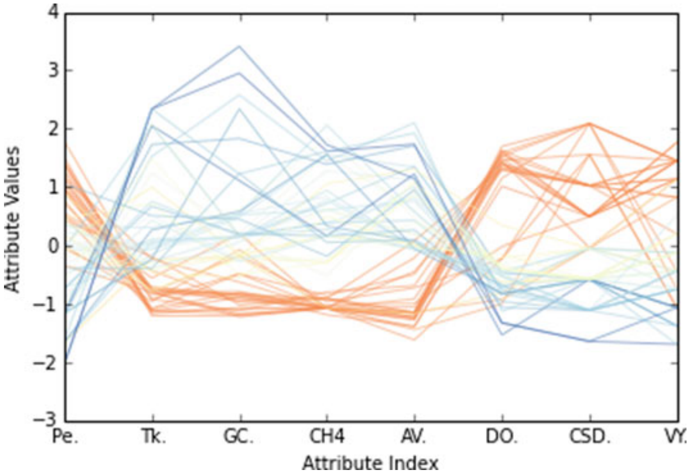


Fig. 2. Parallel graph of data

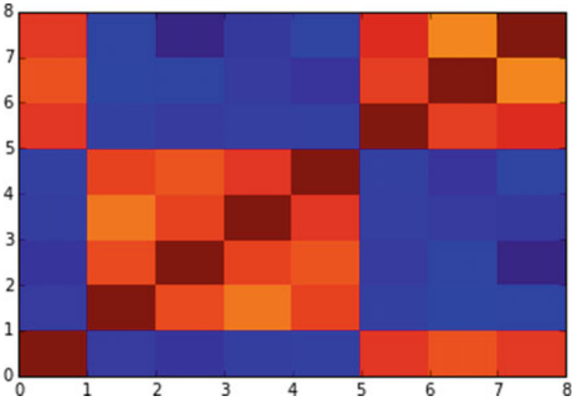


Fig. 3. Correlation heat map of data

4.3 High-Impact Factors Screening by Lasso

Using Python language programming for machine learning, the Lasso regression fitting results under different alpha values were obtained. Mean square error under different values was calculated by cross-validation, and the prediction accuracy of each model was compared, as shown in Fig. 4.

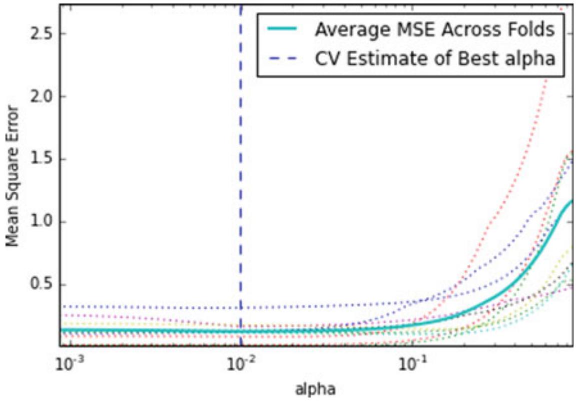


Fig. 4. Figure of alpha and mean square error

Figure 4 shows the trend of the model prediction mean square error (MSE) for the parameter alpha value. Apparently, with the increase in the alpha value, the screening effect of the model becomes more obvious, and the number of high-impact factors selected by the model is less, but the prediction mean square error also increases. Therefore, appropriate parameter values should be chosen to make a trade-off. It can be known from observation that when the value of alpha is around 10^{-2}, six high-impact factors were extracted from the first nine influence factors, and three low-impact factors were eliminated, which simplified the index system to some extent. Meantime, mean square error of the model is controlled below 0.5, which can achieve both the prediction accuracy and the factor screening.

At the same time, the variation of the gas emission in the Lasso regression with the parameter is shown in Fig. 5. It can be seen that as the alpha value increases gradually, the degree of compression increases, and the number of variables selected into the model decreases. The dashed line is the position of the alpha.lse value, which corresponds to a more concise model within one standard error. Therefore, this article selects alpha.lse for variable screening.

It can be seen from Table 2 that after Lasso regression is realized by LARS or glmnet, at best alpha, three attributes are excluded, namely gas permeability, volatile yield, and air volume, and the sparse vector solution is obtained.

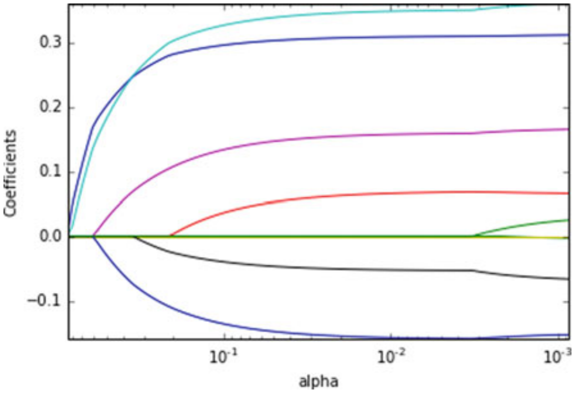


Fig. 5. Regression coefficient with alpha

Table 2. Screen results of each factor by the Lasso algorithm

Algorithm	X_4	X_1	X_8	X_5	X_3	X_7	X_2	X_9	X_6
LARS	0.34795	0.30816	0.15724	0.15676	0.06543	0.05217	0	0	0
Glmnet	−0.17406	0.31524	0.04347	0.15781	−0.31221	0.08357	0	0	0

4.4 Comparison of Prediction Results

In order to better verify the prediction effect, the results are compared with principal component analysis prediction, as shown in Table 3. The prediction of 59–62 samples showed that the Lasso average prediction error was 4.8%, and the principal component regression prediction average error was 14.4%. The prediction accuracy based on the Lasso multiple regression model was higher, which also indicated that the Lasso feature screening method was superior to principal component analysis method.

Table 3. Prediction of gas emission based on Lasso and principal component regression

No.	Measured value	Prediction by Lasso	Prediction by principal component regression
59	33.05	31.61	28.76
60	37.06	36.18	27.91
61	60.82	55.33	53.47
62	56.41	58.34	51.82

Meanwhile, the average relative variance (ARV) and root mean square error (RMSE) are introduced as the error evaluation criteria to comprehensively evaluate the prediction accuracy and generalization ability of the model. The smaller the ARV value, the stronger the generalization ability of the prediction model. The smaller the RMSE value, the higher the accuracy of the prediction model. A comprehensive comparison of the effects of two different prediction models is shown in Table 4.

Table 4. Comparison of two prediction models

Index	Lasso penalty regression algorithm	Principal component analysis
ARV	0.006	0.012
RMSE	3.40	4.89

Obviously, prediction results from Lasso penalty regression algorithm have lower ARV and RMSE compared with principal component regression model, indicating its stronger generalization ability and higher accuracy.

5 Conclusion

The research shows that the amount of mine gas emission is influenced by many factors. Its high-dimensional characteristics easily lead to dimension disaster. In order to eliminate the collinearity of attributes and avoid the over-fitting of functions, Lasso algorithm is used to reduce the dimension of variables. Finally, the gas emission is predicted and analyzed on the public data. The results show that the prediction model based on Lasso has higher accuracy and better generalization performance than the principal component multiple regression algorithm, and has a wide application prospect in machine learning.

Acknowledgements. This article is sponsored by National Science and Technology Major Project of China (2016ZX05045-007-001).

References

1. Qi QJ, Xia SY (2018) Construction of gas emission forecasting sharing platform based on sub-source prediction method. J Min Saf Environ Prot 45(02):59–64
2. Li ZY (2018) Analysis of distribution law of coal mine gas emission and prediction of GM (1,1) prediction. J Coal Mine Mod 2018(02):39–41
3. Hu K, Wang SZ, Han S, Wang S (2017) Prediction of gas emission in mining face based on TLBO-LOIRE. J Appl Basic Eng Sci 25(05):1048–1056
4. Efron B, Hastie T, Johnstone I (2004) Least angle regression. J Math Stat 32(2):407–499
5. Zou H, Trevor H (2005) Regularization and variable selection via the elastic net. J R Stat Soc 67(2):301–320

Water Meter Reading Area Detection Based on Convolutional Neural Network

Jianlan Wu(✉), Yangan Zhang, and Xueguang Yuan

Beijing University of Posts and Telecommunications, No. 10, Xitucheng Road,
Haidian District, Beijing 100876, China
wujianlan@bupt.edu.cn

Abstract. The water meter is a device for measuring the amount of water used by each household. Remote meter reading is one of the main ways to solve the waste of human resources caused by regular manual door-to-door access to mechanical water meter readings. The current use of image acquisition and then accurate reading of the water meter image is one of the ways of remote meter reading. In this paper, the convolutional neural network is used to predict the reading area, and then the non-maximum suppression algorithm (NMS) is used to remove highly overlapping results from prediction region results to obtain the position of the reading area. The experimental results show that with using the method proposed in this paper in the actual application scenario, the IoU of the images of 1000 test sets are all above 0.8 and then combined with the three-layer BP neural network for character recognition, the accuracy rate reaches 98.0%.

Keywords: Convolution neural network · Image detection · Water meter number reading

1 Introduction

At present, most water meters are still mechanical water meters, and the update speed of smart water meters is slow. For users who have not replaced smart water meters, the work of water meter readings is mainly done by humans every month, which is not only inefficient, but also has potential safety hazards. A lot of human resources are wasted, so remote meter reading is required. One of the remote meter reading methods uses image acquisition and then recognizes the number on the picture.

Image-based remote meter reading is mainly divided into two steps. The first step is the detection of the reading area and the second step is the recognition of characters. Among them, the detection of the reading area is the primary premise of character recognition. Common methods for detecting the reading area are: method based on Canny edge detection [1–3] and Hough transform [4–9], method based on projection [10–12].

The method based on Canny edge detection and Hough transform, the Canny edge detection algorithm is used to obtain the edge of the image. The Hough circle detection removes the pixel interference between the circular area and the area where the circular metal casing is located and then finds the water meter character by Hough line detection. The line of the four sides of the number box is located, and finally, the

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 173–179, 2020.
https://doi.org/10.1007/978-981-15-0187-6_20

positioning is based on the number and geometric features of the number box. However, such a detection method has obvious disadvantages. It is easily affected by bubbles on the dial of the water meter, the angle of inclination of the image, blurring, etc., and the detection accuracy is low.

The method based on projection is a method of combining the gray values of each pixel of the obtained binarized image in a certain cross-sectional direction. The projection method is to count the number of black pixels in the horizontal and vertical directions displayed in the image reading area, and draw a corresponding projection image according to the counted number of pixels, and then locate the position where the number on the dial is located. Specifically, the horizontal projection is to judge the pixel value of each row of the obtained binarized image, and the number of gray values of all the black pixels is counted; the vertical projection is to judge the pixel value of each column of the obtained binarized image and count the number of gray values of all the black pixels of the column. However, such detection methods are also susceptible to many factors, such as light spots, stains, partial occlusions, etc., on the dial of the water meter, and the accuracy is also low.

In this paper, the convolutional neural network is used to predict the reading area, and then the non-maximum suppression algorithm (NMS) is used to remove highly overlapping results from prediction region results to obtain the position of the reading area. The IoU of the images in the 1000 test sets are all above 0.8 and combined with the three-layer BP neural network for character recognition, the accuracy rate is 98.0%. Compared with other methods, the reading results of the method in this paper are not easily interfered by external factors such as bubbles, spots, blurs, etc., and the sensitivity is low.

The paper is organized as follows. Section 1 is talked about introduction to this research. Section 2 introduces the specific implementation method of the predicted reading area. Section 3 contains the experimental results and analysis. Section 4 summarizes the paper.

2 System Implementation

The network structure of the method is shown in Fig. 1. The implementation mainly includes the following parts. Firstly, the convolutional neural network is used to predict the reading area. Secondly, the non-maximum suppression algorithm (NMS) is used to remove highly overlapping results from prediction region results. Finally, the accuracy is calculated by the result of character recognition and thus compared with other algorithms.

2.1 Predict the Reading Area with Convolutional Neural Network

To perform character recognition on the readings in the water meter image, the first thing to do is to detect the correct reading area and eliminate character interference in the non-reading area, which is critical for character recognition of the readings.

The specific structure of the convolutional neural network is shown in Fig. 1. It consists of several convolutional layers and several pooling layers, including five two-

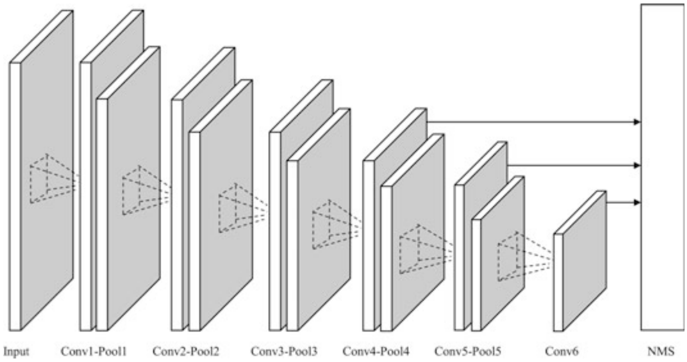


Fig. 1. Network structure for detection of reading area

dimensional convolutional and pooling layers and a separate convolutional layer, the first volume. The first convolution kernel is 5 × 5, the padding is set to the same mode, and the step size is 1. The convolution kernel of the remaining convolutional layers is 3 × 3, the padding is set to the same mode, and the step size is 1. The pooling layer is of the pool size 2 × 2, and the step size is 1. In the training process of the network, the Adam algorithm [13] is used instead of the traditional random gradient descent algorithm to perform a gradient optimization update.

The input of the whole network is the image data of the water meter, which is regarded as two-dimensional data. If the image is a RGB image of three color channels, we use three convolution kernels. If the image is a grayscale image, the gray value is copied to the corresponding three color channels. The overall loss function is composed of a position loss function and a confidence loss function. The specific calculation formula is as follows:

$$L(x, c, l, g) = 1/N(L_{\mathrm{conf}}(x, c) + \alpha L_{\mathrm{loc}}(x, l, g)) \tag{1}$$

where L is the overall loss function; x is the match between the i-th prediction box and the j-th real box, and the value range is $\{0, 1\}$; c is the confidence, l is the prediction box, and g is the real box. L_{conf} is a confidence loss function and L_{loc} is a position loss function; N is the number of matching default boxes and α is the weight of L_{conf} and L_{loc}, which is set to 1.

2.2 Remove Highly Overlapping Results with NMS

As shown in Fig. 1, we add the output of the fourth, fifth, and sixth convolutional layer to the end of the network and perform non-maximum suppression (NMS) [14–16] to remove the high overlap predicted boxes to obtain the optimal solution. The purpose of the NMS algorithm is to eliminate redundant boxes and find the best object location. The core idea of NMS to remove high-overlapping predicted boxes is based on the premise that there is no overlap or low overlap between the hypothetical instances.

Firstly, we sort the predicted boxes according to the confidence level and select the predicted box with the highest confidence. Secondly, set an overlap threshold and traverse other adjacent predicted boxes. When the overlap between the adjacent predicted box and the highest confidence predicted box is higher than the threshold we set, the adjacent predicted box is removed, and when it is lower than the threshold we set, the adjacent predicted box is retained. Then repeat the above steps by selecting one of the most confident predicted boxes in the unprocessed predicted boxes.

3 Experiment Result

The water meter reading area detection method in this paper is mainly evaluated from two aspects. One is the overlap between the predicted area and the real area, and the other is the accuracy of the reading recognition combined with the three-layer BP neural network. And in terms of accuracy, the method of this paper is compared with other methods. In addition, because there is currently no authoritative dataset for water meter readings, this paper is based on our own dataset for evaluation.

3.1 Dataset

The acquisition and labeling of image datasets is the basis of neural network training. The size of datasets and the quality of data annotation directly affect the training effects and prediction results of neural networks. The images required for training and testing the model are randomly divided into training set, verification set, and test set. The ratio of the three data sets can be in the range of 4:1:1–8:1:1. We use the camera to directly shoot the water meter dial, and adjust the position of the camera and the reading of the water meter, and mark the acquired picture, including the coordinate values of the two endpoints of the diagonal of the reading area and the reading of the water meter value.

For the size of the water meter image, the neural network of this paper does not have strict requirements for this and can use different sizes of pictures. For the color channel of the water meter picture, RGB three color channels and grayscale images can be used. For the tilt problem of the image, there is no need to tilt the image. We only need to tilt a part of the image at an appropriate angle. This part of the image can be randomly placed into the training set, verification set, and test set, which can increase robustness of the model. In addition, in order to better simulate the real application scenario, we can also grayscale and blur the image, add noise, light spots to the image, which can enhance the generalization of the model.

This paper collected 6000 water meters, randomly divided the pictures into three parts according to the ratio of 4:1:1, 4000 pictures as the training set, 1000 pictures as the verification set, and 1000 pictures as the test set.

3.2 The Overlap Between Predicted Area and Real Area

The degree of overlap between the predicted area and the real area (IoU) is a criterion for measuring the accuracy of detecting the corresponding object in a specific data set. The calculation method is the area where the two regions overlap and the area where the two regions are combined. The value of IoU is between 0 and 1. The larger the value, the better the prediction effect and the better the work for subsequent character recognition.

The test result of the reading area IoU is that the IoU of all pictures is above 0.8, and the specific distribution is shown in Table 1. As it can be seen from the test results in Fig. 2, the confidence of this predicted box is 0.99, and the predicted results are completely in line with expectations.

Table 1. Distribution of IoU

IoU interval	Number of images	Proportion (%)
IoU $\geq$ 0.9	554	55.4
0.9 > IoU $\geq$ 0.8	446	44.6

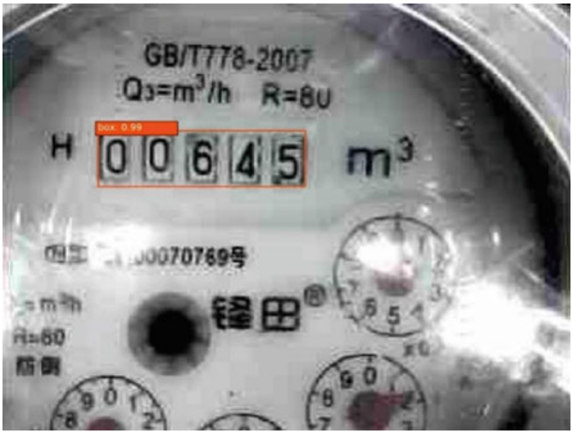


Fig. 2. Results of the predicted reading area

3.3 Recognition Accuracy with Three-Layer BP Neural Network

Accuracy is the standard used to measure character recognition. The calculation method is the number of pictures that are correctly recognized by all five characters divided by the number of all pictures. The accuracy rate is between 0 and 1. The higher the value, the more accurate the recognition result.

The test result of character recognition is that 980 images of all five characters are correctly recognized, and the accuracy rate is 98.0%, which can be applied to reality. Figure 3 shows the test results in the actual scenario. The predicted result of the reading is 00645, and the result is in line with expectations.

In order to better compare the results with other methods, we have tested various methods on the dataset of this paper. Because different methods have different ideas, not all methods can be evaluated with IoU. However, all methods evaluate the accuracy of the readings, so the accuracy of the readings is used here for evaluation. The test

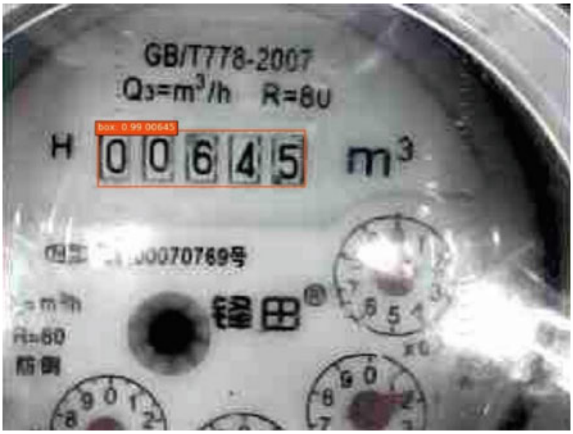


Fig. 3. Predicted results of readings

Table 2. Distribution of IoU

Method	Accuracy rate (%)
Method based on projection and BP neural network	90.1
Method based on Canny edge detection, Hough transform, and template matching	93.3
Proposed method with BP neural network	98.0

results are shown in Table 2. It can be seen that the accuracy rate based on projection method and BP neural network method is 90.1%, and the accuracy rate based on Canny edge detection, Hough transform, and template matching method is 93.3%, and the accuracy of the proposed method is 98.0%. There are obvious advantages in the proposed method.

4 Conclusion

This paper proposes a method based on convolutional neural network for water meter reading area detection, which lays a foundation for the recognition of readings. In this paper, the convolutional neural network is used to predict the reading area, and then the NMS algorithm is used to remove highly overlapping results from prediction region results to obtain the position of the reading area. The experimental results show that the IoU of the images of 1000 test sets are all above 0.8 and combined with the three-layer BP neural network for character recognition, the accuracy rate reaches 98.0%. The method of this paper achieves the detection of water meter readings efficiently and accurately. The method is not only unsusceptible to external environmental factors, but also has generalization ability, and the model structure is simple, requiring less computing resources.

References

1. Sun K, Guo X, Zhang W, Wang X (2018) Edge detection based on improved canny algorithm. Comput Program Ski Maint 6:165–168+173
2. Duan S, Yin C, Li D (2018) Improved adaptive Canny edge detection algorithm. Comput Eng Des 39(06):1645–1652
3. Liu W, Zhao X, Gao D (2018) Research on Canny edge detection algorithm based on multiscale. Beijing Surv Mapp 32(1):127–131
4. Qu H, Wang Y, Wu C (2017) Hough transform detection circle algorithm based on diameter accumulation. J Xi'an Univ Posts Telecommun 22(05):89–93
5. Meng YC, Zhang ZP, Yin HQ, Ma T (2018) Automatic detection of particle size distribution by image analysis based on local adaptive canny edge detection and modified circular Hough transform. Micron 106
6. Wang T (2017) Research on geometric shape detection algorithm based on improved Hough transform. Nanjing University of Aeronautics and Astronautics
7. Tang Y, Li D, Zhu S, Huang X (2017) Research on lane recognition algorithm based on canny and Hough transform. Automob Pract Technol 22:81–83
8. Chen Y, Yang Y (2017) Two improved algorithms based on hough transform of hough transform. Semicond Optoelectron 38(5):745–750
9. Zheng F, Luo S, Song K, Yan CW, Wang MC (2017) Improved lane line detection algorithm based on hough transform. Pattern Recognit Image Anal 28(2)
10. Zhai H (2017) Vehicle license plate tilt correction algorithm based on improved projection detection. Logist Technol 36(1):89–93
11. Zhang W (2017) Research on target identification algorithm based on improved projection method. Fire Command Control 42(2):53–55+59
12. Fang X (2017) ID card segmentation algorithm based on projection method and Caffe framework. Comput Eng Appl 53(23):113–117
13. Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: International conference on learning representations
14. Hosang J, Benenson R, Schiele B (2006) Efficient non-maximum suppression. In: International conference on pattern recognition. IEEE Computer Society, pp 850–855
15. Neubeck A, Gool LV (2006) Efficient non-maximum suppression. In: International conference on pattern recognition. IEEE Computer Society, pp 850–855
16. Rothe R, Guillanumin M, Gool LV (2015) Non-maximum suppression for object detection by passing messages between windows. In: Computer vision—ACCV 2014. Springer International Publishing, pp 290–306

A Dependency-Extended Method of Multi-pattern Matching for RDF Graph

Yunhao Sun[1], Fengyu Li[1], Guanyu Li[1(✉)], Heng Chen[1,2], and Wei Jiang[1]

[1] DaLian Maritime University, Dalian 116026, China
rabitlee@163.com,syh8086@163.com

[2] DaLian University of Foreign Languages, Dalian 116044, China

Abstract. The problem of multi-pattern matching for RDF graph is an extended problem of subgraph isomorphism, where Resource Description Framework (RDF) is a graph-based data model for information sharing on Web. In real world, concurrent execution of multi-queries is more realistic than single query. However, the problem about re-computations of common subgraphs always limits the time efficiency of matching processing. To solve this problem, an algorithm of multi-pattern matching for RDF graph is proposed, which can response to multiple queries through one traversal of RDF graph. The experimental results show that our algorithm can avoid the re-computations of common subgraphs and improve up to 70% of time efficiency than basic line algorithm.

Keywords: RDF graph · Multi-pattern matching · Dependent tree · Node fragment table

1 Introduction

Resource Description Framework (RDF) [5] is a graph-based data model that is the basic model for semantic identification of resources in *Semantic Web* [4]. The pattern matching for RDF graph (PM for short) refers to finding all the RDF subgraphs that are isomorphic to query graphs. The multi-pattern matching for RDF graphs (M-PM for short) is an extension of PM problem. A core challenge of M-PM problem is to avoid the re-computations about the matching processing on common subgraphs of multiple query graphs. To cope with the core challenge, an algorithm of M-PM problem is proposed in this paper, which can response the multiple query graphs by one traversal on data graphs.

The most of researches pay more attention on the single-pattern matching processing and multi-patterns matching optimization. For the single-pattern matching processing, a relational approach is usually used to index and match RDF graphs. Weiss et al. [1] and Perez et al. [6] employ an index-based solution by indexing and matching RDF graphs on a B$^+$-tree. Srdjan Komazec et al. [7]

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 180–188, 2020.
https://doi.org/10.1007/978-981-15-0187-6_21

vertically partition the RDF graphs into a set of tables based on bounded labels of triple patterns and use a bidirectional index on top of it to locate the required tables. However, it is hard to migrate the technique of single-pattern matching to M-PM problem, due to the complex dependent relations among multiple query graphs.

The multi-pattern's matching optimization is to identify common tasks among multiple query graphs and select one exact plan for each query graph. Ismail et al. [8] propose a dynamic programming solution. The solution can avoid to generate redundant candidates and reaches the solution set before generating larger candidates. Zahid Abul-Basher et al. [2] propose a framework for multiple query optimization. The key idea of this framework is to expand repeatedly the search wavefront until no new answers are produced, where each search wavefront is guided by non-deterministic finite automata. However, those heuristic algorithms pay a big cost on training plans.

Therefore, a novel algorithm of M-PM problem is proposed. Firstly, a dependent tree (D-Tree) is designed for reflecting the inclusion of relation among multiple queries. D-Tree exactly describes the executed orders of multiple BGP graphs. However, the inclusion relations are difficult to operate the crossover between any two inclusive BGP graphs, because inclusion relation is an abstract and single representation. Then, a node fragmentation table (NFT) is proposed to solve the crossover problem between inclusive BGP graphs. NFT depicts the inclusion relations in detail based on residual edges among BGP graphs. Finally, a M-PM algorithm is designed based on D-Tree and NFT, which can response to multiple BGP graphs through one traversal of RDF graph. M-PM algorithm can effectively avoid the re-computations about the matching processing on common subgraphs of multiple query graphs.

2 Dependent Tree and Node Fragment Tables

The formal definitions of basic graph pattern (BGP) and data graph (DG) are first given, and then, our problem definition is given.

2.1 Problem Definition

Definition 1. (*Basic Graph Pattern*) A basic graph pattern BGP(V_q, E_q, L_q, ϕ_q, $vars_q$) is a directed labeled graph, where V_q is a set of vertexes; E_q represents a multi-set of directed edges; $E_q : (u, v)$ is a directed function denoting a directed edge from u to v, $u, v \in V_q$; L_q is a set of edge and vertex labels; $vars$ is a set of query variables, and ${}_q$: $V_q \cup E_q \rightarrow (L_q \cup vars_q)$ is a labeling function that maps a vertex or an edge to the corresponding label.

Definition 2. (*Data Graph*) A data graph DG(V_d, E_d, L_d, ϕ_d) is a directed labeled graph, where V_d is a set of vertexes; E_d represents a multi-set of directed edges; $E_d : (u, v)$ is a directed function denoting a directed edge from u to v, $u, v \in V_d$; L_d is a set of edge and vertex labels, and ${}_d$: $V_d \cup E_d \rightarrow (L_d \cup vars_d)$ is a labeling function that maps a vertex or an edge to the corresponding label.

The difference between BGP and DG is that the label of DG must be a constant and the label of BGP can be a variable. The variables at the predicate position are not considered in this paper because such variables are not common in real-world query graphs, as shown in a previous study [3].

The basic graph pattern matching (B-PM) refers to finding all data subgraphs that are isomorphic to a BGP. However, concurrent execution of multi-queries is more realistic than single query in real world. And re-computations of common subgraphs always limit the time efficiency of matching processing. Therefore, an extended problem definition of B-PM is given as follows.

Definition 3. (*M-PM Problem*) Given a set of basic graph patterns Q =(q_1, q_2,$\cdots$ q_m) and a set of data graphs D=(d_1, d_2, $\cdots$, d_n), $m, n \in N^+$. A multi-pattern matching (M-PM) refers to finding all data subgraphs from D that are isomorphic to basic graph patterns of Q.

To reflect the inclusion relation among multiple BGP graphs, a dependent tree (D-Tree) is designed. D-Tree exactly describes the executed orders of multiple BGP graphs. However, the inclusion relations are difficult to operate the crossover between any two inclusive BGP graphs, because inclusion relation is an abstract and single representation. Therefore, a node fragmentation table (NFT) is proposed to solve the crossover problem between inclusive BGP graphs. NFT detailedly depicts the inclusion relation based on residual edges among BGP graphs. Based on D-Tree and NFT, a M-PM algorithm is designed to response to multiple BGP graphs through one traversal of RDF graph. The working mechanisms about D-Tree and NFT are explained in Sect. 2.2.

2.2 Dependent Tree and Node Fragmentation Table

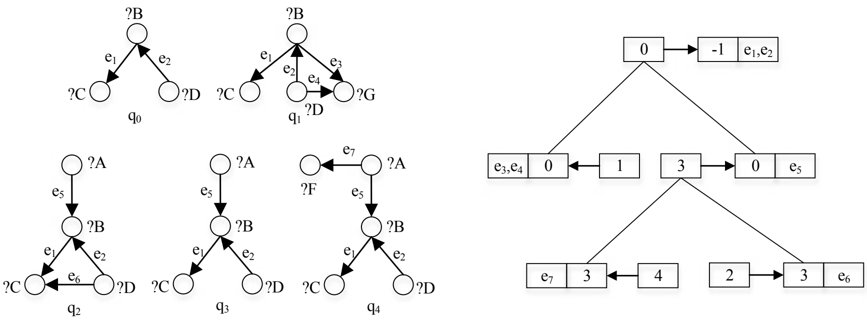
A D-Tree describes the inclusion relations among multiple BGP graphs.

The dependent relation among BGP graphs is presented by the dominating set of queries. Given any two BGP graphs q_1 and q_2, satisfying $E_{q_1} \subset E_{q_2}$, then q_1 dominates q_2, defined as $q_1 \prec q_2$. Given any two BGP graphs q_1, q_2, and an edge e, satisfying $e \notin q_1$, $e \in q_2$ and $q_1 \prec q_2$, then e is a residual edge of q_2 (RE_{q_2} for short). That is, q_2 is at rear of q_1 in matching processing if and only if $q_1 \prec q_2$.

Figure 1 represents the multiple BGP graphs and a dependent Tree. Figure 1. (1) depicts the multiple BGP graphs, and there are dependent relations among BGP graphs. Figure 2b presents a D-Tree, where 0, 1, . . ., n refer to the subscripts of q_0, q_1, . . ., q_n, $n \in N^+$ and -1 is a subscript of virtual node that assigned to a root node of each tree. Each node of D-Tree also contains the residual edges with its parent node.

A node fragmentation table (NFT) is to solve the crossover problem between inclusive BGP graphs. NFT detailedly depicts the inclusion relation among BGP graphs based on residual edges.

NFT is classified as inner NFT and outer NFT, where inner and outer NFTs are constructed by inner and outer edges, respectively. Given a BGP graph q and a vertex $u \in V_q$, there is a vertex v of V_q, such that $e(u,\ v) \in E_q$, then edge e is an inner edge of vertex u. Given any two BGP graphs q_1, q_2 and a



(a) Multiple BGP Graphs

(b) A D-Tree of Multiple BGP Graphs

Fig. 1. Multiple BGP graphs and a dependent tree (D-Tree)

vertex u, $u \in V_{q_1}$ and $u \in V_{q_2}$, there is a vertex v of q_2, satisfying $q_1 \prec q_2$, such that $e(u, v) \notin E_{q_1}$ and $e(u, v) \in E_{q_2}$, then edge e is a outer edge of vertex u. NFT containing inner edges is called as inner NFT (iNFT for short) and NFT containing outer edges is called as outer NFT (oNFT for short).

NFT contains two columns: vertexes and edges. In each row of NFT, the vertex unit only holds one vertex, and the edge unit holds the inner or outer edges coupled with vertex unit. Thus, the inner NFT and outer NFT are formed as iNFT(V_q, iE_{V_q}) and oNFT(V_q, oE_{V_q}), respectively.

The NFT is constructed based on D-Tree. In D-Tree, the vertexes and edges of a root node are only assigned into iNFT, because no node dominates a root node. Given any a node q_1 (not a root node), its parent node q_2 and a vertex u of q_2, there is a vertex v of q_1, such that $e(u, v) \in RE_{q_1}$, then u and $e(u, v)$ are assigned into oNFT. Given any a node q_1 (not a root node) and a vertex u of q_1, there is a vertex v of q_1, such that $e(u, v) \in RE_{q_1}$, then u and $e(u, v)$ are assigned into iNFT.

Figure 2 describes the node fragment table (NFT). The parent of q_1 is q_0, which can be found from Fig. 1b. For a vertex $?B$, there is a vertex $?G$ of q_1, such that $e_3(?B, ?G) \in RE_{q_1} = \{e_3, e_4\}$, then $?B$ and e_3 are assigned into oNFT. For a vertex $?G$, there is a vertex $?B$ of q_1, such that $e_3(?G, ?B) \in RE_{q_1} = \{e_3, e_4\}$,

?B	[e_1, e_2]
?C	[e_1]
?D	[e_2]
?G	[e_3,e_4]
?A	[e_5]
?F	[e_7]

(a) Inner Table

?B	[e_3,e_5]
?C	[e_6]
?D	[e_4,e_6]
?A	[e_7]

(b) Outer Table

Fig. 2. Node fragment table (NFT)

then $?G$ and e_3 are assigned into iNFT. For a root node of D-Tree, all vertexes and edges are assigned into iNFT. Note that the directed characteristic of edges are not considered; that is, NFT is executed on ground graphs of directed BGP graphs.

3 M-PM Algorithm

M-PM algorithm is based on D-Tree and NFT. The matched results were response sequentially with multiple BGP graphs through a pre-order traversal on D-Tree.

Before description of MP algorithm, a vertex relation between BGP and data graph is given. Given any a vertex of data graph v_d, there is a pattern vertex v_q of one BGP graph matching v_d, then v_d is an instance of v_q. Given any two vertexes of data graph v_{d_1} and v_{d_2}, there are two pattern vertexes v_{q_1} and v_{q_2}, satisfying $e\ (v_{d_1}, v_{d_2})= e(v_{q_1}, v_{q_2})$, v_{d_1} is an instance of v_{q_1} and v_{d_2} is an instance of v_{q_2}, then $d_t\langle v_{d_1}, e, v_{d_2}\rangle$ is a triple instance of $d_t\langle v_{q_1}, e, v_{q_2}\rangle$.

In M-PM algorithm, the results of multiple query graphs are incrementally acquired through a pre-order traversal on D-Tree. The process of obtaining the results of root node q_0 is described as follows:

(1) Giving a fixed pattern vertex v_{q_0} of q_0, acquiring a data vertex v_d that is an instance of v_{q_0} and setting v_d as an initial data vertex.

The fixed pattern vertexes are chosen by computation of inner table and D-Tree. A pattern vertex v_{q_0} is fixed if and only if $RE_{q_0} \cap iE_{v_{q_0}} \neq \emptyset$. The fixed pattern vertex v_{q_0} is obtained by a function F_i: $RE_{q_0} \cap iNFT \rightarrow \langle v_{q_0}, iE_{v_{q_0}}\rangle$.

(2) Obtaining other vertexes sequentially by one-hop traversal from v_d and setting as initial data vertexes of next one-hop traversal.

In one-hop traversal on data graphs, given any an initial data vertex v_{d_1} and its a obtained vertex v_{d_2}, there is a pattern vertex v_q, satisfying v_{d_1} is an instance of v_q and $e(v_{d_1}, v_{d_2}) \in iE_{v_q}$, then triple instance $d_t(v_{d_1}, e, v_{d_2})$ is written into the results of q_0. And if $e(v_{d_1}, v_{d_2}) \in oE_{v_q}$, triple instance $d_t\langle v_{d_1}, e, v_{d_2}\rangle$ is written into temp results.

(3) Carrying out the results of q_0 until there are not new triple instances to be written.

The process of obtaining the results of child nodes is different with root node. It needs to be assisted by parent nodes and temp results.

(1) Copying parent node results to child node results, because parent node results are a part of child node results.

(2) Giving a pattern node v_{q_1} of the child node q_1, then acquiring a data vertex v_d that is an instance of v_{q_1} and setting v_d as an initial data vertex on temp results.

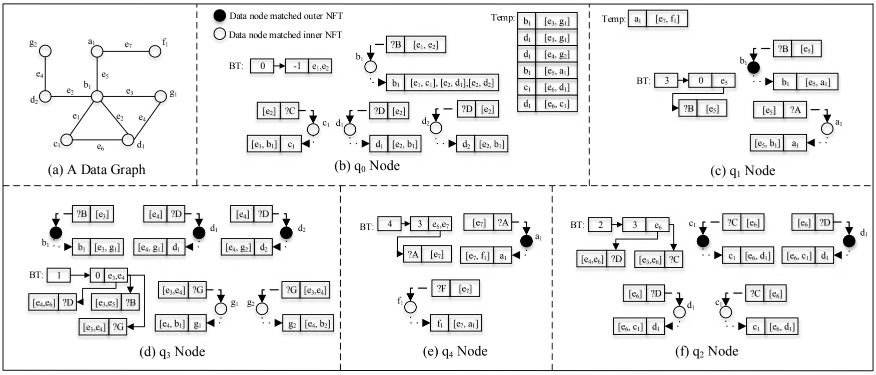
The fixed pattern vertexes are chosen by computation of outer table and D-Tree. A pattern vertex v_{q_0} is fixed if and only if $RE_{q_0} \cap oE_{v_{q_0}} \neq \emptyset$. The fixed pattern vertex v_{q_0} is obtained by a function F_o: $RE_{q_0} \cap oNFT \rightarrow \langle v_{q_0}, oE_{v_{q_0}}\rangle$.

(3) Obtaining other vertexes of temp results sequentially by one-hop traversal from v_d and setting as initial data vertexes of next one-hop traversal.

In one-hop traversal on temp results, given any an initial data vertex v_{d_1} and its a obtained vertex v_{d_2}, there is a pattern vertex v_{q_1}, satisfying v_{d_1} is an instance of v_{q_1} and $e(v_{d_1}, v_{d_2}) \in RE_{q_1} \cap oE_{v_{q_1}}$, then triple instance $d_t(v_{d_1}, e, v_{d_2})$ is written into the results of q_1. The one-hop traversal on temp results is continuous executed until no new triple instance of temp results is written.

(4) Taking final initial data vertexes on temp results as initial data vertexes on data graphs, and executing the (2) and (3) in the process of obtaining the results of root node.

(5) Re-executing (1)–(4) until results of all BGP graphs are obtained.



(a) A Data Graph

(b) q_0 Node

(c) q_1 Node

(d) q_3 Node

(e) q_4 Node

(f) q_2 Node

Fig. 3. Processing of multi-patterns matching for RDF graphs

The processing of M-PM is described in Fig. 3a represents a matched data graph. (b)–(f) depicts the obtained results of multiple BGP graphs on D-Tree in Fig. 2.

$\mathbf{q_0}$. The results of q_0 are only assisted with inner table, because q_0 is a root node of D-Tree. Firstly, F_i: $RE_{q_0} \cap iNFT \rightarrow \langle ?B,\, iE_{?B}(e_1, e_2)\rangle$, $\langle ?C,\, iE_{?C}(e_1)\rangle$ and $\langle ?D,\, iE_{?D}(e_2)\rangle$, and arbitrarily choosing one of ?B, ?C, and ?D as a fixed pattern vertex. Assumed that ?B is a fixed pattern vertex, then b_1 is an instance of ?B and set as an initial data vertex on (a). Through one-hop traversal from b_1, triple instances $[b_1, [e_1, c_1], [e_2, d_1], [e_2, d_2]]$ are written into the results of q_0, and triple instances $[b_1, [e_3, a_1], [e_5, g_1]]$ are written into temp results. Then, c_1, d_1 and d_2 are obtained sequentially and set as the next initial vertexes, respectively. Through one-hop traversals of c_1, d_1, and d_2, triple instances $[c_1, [e_1, b_1]]$, $[d_1, [e_2, b_1]]$, and $[d_2, [e_2, b_1]]$ are written into the results of q_0, and triple instances $[d_1, [e_4, g_1]]$, $[d_2, [e_4, g_2]]$, $[c_1, [e_6, d_1]]$ and $[d_1, [e_6, c_1]]$ are written into temp results. Finally, final results of q_0 are carried out because b_1 has included into results of q_0.

$\mathbf{q_1}$. The results of q_1 are assisted with inner and outer tables, and final results of q_0 are copied to the results of q_1, because q_0 is a parent node of q_1. Firstly, $RE_{q_1} \cap oNFT{=}\langle ?B,\, oE_{?B}(e_5)\rangle$; thus, ?B is a fixed pattern vertex, then b_1 is an instance of ?B and set as an initial data vertex on temp results. Through one-hop traversal from b_1, triple instance $[b_1, [e_5, a_1]]$ is written into the results of q_1. Then, a_1 is obtained and set as an initial vertex on (1), because a_1 is not included into the key of temp results. Due to $RE_{q_1} \cap iNFT{=}\langle ?A,\ iE_{?A}(e_5)\rangle$, triple instance $[a_1, [e_5, b_1]]$ is written into the results of q_1, and $[a_1, [e_7, f_1]]$ is written into temp results through one-hop traversal from a_1. Finally, final results of q_1 are carried out because b_1 has included into results of q_1.

The processing of obtained results of q_1, q_2, and q_3 are similar to q_1. Through M-PM algorithm, the results of multiple BGP graphs can be obtained by one traversal of data graphs.

4 Experimental Evaluation

Query Sets. In the selection of query sets, residual edges of query node are an important factor. In the D-Tree of query graphs, the residual edges of non-leaf query node are associated with overlapped query subgraphs. As the depth of D-tree increases, the closer to root node are and the more times the query node overlap. Therefore, the multiple query graphs with the same and different depths are selected into our query sets.

Datasets. A simulated dataset is selected in our experimental analysis. *DBpedia 2015A*[1] describes the information of sports and sport events. It contains 30,000 RDF triple instances with 20 triple patterns. Due to the small number of triple patterns, it is different for *DBpedia 2015A* to reflect the benefits of multi-pattern matching algorithm. Therefore, a simulated dataset based on *DBpedia 2015A* is created, which contains almost 3 million RDF triple instances with 300 triple patterns.

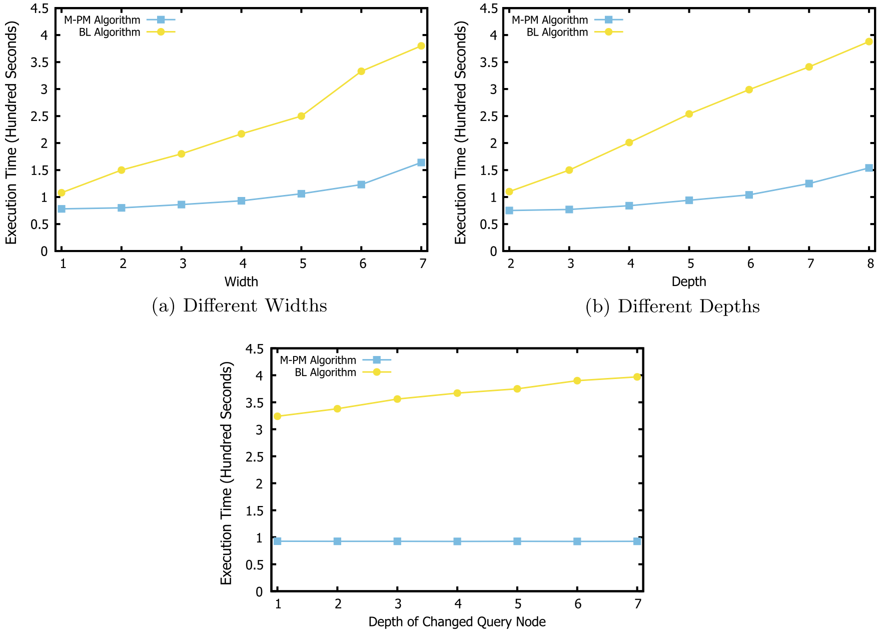
Configurations. All experiments were performed on the Intel Xeon 5118 processor with 24GB RAM. The system is equipped with main memory, and it runs a 64-bit Linux 3.13.0 kernel.

In this experimental analysis, we look at the multi-pattern matching with same and different depth of D-Tree on simulated dataset.

Basic line (BL for short) algorithm is compared with M-PM algorithm. The idea of BL algorithm is to reduce the size of original RDF graph by query parent node and response query child node on reduced RDF graph. However, it does not avoid the rematching on the overlapped RDF subgraphs between parent and child node. Assumed that the size of multiple BGP graphs is n and the quantity of data graphs is m (quantity statistics of triple instance). The time complexity of BL algorithm is similar to $O(n \cdot m)$. And the time complexity of M-PM algorithm is approximatively $O(m) + O(log_m^n \cdot m)$. Therefore, M-PM algorithm is better time-efficient than basic line algorithm.

Figure 4a describes the execution time on different widths of D-Tree with same depth. Figure 4b depicts the execution time on different depths of D-Tree with same width. The root query node contains 80 triple patterns, and each child node has 10 edges more than its parent node. As the width and depth increase,

[1] https://wiki.dbpedia.org/dbpedia-data-set-2015-04.



(a) Different Widths

(b) Different Depths

(c) Changed Depths

Fig. 4. Execution time evaluation

time cost of BL algorithm is much faster than M-PM algorithm. In the linear tread, M-PM algorithm can improve up to 70% of time efficiency compared with BL algorithm. And execution time of M-PM and BL algorithms is roughly the same without differences in depth and width. That is, the execution time of M-PM and BL algorithms is only related to number of *RE*.

Therefore, execution time about depth of changed query node is described in Fig. 4c. The total number of *RE*s is same, and the number of *RE*s with different depths is changed. The deeper depth is, the less number of overlapped subgraph is. In the linear trend, M-PM algorithm changes very litter and BL algorithm keeps growing as the depth increases. That is, M-PM algorithm does not change due to the times of overlapped subgraph increases.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant (No. 61371090, No. 61602076 and No. 61702072). The China Postdoctoral Science Foundation Funded Project (2017M621122 and 2017M611211). The Natural Science Foundation of Liaoning Province(No. 20170540144, No. 20170540232 and No. 20180540003).

References

1. Abadi DJ, Marcus A, Madden S, Hollenbach K (2009) Sw-store: a vertically partitioned DBMS for semantic web data management. VLDB J 18(2):385–406

2. Abul-Basher Z, Yakovets N, Godfrey P, Chignell MH (2016) Swarmguide towards multiple-query optimization in graph databases. In: AMW
3. Arias M, Fernández JD, Martínez-Prieto MA, de la Fuente P (2011) An empirical study of real-world SPARQL queries. *CoRR*, abs/1103.5043
4. Berners-Lee T, Hendler J, Lassila O et al (2001) The semantic web. Sci Am 284(5):28–37
5. Bizer C, Vidal M-E, Weiss M (2018) Resource description framework, 2nd edn. In: Encyclopedia of database systems
6. Broekstra J, Kampman A, Van Harmelen F (2002) Sesame: a generic architecture for storing and querying rdf and rdf schema. In: International semantic web conference. Springer, pp 54–68
7. Komazec S, Cerri D, Fensel D (2012) Sparkwave: continuous schema-enhanced pattern matching over RDF data streams. In: Proceedings of the sixth ACM international conference on distributed event-based systems, DEBS, Berlin, Germany, 16–20 July 2012
8. Toroslu IH, Cosar A (2004) Dynamic programming solution for multiple query optimization problem. Inf Process Lett 92(3):149–155

Machine Learning for RF Fingerprinting Extraction and Identification of Soft-Defined Radio Devices

Su Hu[1](✉), Pei Wang[1], Yaping Peng[1], Di Lin[1], Yuan Gao[2], Jiang Cao[2], and Bin Yu[3]

[1] University of Electronic Science and Technology of China,
Chengdu 611731, Sichuan, China
husu@uestc.edu.cn

[2] Academy of Military Science of PLA, Beijing 100090, China

[3] Beijing Samsung Telecom R&D Center, Beijing 100081, China

Abstract. Radio frequency (RF) fingerprinting technology has been developed as a unique method for maintaining security based on physical layer characteristics. In this paper, we propose the RF fingerprinting by extracting the parameter characteristics such as information dimension, constellation feature, and phase noise spectrum in the transmitted information when applied to the universal software radio peripheral (USRP) software-defined radio (SDR) platform. To achieve a great performance improvement of classification, not only the traditional support vector machine (SVM) classifier, but also the machine-based integrated classifier bagged tree and the adaptive weighting algorithm weighted k-nearest neighbor (KNN) are both discussed. It is demonstrated that the proposed method achieves good classification performance under different signal-to-noise ratios (SNR).

Keywords: RF fingerprinting · Wireless communication · Software-defined radio · Characteristic parameter · Classifier

1 Introduction

With the continuous popularization of mobile communication devices and the rapid development of the Internet of things (IoT) technology, wireless communication plays an irreplaceable role in both military and civilian applications. However, due to its openness, wireless networks are more vulnerable to large-scale malicious attacks than traditional wired networks. Meanwhile owing to artificial intelligence technology [1], several new low-cost devices are sensitive to computational complexity. And traditional methods by employ IP or MAC addresses as identity are not effective as usual. Therefore, in order to reduce potential threats from malicious users, it is urgent to find a new type of security mechanism to identify authorized users and unauthorized users effectively.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 189–204, 2020.
https://doi.org/10.1007/978-981-15-0187-6_22

In the past ten years, RF fingerprint extraction and identification technology [2] of wireless communication equipment have received extensive attention all over the world. This method extracts the 'RF fingerprint' of the device by analyzing the communication signal of the wireless device. Due to the hardware characteristics, different wireless devices have their own unique RF fingerprints, which can be extracted by analyzing the received RF signals. The method of extracting hardware features of devices based on communication signals is called 'radio frequency fingerprint extraction,' and the method of identifying different wireless devices using radio frequency fingerprints is called 'radio frequency identification.'

In the field of correctly identifying communication devices and increasing wireless communication security, scholars have done a lot of researches on 'radio frequency fingerprint extraction.' Most achieve RF identification by analyzing the transient signals, such as extracting the radio frequency fingerprint in the Bluetooth communication signal [3]. However, this technology requires too much precision of identification device. Recently, RF fingerprint extraction based on steady-state signals and identification technology has received widespread attention [4,5].

This paper proposes a classification method by integrating different RF fingerprint features and unique classifiers and carries out extensive experiments to evaluate the performance. In particular, we employ six features, including two fractal dimension features, phase noise spectrum, and three constellation features extracted from the received information. Three classifiers are adopted to adaptively combine different features. A test bed is constructed by a low-cost USRP SDR platform as transceivers. During our work, extensive experiments to investigate the classification performance under different channel conditions are carried out. And experimental results show that the proposed RF fingerprinting identification method has an excellent classification performance even at low SNR conditions.

The content of this paper is arranged as follows: Sect. 2 introduces the extraction and recognition model of RF fingerprinting. Section 3 explains the main fingerprint features extracted in the experiment and their physical meanings. In Sect. 4, we describe the construction of the RF fingerprinting experimental environment and the process of data processing. Section 5 shows the main recognition results for different classifiers and different identification parameters. Finally, this paper is summarized in Sect. 6.

2 Overview of RF Fingerprinting Model

The RF fingerprinting extraction and identification process of wireless communication device [6] are mainly composed of data generation and acquisition module, as well as data, preprocess, and classification. As shown in Fig. 1, during data generation and acquisition module, after the radio frequency signal is sent by the wireless transmitter, the received signal is sent to the identification system through some necessary process. In Fig. 2, during the data classification process, RF fingerprinting features are extracted after the preprocess is accomplished and sent to the classifier for the purposes of training and testing.

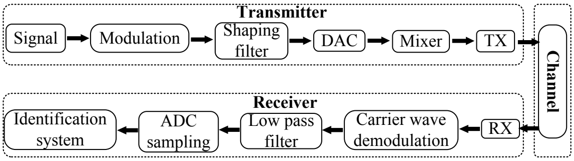


Fig. 1. Wireless communication fingerprinting data generation and acquisition model

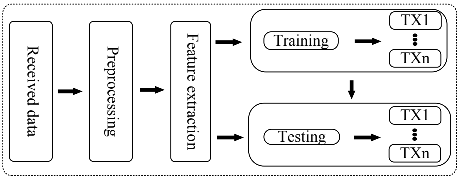


Fig. 2. Wireless communication fingerprinting data preprocess and classification model

Due to the tolerance of analog devices in the transmitter, the baseband signal will produce a unique fingerprint signature during digital processing in the transmitter such as filter, digital-to-analog converter, and mixer. Whether it is an integrated circuit or a non-integrated circuit, its essence is composed of electronic components, and the tolerance of the electronic components causes a tolerance effect of the final device. This includes the tolerances of the electrical parameters and standard values of the electronic components due to the material and processing error of the components during the production process of the components. As well as the parameters of the component working process caused by the different degrees of component aging and changes in the working environment of the components. That results in a certain difference in the hardware parameters of the wireless communication device, which are manifested as the frequency offset of the oscillator, the phase noise, the nonlinear distortion of the power amplifier, and the distortion of the filter. These hardware differences are important parameters for identifying the fingerprint characteristics of a wireless transmitter. For an input signal $w\left(t\right)$ through the transmitter, the transmission signal after TX antennas is given by:

$$A_{tx} = F_{tx}\left[w\left(t\right)\right] \cdot e^{j\left(2\pi f_{tx} + \phi_{tx}\right)}, \tag{1}$$

where $F_{tx}\left(\cdot\right)$ represent the transmitter function; f_{tx} and ϕ_{tx} are the phase and frequency offset of the transmitter, respectively. The transmit waves then propagate through the wireless channel and finally reach the RX antenna at the receiver. The arrived signal is expressed as:

$$A_{rx} = A_{tx} \otimes h\left(t\right) + n\left(t\right), \tag{2}$$

where $h\left(t\right)$ represent the channel function and $n\left(t\right)$ represent the noise interference.

Based on Eqs. (1) and (2), in addition to the hardware characteristics of the transmitter, the signal is also affected by the channel. Thus, the characteristics of the received signal are divided into channel-based features and transmitter-based features. Assuming the signals go through the same channel, the characteristics of the identification signals are essentially derived from transmitter-based features, also known as device fingerprints.

According to the requirements of signal fingerprint extraction, the identification system needs to carry out several preprocesses, such as phase compensation, energy normalization, and discarding the unqualified signal. The fingerprint recognition system detects the preprocessed signal transformation and extracts the relevant signal features in the time domain, the frequency domain, or the wavelet domain. Then the feature vector of the identification signal is composed by the extracted features. Finally, as shown in Fig. 2, it becomes a typical classification problem consisting of two parts: training and testing. During the training stage, the transmitted signals of all target transmitters are tagged with unique tags to form a signal recognition fingerprint library. In addition, during the testing stage, the relevant received signal features are extracted and compared with the fingerprint database to obtain the recognition results. Through the above analysis, it is easy to see that there are two main factors affecting the identification efficiency: the selection of signal characteristics and classifiers.

3 Recognition Methods

Due to the characteristics of the transmitter hardware, the received signal contains various FR fingerprint features. Hence, the appropriate selection of signal features is an important factor for RF fingerprint classification. From the existing research, it is known that RF fingerprints include time-domain envelopes, wavelet coefficients, and so on [7]. In this paper, fractal features, phase noise spectrum, and constellation features are selected as feature vectors.

3.1 Fractal Dimension

In recent years, the fractal theory has been widely concerned because it can effectively measure the complexity and irregularity of signals [8] and has been successfully applied in radar radiation source signals. Similar to radar signals, the characteristics of the wireless communication signals are mainly reflected in the variation, and distribution of frequency, phase, and amplitude. Therefore, the signal pulse can be identified by measuring the complexity of the signal waveform. In [8], the box dimension in fractal theory can reflect the geometric scale of the fractal set. The information dimension can reflect the distribution of the fractal set in the regional space. And the wireless communication signal as a time series can be characterized by the fractal dimension.

The calculation processes of the wireless communication source signal box dimension are shown as follows. The signal sequence $\{s\left(i\right), i = 1, 2, \ldots, N\}$ is obtained by preprocessing the communication signal and extracting the signal envelope, where N is the length of the signal sequence; place the signal sequence $\{s\left(i\right)\}$ in the unit square, and the minimum interval on the abscissa is $d = 1/N$, let

$$N\left(d\right) = N + \frac{\left\{\sum\limits_{i=1}^{N-1} \max\left[s\left(i\right), s\left(i+1\right)\right]d - \sum\limits_{i=1}^{N-1} \min\left[s\left(i\right), s\left(i+1\right)\right]d\right\}}{d^2}. \tag{3}$$

Calculate box dimension:

$$D_b = -\frac{\ln N\left(d\right)}{\ln d}. \tag{4}$$

The fractal box dimension only reflects the geometric scale. In order to utilize the distribution information of the fractal set in the regional space, the calculation processes of wireless communication source signal fractal dimension are shown as follows: preprocess the communication signal, extract the signal envelope, and obtain a signal sequence $\{s\left(i\right), i = 1, 2, \ldots, N\}$, where N is the length of the signal sequence; the signal sequence is reconstructed according to the following method to reduce the influence of part of the in-band noise, and at the same time, it is convenient to calculate the information dimension

$$s_0\left(i\right) = s\left(i+1\right) - s\left(i\right), i = 1, 2, \ldots, N-1. \tag{5}$$

Calculate fractal dimension, let $S = \sum\limits_{i=1}^{N-1} s_0\left(i\right)\ p\left(i\right) = \frac{s_0(i)}{S}$, thus

$$D_I = -\sum_{i=1}^{N-1} \{p\left(i\right) \times \lg\left[p\left(i\right)\right]\}. \tag{6}$$

Box dimension and information dimension are used to measure the complexity of the signal spectrum shape, which contains the information of the signal amplitude, frequency, and phase.

3.2 Phase Noise Spectrum

Phase noise [9] is an important indicator reflecting the frequency stability and frequency reliability of signal sources. In phase noise measurement technology, power spectrum estimation is an indispensable part of any phase noise measurement system.

In this paper, we choose Welch spectral estimation. The calculation method of Welch spectral estimation is as follows. Firstly, divide the signal data of length N into L segments, and each segment has M samples. Secondly, select the appropriate window function to weight each piece of data separately and determine the period diagram of each segment. Finally, analysis the power spectrum for each period diagram of each segment and compute the average of the N estimated results.

The formula of phase noise spectrum is derived as follows

$$\tilde{P}_W(\omega) = \frac{1}{L}\sum_{i=1}^{L}\hat{P}_i(\omega) = \frac{1}{MUL}\sum_{i=1}^{L}\left|\sum_{m=1}^{M}x_i(m)\,d(m)\exp(-j\omega m)\right|^2, \tag{7}$$

where $\tilde{P}_W(\omega)$ represents the result of spectrum estimation; $\hat{P}_i(\omega)$ represents the ith spectrum estimation $d(m)$ represents the window function, and $U = \frac{1}{M}\sum\limits_{m=1}^{M}|d(m)|^2$ represents the normalization factor. Equation (7) can be simplified as follows:

$$\begin{aligned}\tilde{P}_W(\omega) &= \frac{1}{MUL}\sum_{i=1}^{L}\left|\sum_{m=1}^{M}x_i(m)\,d(m)\exp(-j\omega m)\right|^2\\ &\approx \frac{1}{MU}\sum_{m=1}^{M}\sum_{n=1}^{M}d(m)\,d^*(m)\left[\frac{1}{L}\sum_{i=1}^{L}x_i(m)\,{x_i}^*(m)\right]\exp(-j\omega m)\\ &\approx \sum_{\tau=-(M-1)}^{M-1}W(\tau)\,\tilde{r}(\tau)\exp(-j\omega m)\end{aligned} \tag{8}$$

where $W(\tau) = \frac{1}{MU}\sum\limits_{m=1}^{M}d(m)\,d^*(m-\tau)$ represents the normalized power of the time window. In this paper, we adopt Hamming window and take carrier frequency $F_c = 50\,\mathrm{kHz}$, and then the phase noise spectrum estimation at $F_c = 50\,\mathrm{kHz}$ could be calculated.

3.3 Constellation Features

The constellation diagram reflects the intuitive geometric representation of the signal point set, and the analysis of constellation features can be used to distinguish between different modulation methods. Not only that, but the differences in transmitter hardware in the same modulation mode will be reflected in the constellation. Therefore, constellation features are important parameters for signal fingerprinting.

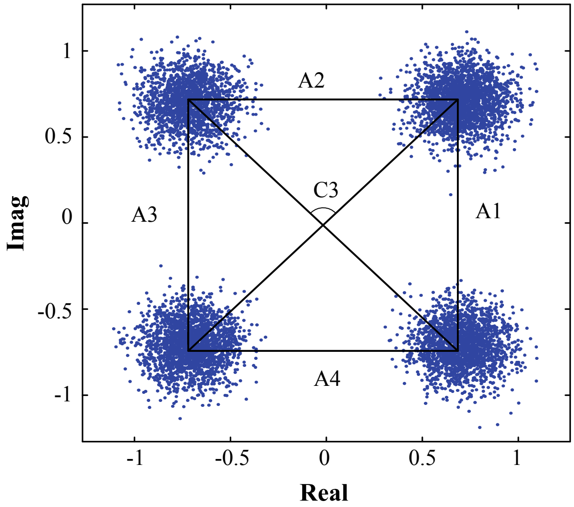


Fig. 3. Constellation diagram for QPSK with SNR = 10 dB

The constellation of QPSK signal when SNR = 10 dB is showed as Fig. 3. The average of the four constellation point sets is taken to form a quadrilateral. And A_1, A_2, A_3, A_4 represent the length of each side separately, K_1, K_2 represent two diagonals as Fig. 3 shows.

The constellation features are defined as follows

- c_1 represents the ratio of the longest to the shortest side length and can be expressed as

$$c_1 = \frac{\max\left(A_1, A_2, A_3, A_4\right)}{\min\left(A_1, A_2, A_3, A_4\right)}. \tag{9}$$

- c_2 represents the longest to shortest ratio of the diagonal and can be expressed as

$$c_2 = \frac{\max\left(K_1, K_2\right)}{\min\left(K_1, K_2\right)}. \tag{10}$$

- c_3 indicates the maximum angle between the diagonals and can be expressed as

$$c_3 = \pi - \arccos\left(\frac{K_1 \cdot K_2}{|K_1| \times |K_1|}\right). \tag{11}$$

3.4 SNR Estimation

In order to better analyze the classification results, SNR is selected as the performance measurement parameter, based on the practical need for signal detection under noncooperative communication conditions. In this paper, a small prior knowledge SNR estimation algorithm based on high-order moments is selected from many SNR estimation methods [10].

For a narrowband signal $s\left(t\right)$ interfered by additive white Gaussian noise $n\left(t\right)$, the received signal can be expressed as

$$f\left(t\right)=s\left(t\right)+n\left(t\right), \tag{12}$$

where $s\left(t\right)=\left[s_I\left(t\right)+js_Q\left(t\right)\right]e^{j\omega_0 t}$ $n\left(t\right)=\left[n_I\left(t\right)+jn_Q\left(t\right)\right]e^{j\omega_0 t}$. The second moment and the forth moment of $f\left(t\right)$ are defined as

$$m_2=E\left[f^*\left(t\right)f\left(t\right)\right], m_4=E\left\{\left[f^*\left(t\right)f\left(t\right)\right]^2\right\}. \tag{13}$$

According to the narrowband random signal theory, if P_s P_n denote the power of the signal and noise, then

$$\begin{aligned} E\left[s^{*2}\left(t\right)s^2\left(t\right)\right]&=\left\{E\left[s^*\left(t\right)s\left(t\right)\right]\right\}^2={P_s}^2\\ E\left[n^{*2}\left(t\right)n^2\left(t\right)\right]&=\left\{E\left[n^*\left(t\right)n\left(t\right)\right]\right\}^2={P_n}^2. \end{aligned} \tag{14}$$

After that the second moment and the forth moment of $f\left(t\right)$ can be rewritten as

$$m_2=P_s+P_n, m_4={P_s}^2+4P_sP_n+2{P_n}^2. \tag{15}$$

Based on Eqs. (14–15), SNR is obtained by

$$SNR=10\lg\left(\frac{P_s}{P_n}\right)=10\lg\left(\frac{\sqrt{2{m_2}^2-m_4}}{m_2-\sqrt{2{m_2}^2-m_4}}\right). \tag{16}$$

4 Experimental Environment Construction and Data Collection Classification

4.1 Experimental Environment Construction and Data Collection

The experimental system is shown in Fig. 4, which works at 2.4 GHz industrial and scientific band. The devices used in the experiment include two NI PXIe-1085 and three USRP-RIO-2943 devices. PXIe is a PC-based platform used for data transmission and processing at the transmitter and receiver side. USRP RIO-1 and RIO-2 with four antennas are used as four transmitters. USRP RIO-3 is used as the receiver for capturing RF signals with a sampling rate of 10 M sample/s. We aim to classify 4 USRP RIO transmitters by integrating the six features discussed in the above section, namely box dimension, fractal dimension, phase noise spectrum, three constellation features and combined using a classifier. The captured baseband signal from USRP is transferred to a PC and processed offline.

In the collection stage, we generate a series of random bits, perform QPSK modulation, and add carrier frequency $F_c = 50$ kHz for up-conversion processing. Each frame is composed of a data section with 8000 samples and a synchronize sequence with 256 samples. After that, use the PXIE to send the data through RIO TX cyclically. Then, each frame will be sent cyclically in a specify transmitting antenna and the same data frame is transmitted in these four transmitting antennas, while the receiving antenna record the signal over the real channel. 10,000 times per terminals are simulated in order to obtain a reliable result.

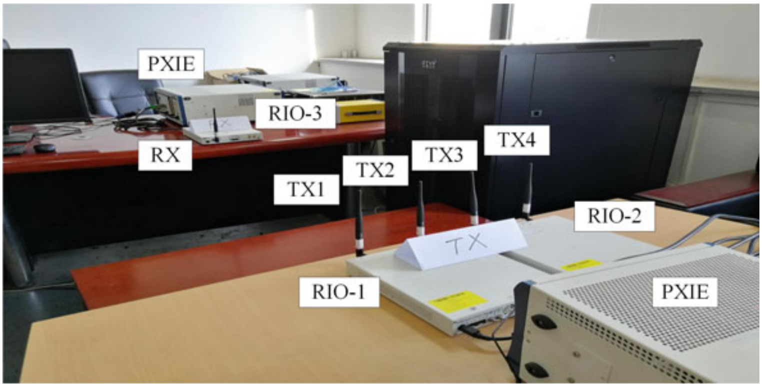


Fig. 4. Experimental environment for RF fingerprinting identification

4.2 Data Classification

In the classification stage, the acquired data will be processed and analyzed in MATLAB R2017b, including fingerprint extraction and classification. The captured data will be divided into five sets with the same size, four of which are used as training data and the remaining set of data is used for testing. Then cross-validation is applied to the data. During the cross-validation stage, the captured data will be tested for k times. Finally, the classification rate is averaged among the results. The classification rate β can be defined as $\beta = 1 - N_{error}/N_{test}$, where N_{error} is the number of classification error and N_{test} is the number of total test.

During this stage, three classifiers including bagged tree, weighted KNN, and fine Gaussian SVM are adopted for classification. Many schemes employing advanced classifier, such as linear regression, decision tree, KNN, SVM, naive Bayes, artificial neural network. As shown in Table 1, linear regression demands linear data with poor classification accuracy; decision tree can hardly handle missing data and always neglects the relationship between each other; the training of an artificial neural network requires a large amount of data; KNN needs a large amount of computation, and poor classification rate will be made when the sample is unbalanced; SVM is sensitive to the missing data when the sample imbalance is large; naive Bayes requires a priori probability with poor classification accuracy. Considering the complexity and effectiveness in the RF fingerprinting classification, in addition to the traditional SVM algorithm, we employ two more optimized classification algorithms: bagged tree and weighted KNN.

Bagged tree randomly selects P times in P training sets to form a new P training sets so that you can generate any number of new training sets based on one training data, and each training set also has a high repetition rate. In this way, the bagged tree can effectively reduce the ability of the classifier.

Weighted KNN-in traditional Euclidean distances, the weights of the features are the same, but obviously, it is not realistic. Equivalent weights make the

Table 1. Characteristics of common classifiers

Classifier	Complexity	Sensitivity	Classification accuracy	Special requirements
Linear regression	Low	Medium	Low	Linear data
Decision tree	Low	High	High	Neglected correlation
KNN	Medium	High	High	Complex computation
SVM	Medium	High	Medium	Complex computation
Naive Bayes	Low	Low	Low	Priori probability
Artificial neural network	High	Low	High	Massive data

calculation of similarity between feature vectors inaccurately affect classification. The weighted KNN weights the contributions of the Pth neighbors and assigns the larger weights to the nearest neighbors. The problem of imbalanced KNN sample is effectively solved by this method.

5 Experimental Result

Table 2 contains the average accuracy from four terminals by using one feature with SNR = 9.07 dB and three classifiers. It is obvious that these features are all distinguishable with a good identification rate, while constellation side length, constellation, angle and phase noise spectrum are the best features among them.

Table 2. Classification rate of single feature for QPSK with SNR = 9.07 dB and three classifiers

QPSK	Bagged tree (%)	Weighted KNN (%)	Fine Gaussian SVM (%)
Box dimension	91.9	92.1	77.3
Fractal dimension	89.2	90.2	72.0
Constellation side length	94.1	94.2	90.1
Constellation diagonal length	84.1	84.1	52.5
Constellation angle	93.4	93.8	89.7
Phase noise spectrum	93.8	94.1	89.7

For the classifier, weighted KNN achieves a better performance than bagged tree and fine Gaussian SVM in a single feature classification.

In order to more intuitively reflect the classification effect of different parameter features, multiple features are considered as the classification features for the purpose of visual comparison. Figure 5 illustrates the dot plot of 300 sample points per terminal with QPSK modulation when using box dimension, fractal dimension, and phase noise spectrum as input features. It can be seen that when those features are used as input features, except for some points overlapped with each other, others can be clearly distinguished. Figure 7 is the front view of Fig. 5, which also presents the dot plot when using box dimension and phase noise spectrum as input features. Figure 8 is the top view of Fig. 5, which presents the dot plot when using box dimension and fractal dimension as input features. Figure 6 shows the dot plot using constellation side length, constellation diagonal length, and constellation angle as input features. Figure 9 is the front view of Fig. 6, which presents the dot plot when using constellation diagonal length

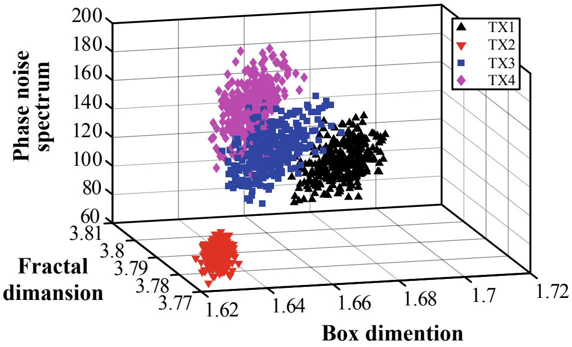


Fig. 5. View of box dimension, fractal dimension, and phase noise spectrum with SNR = 9.07 dB

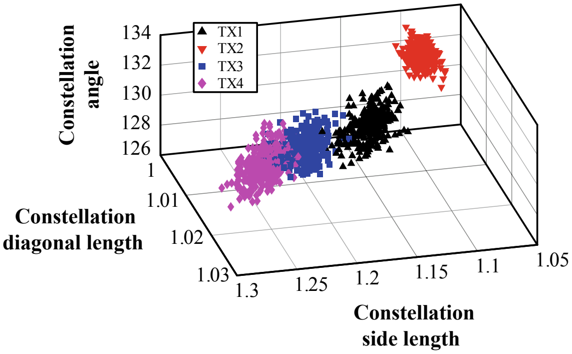


Fig. 6. View of constellation side length, constellation diagonal length, and constellation angle with SNR = 9.07 dB

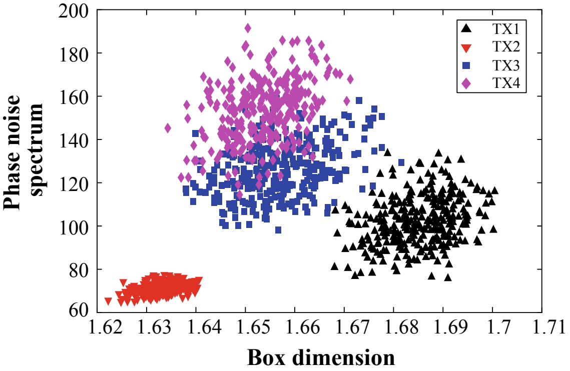


Fig. 7. View of box dimension and phase noise spectrum with SNR = 9.07 dB

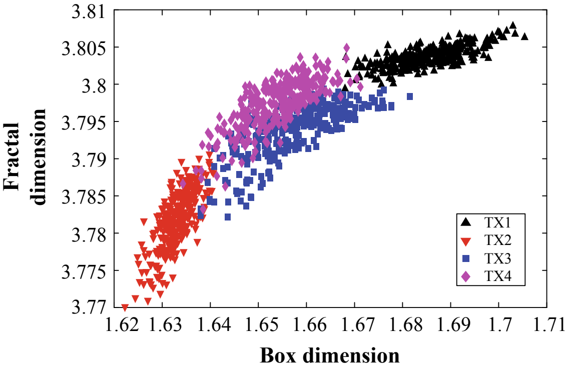


Fig. 8. View of box dimension and fractal dimension with SNR = 9.07 dB

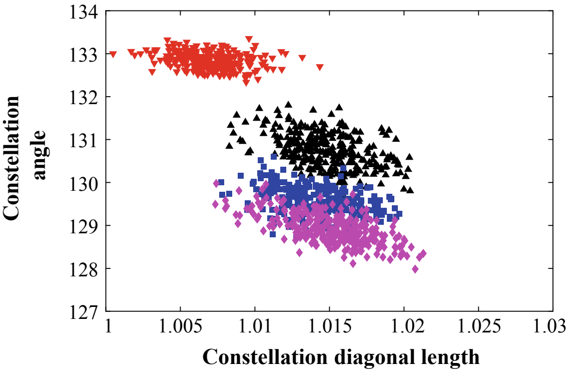


Fig. 9. View of constellation diagonal length and angle with SNR = 9.07 dB

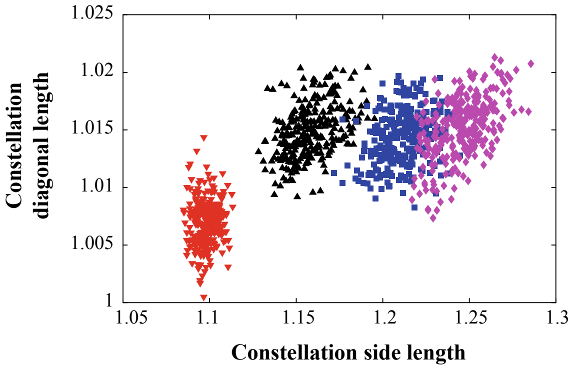


Fig. 10. View of constellation side length and diagonal length with SNR = 9.07 dB

and constellation angle as input features. Figure 10 is the front view of Fig. 6, which presents the dot plot using constellation side length and diagonal length as input features.

Table 3 records the average accuracy when multiple features are considered as the classification features with SNR = 9.07 dB and three classifiers. As shown in the table, the classification performance of three features combination performs better than two features combination. Compared with single feature input in Table 2, the performance superiority of bagged tree achieves better stability than other classification.

Table 3. Classification rate of several types of features for QPSK with SNR = 9.07 dB and three classifiers

QPSK	Bagged tree (%)	Weighted KNN	Fine Gaussian SVM
Box dimension and fractal dimension	97.2	94.7	85.6
Constellation side length and constellation diagonal length	97.5	95.8	88.9
Phase noise spectrum and constellation angle	97.7	96.7	90.9
Box dimension, fractal dimension, and phase noise spectrum	98.5	98.3	95.5
Constellation side length constellation diagonal length and constellation angle	98.2	98.0	94.9

In summary, the constellation feature does well than others. Either bagged tree or weighted KNN is better than fine Gaussian SVM, and the accuracy reaches to 98.5%. As the number of feature inputs increases, the performance of those three classifiers approaches similar. Also, it proves that the method of extracting feature values before using classifier is effective.

The above all based on the real laboratory environment does not change much; otherwise, the accuracy will be influenced. In order to better analyze the performance of the feature parameter identification method, the channel environment must also be taken into consideration. For the sake of simplicity, only the difference in SNR is considered. Since SNR in the experimental environment is difficult to change, we add Gaussian white noise to the signal emitted by the transmitter in this paper, and SNR of the received signal is the superposition of Gaussian white noise and channel noise, so that SNR can be changed.

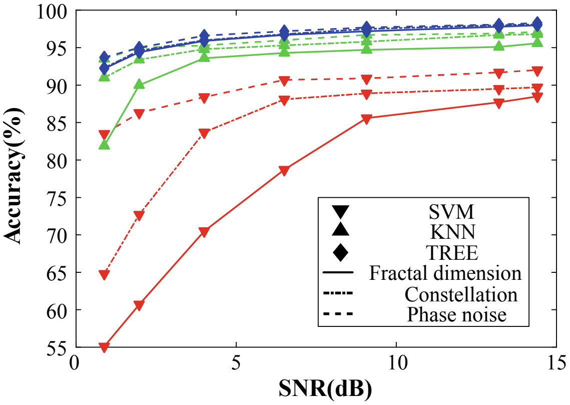


Fig. 11. Average classification accuracy versus SNR

Three different classifiers and three different combinations are adopted for training and testing. 10,000 tests per terminal are conducted to evaluate classification accuracy. Figure 11 plots classification accuracy from SNR = 0 to 15dB. As shown in Fig. 11, the following observations can be made: Considering the same parameter combination, for all three classifiers, the classification accuracy improves with an increasing SNR value; given the same SNR value, for all three parameter combinations, phase noise spectrum is better than the constellation feature and information dimension; under the same SNR value, for all three classifiers, bagged tree and weighted KNN classifiers are significantly better than fine Gaussian SVM classifier, and bagged tree is slightly better than weighted KNN classifier; as the SNR value increases, the performance gap of the classifier gradually shrinks, and bagged tree always maintains good performance.

6 Conclusion

This paper proposed a classification method by integrating different RF fingerprint features and unique classifiers and carried out extensive experiments to evaluate the performance. The contribution and novelty are three aspects. First, six different features in three special fields are adopted and found effective in classifying four terminals. Second, three classifiers were utilized to adaptively integrating features with the weights. Finally, a test bed consistsing of low-cost USRP devices as transceivers were constructed. Compared to the existing work, much more experiments were carried out to evaluate the performance of RF fingerprint under different channel conditions. In addition to bagged tree, weighted KNN such good classifiers, we will also design a more robust classifier by taking into account channel influences. Furthermore, the application of machine learning in signal fingerprinting [11] will be considered in our future work.

Acknowledgements. This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0807101, in part by the National Natural Science Foundation of China under Grant 61701503/61571082, and in part by the Ministry of Science and Technology of China (MOST) Program of International S&T Cooperation under Grant 2016YFE0123200. The work of S. Hu was supported in part by National Natural Science Foundation of China through the Research Fund for International Young Scientists under Grant 61750110527.

References

1. Hariyanto, Sudiro SA, Lukman S (2015) Minutiae matching algorithm using artificial neural network for fingerprint recognition. In: Proceedings of the 2015 3rd international conference on artificial intelligence, modelling and simulation (AIMS), Kota Kinabalu, Malaysia, Dec 2015
2. Cripps SC (2000) RF power amplifiers for wireless communications. IEEE Microw Mag 1(1):64–64
3. Eagle N, Pentland A (2006) Reality mining: sensing complex social systems. Pers Ubiquit Comput 10(4):255–268
4. Kennedy IO, Scanlon P, Mullany FJ, Buddhikot MM, Nolan KE, Rondeau TW (2008) Radio transmitter fingerprinting: a steady state frequency domain approach. In: Proceedings of the IEEE 68th vehicular technology conference (VTC), Calgary, BC, Canada, Sept 2008
5. Uzundurukan E, Ali AM, Kara A (2017) Design of low-cost modular RF front end for RF fingerprinting of bluetooth signals. In: Proceedings of the 25th signal processing and communications applications conference (SIU), Antalya, Turkey
6. Bertoncini C, Rudd K, Nousain B, Hinders M (2012) Wavelet fingerprinting of radio-frequency identification (RFID) tags. IEEE Trans Ind Electron 59(12):4843–4850
7. Rehman SU, Alam S, Ardekani IT (2014) An overview of radio frequency fingerprinting for low-end devices. In: Proceedings of the IGI Global
8. Wu L, Zhao YQ, Wang Z, Abdalla FYO, Ren GH (2017) Specific emitter identification using fractal features based on box-counting dimension and variance dimension. In: Proceedings of the 2017 IEEE international symposium on signal processing and information technology (ISSPIT), Bilbao, Spain, Dec 2017

9. Zhao CD, Huang MM, Huang LF, Du XJ, Guizani M (2017) A robust authentication scheme based on physical-layer phase noise fingerprint for emerging wireless networks. Comput Netw 128(9):164–171
10. Wang AF, Xu H (2012) Comparison of several SNR estimators for QPSK modulations. In: Proceedings of the 2012 international conference on computer science and service system, Nanjing, China, Aug 2012
11. Wu QY, Feres C, Kuzmenko D, Zhi D, Yu Z, Liu X (2018) Deep learning based RF fingerprinting for device identification and wireless security. Electron Lett 54(24):1405–1407

Analysis on Suppression of Echo Signal of Target Body and Translation in Micro-Doppler Signal Processing

Tang Bo(✉) and Sun Qiang

College of Weaponry Engineering, Naval University of Engineering,
Wuhan, Hubei, China
39023784@qq.com

Abstract. In order to suppress the echo signal of target body and translation in Micro-Doppler signal processing, the influence is analyzed based on the feature of echo signal. The method of quadrature reception and translation compensation is proposed to suppress the interference of target body echo signal and translation. Theoretical analysis and simulation show that this method can effectively suppress the interference caused by the echo signal of target body and translation.

Keywords: Micro-Doppler · Echo of target body · Translation · Object detection · Feature recognition

1 Introduction

With the extensive research and application of Micro-Doppler technology in radar field, it provides a new way and method for object detection and recognition [1]. By analyzing the echo signal of the target structural parts and extracting the Micro-Doppler feature, the physical and moving characteristics of the target structural parts can be obtained, and the object detection, feature recognition and classification can be carried out [2].

However, the echo of the target structure part is always obtained along with the echo of the target body, and at the same time, it is accompanied by the translation information. These are all interference for the Micro-Doppler feature extraction of the echo signal [3]. Therefore, suppressing the target body echo and translation must be considered in the Micro-Doppler feature extraction [4].

In this paper, suppression of target body echo and translation are analyzed mainly with the rotation of propellers in water.

2 Echo Signal Analysis

Assume that the acoustic detection equipment emits a single-frequency continuous wave with a frequency of f_c [5], i.e.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 205–211, 2020.
https://doi.org/10.1007/978-981-15-0187-6_23

$$s_t(t) = \exp(j2\pi f_c t) \tag{2.1}$$

Then, the echo signal of scattering point p received by the acoustic detection equipment is

$$s_r(t) = a\exp\left\{j2\pi f_c\left[t - \frac{r(t)}{c}\right]\right\} \tag{2.2}$$

where a is the amplitude of echo signal, c is the sound speed, $r(t) = R_0 + vt + R_{\text{rotating}}\hat{r}_0$ is the distance between the scattering point p and the acoustic detection equipment at time t, where R_0 is the distance from the origin of the acoustic detection device to the reference coordinate system, v is the target translational velocity, R_{rotating} is the rotation matrix in the reference coordinate system, and $\hat{r}_0$ is the initial vector of scattering point p in the reference coordinate system [6].

For multiple scattering points, the total echo signal is

$$s_r(t) = \sum_{i=0}^{N} a_i \exp\left\{j2\pi f_c\left[t - \frac{r_i(t)}{c}\right]\right\} \tag{2.3}$$

Assuming that the acoustic detection equipment emits a single-frequency continuous wave at a frequency of 600 kHz, the reference coordinate system is located $(U = 0, W = 400\sqrt{2}, V = 200)$, then $R_0 = \left(0, 400\sqrt{2}, 200\right)^{\mathrm{T}}$, and the initial Euler angle is $(\phi_e = 0°, \theta_e = 0°, \varphi_e = 0°)$, At the initial moment, the coordinates of the propeller blade in the target local coordinate system are $[x_0 = 0, y_0 = 0, z = (0 \sim 2)]$, the angular velocity is $\omega = (\omega_x = 8\pi, \omega_y = 0, \omega_z = 0)^{\mathrm{T}}$, and the target translational velocity is zero. The distance variation between the equivalent scattering point of the single blade and acoustic detection equipment is shown in Fig. 1, and the time–frequency distribution of the single-blade echo signal is shown in Fig. 2.

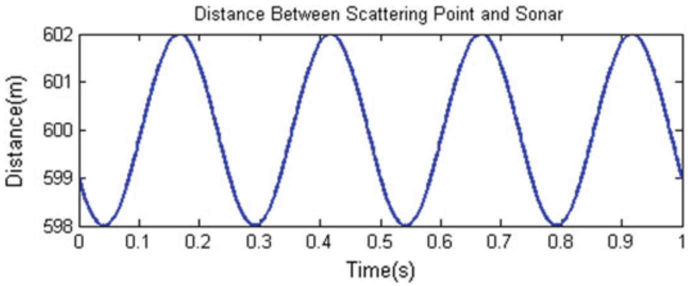


Fig. 1. Distance between scattering point and acoustic detection equipment

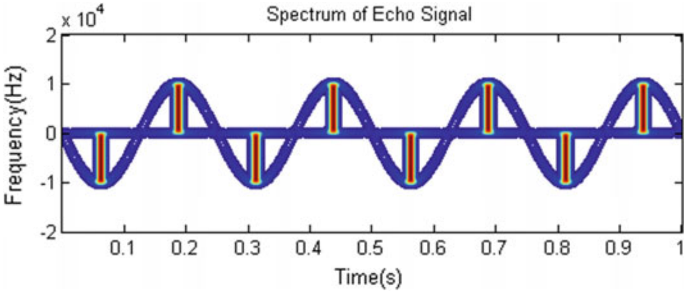


Fig. 2. Time–frequency distribution of echo signals

From Fig. 2, it can be seen that in the case of single propeller blade echoes without translation, the target distance changes periodically, and the spectrum of echo signal oscillates periodically near the zero fundamental frequency.

The propeller is a body target consisting of multiple scattering points. At the same time, the echo signal of target body is also obtained by the detection equipment along with the echo signal of propeller, and the target also has a translation. So the received echo signal can be expressed as

$$\begin{aligned} s_r(t) = s_B(t) + s_M(t) = & \sum_{n=1}^{N} a_{B_n}(t) \cos\left\{ j2\pi f_c \left[t - \frac{2r_{B_n}(t)}{c} \right] \right\} \\ & + \sum_{l=1}^{L} a_{M_l}(t) \cos\left\{ j2\pi f_c \left[t - \frac{2r_{M_l}(t)}{c} \right] \right\} \end{aligned} \tag{2.4}$$

where $s_B(t)$ is the echo of the target body, $s_M(t)$ is the echo of the rotating propeller blade, and N and L are the number of scattering points of the target body and the micro-moving part, respectively, $a_{Bn}(t)$ is the time-varying coefficient of the scattering intensity for the scattering point n of the target body, $a_{Ml}(t)$ is the time-varying coefficient of the scattering intensity for the scattering point l of the rotating component, correspondingly, $r_{Bn}(t)$ is the time-varying distance between the scattering point n of the target body and the detection equipment, and $r_{Ml}(t)$ represents the time-varying distance between the scattering point l of the rotating component and the detection equipment.

Taking the simulation of Figs. 1 and 2 as an example, the translation velocity is $= (0, 20, 0)$ and the intensity of target body echo is 100 times as the intensity of the single-blade echo. Then, the distance between the scattering point and the acoustic detection equipment is shown in Fig. 3. And the time–frequency distribution of the echo signal is shown in Fig. 4.

Just as shown in Fig. 3, due to the translation, the distance between the equivalent scattering point and the acoustic detection equipment is no longer periodic and stable, but the oscillating center is also changing with time. Figure 4 shows that the target body echo and propeller echo are superimposed. Since the target body echo is much stronger than the propeller echo, the echo signals that reflect the periodic changes of the

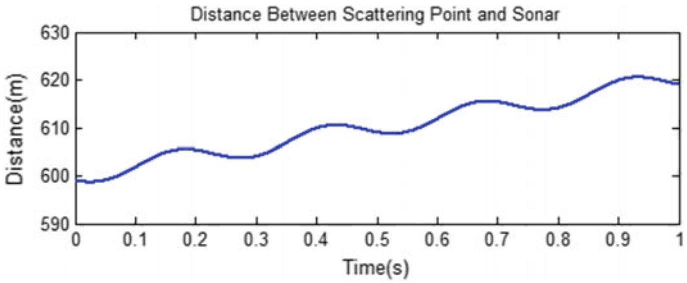


Fig. 3. Distance between scattering point and acoustic detection equipment

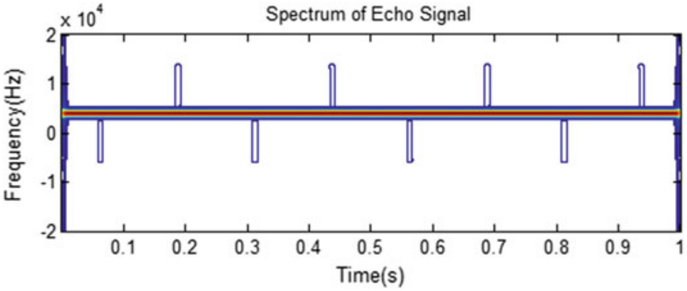


Fig. 4. Time–frequency distribution of echo signals

propeller are almost covered up. At the same time, the translation is also mixed with the rotation of the propeller, which causes base frequency shifting. These put great difficulties to the detection and identification of the rotating object.

3 Analysis of Interference Suppression

In the above, we assumed that the transmitted and received signals are all exponential complex signals, because under narrow-band conditions, the complex exponential signals are equivalent to complex analytical signals. However, for acoustic detection equipment, both the received and transmitted signals are real sine signals. Therefore, the signal should firstly be processed by quadrature reception to obtain the complex exponential expression of the signal.

Assume that the acoustic detection equipment emits a single-frequency continuous wave with a frequency of f_c, i.e.

$$s_t(t) = \cos(2\pi f_c t) \tag{3.1}$$

The echo signal of the scattering point is

$$s_r(t) = a\cos\left\{2\pi f_c\left[t - \frac{2r(t)}{c}\right]\right\} \tag{3.2}$$

The above signals are processed in a quadrature manner as shown in Fig. 5.

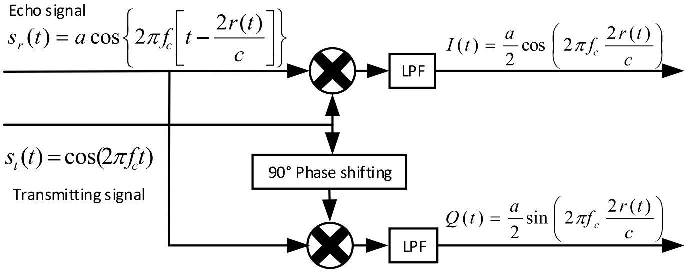

Echo signal
$s_r(t) = a\cos\left\{2\pi f_c\left[t - \frac{2r(t)}{c}\right]\right\}$
$s_t(t) = \cos(2\pi f_c t)$
Transmitting signal
90° Phase shifting
LPF
LPF
$I(t) = \frac{a}{2}\cos\left(2\pi f_c\frac{2r(t)}{c}\right)$
$Q(t) = \frac{a}{2}\sin\left(2\pi f_c\frac{2r(t)}{c}\right)$

Fig. 5. Quadrature processing

Combining I and Q channel output, the complex expression of the signal can be obtained [7], i.e.

$$s_b(t) = I(t) + jQ(t) = \frac{a}{2}\exp\left(j2\pi f_c\frac{2r(t)}{c}\right) \tag{3.3}$$

For echo signal of propellers with echo of the target body and translation as shown in Eq. (2.4), quadrature reception can bring the complex form of the signal out, that is

$$s_b(t) = \sum_{n=1}^{N}\frac{a_{B_n}(t)}{2}\exp\left(j2\pi f_c\frac{2r_{B_n}(t)}{c}\right) + \sum_{l=1}^{L}\frac{a_{M_l}(t)}{2}\exp\left(j2\pi f_c\frac{2r_{M_l}(t)}{c}\right) \tag{3.4}$$

The first part of Eq. (3.4), that is, the movement of the target body, is generally changing slowly. At the same time, the processing time is relatively short. In such a short period of time, all scattering points of the target body are considered as with the same translation velocity which is uniform. That is, $r_{B_n}(t) = r(t) = vt$. So the translation velocity of the target body's scattering point can be approximated by the speed estimate of the acoustic detection equipment and compensated accordingly.

Assuming that the object speed obtained by the acoustic detection equipment is $\frac{d[r(t)]}{dt}$, speed compensation of Eq. (3.4) yields

$$\hat{s}_b(t) = s_b(t)\exp\left(-j2\pi f_c\frac{2r(t)}{c}\right) = \sum_{n=1}^{N}\frac{\sigma_{B_n}(t)}{2} + \sum_{l=1}^{L}\frac{\sigma_{M_l}(t)}{2}\exp\left(j2\pi f_c\frac{2r_{M_l}(t) - 2vt}{c}\right) \tag{3.5}$$

From Eq. (2.2), we know that $r_{M_l}(t) = R_0 + vt + R_{\text{rotating}}\hat{r}_0$, bring it into Eq. (3.5) and ignore the DC component, we can get

$$\hat{s}_b(t) = \sum_{l=1}^{L} \frac{\sigma_{M_l}(t)}{2} \exp\left(j2\pi f_c \frac{2\left(R_0 + R_{\text{rotating}}\hat{r}_0\right)}{c}\right) \tag{3.6}$$

Obviously, Eq. (3.6) in which time–frequency distribution can be obtained by time–frequency analysis is the propeller echo signal which eliminates the echo of the target body and translation. The time–frequency curve only includes the periodically changing Micro-Doppler frequency, which is obviously caused by the rotating motion of the propeller.

4 Simulation

Assuming that the acoustic detection device emits a 100 kHz single-frequency continuous wave signal, the reference coordinate located at (300, 400, 0), the blade length l = 0.25 m, the rotation speed $r = 2r/s$, and at the initial time, the initial rotation angle $\varphi_0 = 0°$, $\beta = 0°$, the target translation speed is $v = (0, 2, 0)$, then the Micro-Doppler feature of single-blade echo signal is shown in Fig. 6.

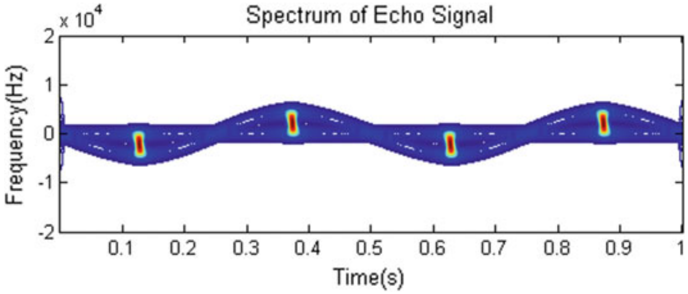


Fig. 6. Micro-Doppler characteristic of echo signal

It can be seen from the figure that the base frequency of the echo signal is located at zero frequency, and the periodic oscillation of the Micro-Doppler frequency is clearly visible. That is, the frequency spectrum transfer caused by the target translation and the echo of the target body is eliminated.

5 Conclusion

From the above analysis, it can be seen that in the process of Micro-Doppler signal acquisition, in addition to the echo of the moving parts, the scattering signal of the target body and the information of the target translation velocity will inevitably be introduced. This has put great difficulties in the extraction of Micro-Doppler features.

Theoretical analysis and simulation experiments show that the target body echo and translation velocity can be effectively eliminated by quadrature reception and translation compensation.

References

1. Zhang Q, Luo Y (2013) Micro-doppler effect of radar targets. National Defense Industry Press, Beijing, pp 22–31
2. Li X, Liu Y, Li K (2016) The micro-motion characteristic of radar target. Science Publishing House, Beijing, pp 1–8
3. Chen P, Hao S, Hu Y, Li Z (2015) Micro-doppler analysis of moving helicopter's rotor blades. TIANJIN Infrared Laser Eng (1):118–121
4. Chen VC et al (2006) Micro-doppler effect in radar: phenomen, model, and simulation study. IEEE Trans. Aerosp Electron Syst 42(1):2–21
5. Tang B, Huang W, Chen X (2019) Research on micro-doppler characteristics of underwater target. Wuhan J Naval Univ Eng 31(1):92–95
6. Yang Q, Zhang Q, Wang M (2015) Doppler feature analysis of ship based on airborne narrow band radar. Beijing Syst Eng Electron 37(12):2733–2738
7. Wu S (2013) The micro-doppler effect in radar. Publishiing House Of Electronics Industry, Beijing, pp 47–50

Weighted Least Square Support Vector Regression Method with GGP-Based Sequential Sampling

Yang Guo[1], Jiang Cao[1], Qi Ouyang[2](✉), and Shaochi Cheng[1]

[1] PLA Academy of Military Science, Beijing 10091, China
guoyangnudt@gmail.com
[2] Beijing Aerospace Control Center, Beijing 10094, China
oyqnudt@hotmail.com

Abstract. Approximation models are widely used in engineering reliability analysis due to the enormously expensive computation cost of limit state functions. In this paper, the weighted least-squared support vector regression (WLSSVR) method is used for model approximation. Sequential modeling is also considered to reduce the training sample number. A WLSSVR method with great gradient point (GGP)-based sequential sampling strategy is established and tested. The results show that the proposed method improves the global approximation accuracy.

Keywords: Support vector machine · WLSSVR · Sequential modeling · Greatest gradient point

1 Introduction

In order to reduce the computational burden in reliability analysis, approximation methods are always be applied to construct a simple and inexpensive approximate model as a replacement of the complex limit state function. Various types of approximation methods have been applied to reliability analysis, for instance, the response surface method (regression polynomial, RSM) [1–6], polynomial chaos expansion (PCE) [7–10], kriging [11–14], neural network (NN) [15–18] and support vector machines (SVM) [19]. As SVM and its improved variant least square support vector machine (LSSVM) method [19] are powerful tool that can provide good approximation accuracy and robustness [20], they have been widely studied in recent researches [21–30].

The performance of SVM/LSSVM is highly depended on the training samples. An effective way to choose the appropriate training samples is to use sequential modeling technology. In this paper, weighted least square support vector machines (WLSSVM) method [19] is used for model approximation. Section 2 gives a brief introduction of WLSSVR method. In Sect. 3, a great gradient point (GGP)-based sampling strategy is developed. This sampling strategy can improve the approximate ability of WLSSVR method for highly nonlinear limit state function. Section 4 presents the application examples of the proposed method.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 212–219, 2020.
https://doi.org/10.1007/978-981-15-0187-6_24

2 Weighted Least Square Support Vector Machines for Regression (WLSSVR)

If the training samples have different importance to construct the LSSVR model, a weighting function can be added to the LSSVR model to describe this difference. The mathematical formula of the weighted LSSVR, named as WLSSVR, is

$$\begin{aligned} \min_{w,b} \quad & J_p(w,e) = \frac{1}{2}\mathbf{w}\mathbf{w}^T + \frac{1}{2}\gamma\sum_{k=1}^{N} W_k e_k^2 \\ \text{s.t.} \quad & y_k = \mathbf{w}^T\varphi(\mathbf{x}_k) + b + e_k \end{aligned} \tag{1}$$

where W_k is the weight at $\mathbf{x}_k$, e_k is the error variable and γ is the penalty constant.

The WLSSVR model can be given by the following solution

$$\begin{bmatrix} \mathbf{0} & \mathbf{1}_v^T \\ \mathbf{1}_v & \boldsymbol{\Omega} + \mathbf{W}^{-1}/\gamma \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{y} \end{bmatrix} \tag{2}$$

where $\mathbf{W} = \mathrm{diag}(W_1, W_2, \ldots, W_N)$, $\mathbf{y} = [y_1; \ldots; y_N]$, $\mathbf{1}_v = [1; \ldots; 1]$, $\boldsymbol{\alpha} = [\alpha_1; \ldots; \alpha_N]$ and $\boldsymbol{\Omega}_{kj} = \langle \varphi(\mathbf{x}_k), \varphi(\mathbf{x}_j) \rangle = K(\mathbf{x}_k, \mathbf{x}_j)$.

3 Proposal Method

3.1 Sequential Sampling Strategies

Since the concerned probability of failure only relies on the sign of the limit state function, it is essential to sample the points with a high potential to cross the failure surface to the training sample set. The most straightforward way is to sample the points on the approximate failure surface to the training sample set, but it is unrealistic to obtain all the points on the approximate failure surface. For this reason, figure out some important points on the approximate failure surface and sample them to the training sample set is a better choice. Since the gradient information can help identify regions with a high degree of nonlinearity, sampling the point with greatest gradient to the training sample set can improve the global approximation quality [31]. The greatest gradient point is named as GGP for simplicity. In this section, GGP-based sequential sampling strategies are proposed.

In order to identify the nonlinear region of the failure surface, the relative derivative is used here. Without loss of generality, we take the last design variable $\mathbf{x}_n$ as the dependent variable, then the relative derivative can be calculated as

$$\frac{\partial \mathbf{x}_n}{\partial \mathbf{x}_i} = -\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_i} \bigg/ \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_n} \quad (i = 1, \ldots, n-1) \tag{3}$$

The great gradient point (GGP) is defined as the point maximizing the norm of the relative derivative vector (labeled as $\mathbf{x}'_n$). The GGP-based sequential sampling strategy can be summarized as,

$$\begin{aligned} \max_{\mathbf{x}} \quad & \left|\mathbf{x}'_n\right| \\ \text{s.t.} \quad & \widetilde{g}(\mathbf{x}) = 0 \\ & l_{\mathbf{x}} \geq \alpha\sqrt[d]{V/N} \\ & \mathbf{x} \in [\mathbf{x}_l, \mathbf{x}_u] \end{aligned} \tag{4}$$

The gradient information is easy to get as it is a by-product of the WLSSVR modeling process. The WLSSVR model with RBF kernel can be expressed as

$$\tilde{g}(\mathbf{x}) = \mathbf{w}^T\varphi(\mathbf{x}) + b = \sum_{i=1}^{N} \alpha_k \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_k\|^2}{2\sigma^2}\right) + b \tag{5}$$

where α_k is the Lagrange multipliers at the optimum. The gradient at point $\mathbf{x}$ is then easily obtained by differentiating Eq. (5),

$$\widetilde{g'}(\mathbf{x}) = -\frac{1}{\sigma^2}\sum_{i=1}^{N} \alpha_k \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}_k\|^2}{2\sigma^2}\right)(\mathbf{x}-\mathbf{x}_k) \tag{6}$$

3.2 Procedure

The main steps of the WLSSVR method with GGP-based sequential sampling are:

Step 1: Generate a Monte Carlo population S of $\mathbf{N_S}$ points in the design space. The N_S points are generated by the crude Monte Carlo according to the distribution of the stochastic design variables. This population keeps same during the whole process.

Step 2: Generate the initial training samples set $\mathrm{T}^{(0)}$. A quasi-random numbers generation algorithm named as Sobol algorithm is used to generate the training samples because the Sobol numbers exhibit comparatively high entropy [26]. The initial training samples are generated as follows:

$$x_{ij} = a_{ij} + (b_{ij} - a_{ij})s_{ij}, \quad i = 1, 2, \ldots, M;\ j = 1, 2, \ldots, d \tag{7}$$

where M is the number of training samples, d is the dimension of the design variable $\mathbf{x}$, s_{ij} is a Sobol number defined in the unit hyper-cube, a_{ij} and b_{ij} are the lower and upper bound of the design variable in the jth dimension, respectively.

Step 3: Construct the WLSSVR model. For the tth iteration, an approximate model $\widetilde{g}^{(\mathrm{t})}(\mathbf{x})$ based on $\mathrm{T}^{(0)}$ is constructed using WLSSVR method.

Step 4: Calculate the value of approximate model at test points. The value of $\tilde{g}^{(\mathrm{t})}(\mathbf{x})$ are calculated based on the Monte Carlo population S.

Step 5: Check the stopping condition. If the number of iterations exceeds the maximum number of iterations, or the global approximation accuracy of the approximation model meets the requirements, the calculation terminates; otherwise, go to Step 6. Whether the model meets the global approximation accuracy requirements is determined by the convergency of the relative change of error rate of test points. The convergence is defined as: the average relative change of error rate in five successive iterations is less than the threshold value. The relative change of error rate of the tth iteration is defined as follows:

$$\epsilon_{\mathrm{t}} = \frac{1}{2N_s}\sum_{i=1}^{N_s}|\mathrm{sign}(\tilde{g}_t(\mathbf{x}_i)) - \mathrm{sign}(\tilde{g}_{t-1}(\mathbf{x}_i))| \tag{9}$$

The average relative change of error rate is defined as:

$$\bar{\epsilon}^{(t)} = \begin{cases} \epsilon^{(t)}, & \text{if } t < 5 \\ \frac{1}{5}\sum\limits_{i=0}^{4}\epsilon^{(t-i)}, & \text{if } t \geq 5 \end{cases} \tag{10}$$

where ϵ_0 is a given value depending on the precision required.

Step 6: Find the GGP. Firstly, $\mathbf{x}_{\mathrm{GGP}}^{(t)}$ is obtained by solving Eq. (4). Then, $\mathbf{x}_{\mathrm{GGP}}^{(t)}$ are sampled into the training sample set $\mathrm{T}^{(t)}$ to construct a new sample point set $\mathrm{T}^{(t+1)}$. Finally, the value of $t + 1$ is assigned to t and go to Step 3.

4 Application Examples

In this section, two examples are used to test the WLSSVR method with great gradient point (GGP) based sequential sampling strategy.

(1) Example 1: A two-dimensional nonlinear function [26]

$$g(\mathbf{x}) = 3.8 + x_2 - \exp(x_1 - 1.7), \quad x_1, x_2 \in [-7, 7] \tag{11}$$

For Example 1, the initial number of training sample points is 40, and the termination condition is $\epsilon_0 = 2 \times 10^{-4}$.

(2) Example 2: [25]

$$g(\mathbf{x}) = \frac{1}{4}\left(\sin(x_1 - 3)(x_2 - 1)^2 + (x_1 - 1)x_4\right) - 3, \quad x_1, x_2, x_3, x_4 \in [0, 10] \tag{12}$$

For example 2, the initial number of training sample points is 70, and the termination condition is $\epsilon_0 = 2 \times 10^{-3}$.

The absolute error rate of the *t*th iteration is defined as:

$$\epsilon_a^{(t)} = \frac{1}{2N_s}\sum_{i=1}^{N_s}\left|\mathrm{sign}(\tilde{g}^{(t)}(\mathbf{x}_i)) - \mathrm{sign}(g(\mathbf{x}_i))\right| \tag{13}$$

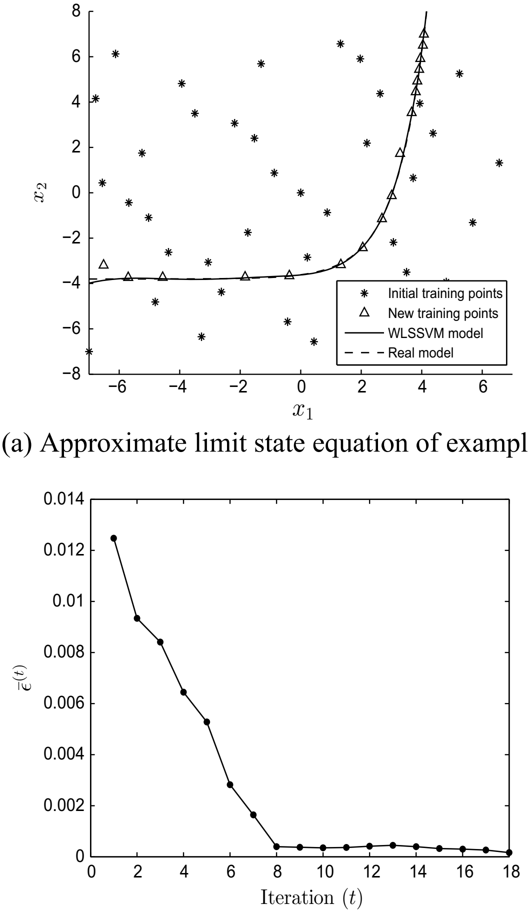
Average absolute error rate is defined as the average of absolute error rate in five consecutive iterations, similar to Eq. (10). The performance of these approximation methods is evaluated by comparing the absolute error rate.

The results of sequential modeling are shown in Table 1, in which $\epsilon_a^{(0)}$ is the initial absolute error rate and $\bar{\epsilon}_a^{(\mathrm{final})}$ is the average absolute error rate of the last iteration. In addition, in the experiment, the approximate model was also constructed by directly generating the same initial set of sample points as the number of sample points required for sequential modeling, and the absolute error rate $\epsilon_a^{\mathrm{direct}}$ is obtained. The results show that the proposed sequential modeling method reduces the absolute error rate of the approximate model. For the nonlinear limit state equation, the GGP sequential modeling method is effective, such as Example 1 and 2. As the complexity of the model increases, the number of iterations required for the convergence of the algorithm increases.

Table 1. Sequential modeling results of WLSSVR based on GGP sequential sampling strategy

Example	Initial number of training points	Final training points number	$\epsilon_a^{(0)}$	$\bar{\epsilon}_a^{(\mathrm{final})}$	$\epsilon_a^{\mathrm{direct}}$
Example 1	40	57	0.0125	0.0017	0.0063
Example 2	70	282	0.2019	0.0344	0.0417

The comparison between the WLSSVR approximate limit state equation and the real limit state equation of Example 1 is shown in Fig. 1. The results show that the approximate failure surface almost coincides with the real failure surface. The sequential approximation modeling algorithm based on GGP only adds a few sample points to achieve a good approximation effect. The GGP points are distributed on both sides of the real limit state equation, and there are relatively more training samples at the higher gradient.



(a) Approximate limit state equation of example 1

(b) The convergence process of the example 1

Fig. 1. WLSSVR sequential modeling results of Example 1 based on GGP

5 Conclusions

A WLSSVR method with GGP-based sequential sampling was developed to overcome the difficulty in approximating the nonlinear region of the failure surface in this paper. This method provides an efficient way for calculating the probability of failure using only a small number of training samples. The results show that the global approximate quality of application example is improved.

Acknowledgements. This work was supported by National Natural Science Foundation of China under Grant No. 61701503.

References

1. Bucher CG, Bourgund U (1990) A fast and efficient response surface approach for structural reliability problems. Struct Saf 7:57–66
2. Rajashekhar MR, Ellingwood BR (1993) A new look at the response surface approach for reliability analysis. Struct Saf 12:205–220
3. Kim S, Na S (1997) Response surface method using vector projected sampling points. Struct Saf 19:3–19
4. Kaymaz I, McMahon CA (2005) A response surface method based on weighted regression for structural reliability analysis. Probab Eng Mech 20:11–17
5. Lee SH, Kwak BM (2006) Response surface augmented moment method for efficient reliability analysis. Struct Saf 28:261–272
6. Allaix DL, Carbone VI (2011) An improvement of the response surface method. Struct Saf 33:165–172
7. Acharjee S, Zabaras N (2007) A non-intrusive stochastic Galerkin approach for modeling uncertainty propagation in deformation processes. Comput Struct 85:244–254
8. Paffrath M, Wever U (2007) Adapted polynomial chaos expansion for failure detection. J Comput Phys 226:263–281
9. Wei DL, Cui ZS, Chen J (2008) Uncertainty quantification using polynomial chaos expansion with points of monomial cubature rules. Comput Struct 86:2102–2108
10. Blatman G, Sudret B (2011) Adaptive sparse polynomial chaos expansion based on least angle regression. J Comput Phys 230:2345–2367
11. Scheremans L, Van GD (2005) Use of Kriging as meta-model in simulation procedures for structural reliability. In: 9th International conference on structural safety and reliability
12. Kaymaz I (2005) Application of Kriging method to structural reliability problems. Struct Saf 27:133–151
13. Hyeon JB, Chai Lee B (2008) Reliability-based design optimization using a moment method and a Kriging metamodel. Eng Optim 40:421–438
14. Echard B, Gayton N, Lemaire M (2011) AK-MCS: an active learning reliability method combining Kriging and Monte Carlo Simulation. Struct Saf 33:145–154
15. Hurtado JE, Alvarez DA (2001) Neural-network-based reliability analysis: a comparative study. Comput Methods Appl Mech Eng 191:113–132
16. Deng J, Gu D, Li X, Zhong Q (2005) Structural reliability analysis for implicit performance functions using artificial neural network. Struct Saf 27:25–48
17. Gomes HM, Awruch AM (2004) Comparison of response surface and neural network with other methods for structural reliability analysis. Struct Saf 26:49–67
18. Papadrakakis M, Lagaros ND (2002) Reliability-based structural optimization using neural networks and Monte Carlo simulation. Comput Methods Appl Mech Eng 191:3491–3507
19. Suykens JAK, Van Gestel T, De Brabanter J, De Moor B, Vandewalle J (2002) Least squares support vector machines. World Scientific Publishing Co. Pre.Ltd, London
20. Zhu P, Zhang Y, Chen G (2011) Metamodeling development for reliability-based design optimization of automotive body structure. Comput Ind 62:729–741
21. Rocco CM, Moreno JA (2002) Fast Monte Carlo reliability evaluation using support vector machine. Reliab Eng Syst Saf 76:237–243
22. Hurtado JE (2007) Filtered importance sampling with support vector margin: a powerful method for structural reliability analysis. Struct Saf 29:2–15
23. Chen KY (2007) Forecasting systems reliability based on support vector regression with genetic algorithms. Reliab Eng Syst Saf 92:423–432

24. Basudhar A, Missoum S, Harrison Sanchez A (2008) Limit state function identification using support vector machines for discontinuous responses and disjoint failure domains. Probab Eng Mech 23:1–11
25. Basudhar A, Missoum S (2008) Adaptive explicit decision functions for probabilistic design and optimization using support vector machines. Comput Struct 86:1904–1917
26. Hurtado JE, Alvarez DA (2010) An optimization method for learning statistical classifiers in structural reliability. Probab Eng Mech 25:26–34
27. Bourinet JM, Deheeger F, Lemaire M (2011) Assessing small failure probabilities by combined subset simulation and support vector machines. Struct Saf 33:343–353
28. Tan X, Bi W, Hou X, Wang W (2011) Reliability analysis using radial basis function networks and support vector machines. Comput Geotech 38:178–186
29. Moura MDC, Zio E, Lins ID, Droguett E (2011) Failure and reliability prediction by support vector machines regression of time series data. Reliab Eng Syst Saf 96:1527–1534
30. Dai H, Zhang H, Wang W (2012) A support vector density-based importance sampling for reliability assessment. Reliab Eng Syst Saf 106:86–93
31. Hwang JP, Park S, Kim E (2011) A new weighted approach to imbalanced data classification problem via support vector machine with quadratic cost function. Expert Syst Appl 38:8580–8585

Data Stream Adaptive Partitioning of Sliding Window Based on Gaussian Restricted Boltzmann Machine

Wei Wang(✉) and Mengjun Zhang

Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin 300387, China
weiwang@tjnu.edu.cn

Abstract. In this paper, the adaptive partitioning problem of data stream under sliding window is discussed. Gaussian restricted Boltzmann machine (GRBM) model supporting decimal input is proposed, which can be trained through iteration for data reconstruction subsequently. At the same time, a data stream adaptive block algorithm based on Kullback–Leibler divergence (KL distance) is proposed to compare the probability distribution difference in the sliding window. Then, obtain the predicted value by the distribution of the previous data and determine whether the KL distance is within the confidence interval, so as to realize the adaptive adjustment of the sliding window, and the divided of data stream.

Keywords: Data stream · Gaussian–Bernoulli restricted Boltzmann machine model · Kullback–Leibler divergence

1 Introduction

Data stream classification is a core problem in data mining due to its characteristics of continuous, real-time arrival, large amount, unrestricted, and unpredictable. The sliding window was adopted to ensure that the current data are the latest valid data in the data stream. However, the choice of window size will seriously affect the results of the experiment, for that too small window selection will lead to incomplete information collection while too large window selection will affect the efficiency of learning. So, how to choose the length of the window is a problem worth studying owing to appropriate length that can effectively improve the possibility and efficiency of the algorithm, reduce energy consumption, and can obviously save memory. Sliding window is used by Bifet in [1] which size is not fixed but recalculated online according to the rate of change observed from the data of the window itself. Judgement mechanisms of false positives and false negatives were used while it is inefficient in time and memory. The sliding window can also be incorporated with different prediction mechanisms such as [2], making them suitable for online prediction by adapting the number of the samples.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 220–228, 2020.
https://doi.org/10.1007/978-981-15-0187-6_25

In this article, Gaussian–Bernoulli restricted Boltzmann machine model (GRBM) and Kullback–Leibler divergence are presented for adaptive window adjustment. GRBM is an extended model based on RBM, which is a recursive neural network proposed by Salakhutdinov, and can solve some difficult tasks through the intrinsic characteristics of data [3]. By comparing the probability distribution of reconstructed data from a trained GRBM and input data, it can judge whether the model trained with the previous data block is applicable to the current data block; furthermore, it is judged whether there is a significant difference between the adjacent two data blocks, that is, whether the sliding window needs to be adjusted. Kullback–Leibler divergence can be introduced to judge whether the probability distribution is consistent or not. In addition, the concept of confidence interval is used as a criterion of judgment, the adjustment instruction will issue to the window when the KL distance value is greater than the confidence upper limit.

This paper is mainly divided into four parts. In Sect. 1, the relevant research background and the research content of this paper are introduced briefly. In Sect. 2, the basic model and a sliding window adjustment algorithm based on KL distance are proposed. Section 3 carries out relevant experiments and analyzes the results. Section 4 summarizes the entire article and establishes the future research direction.

2 Data Flow Blocking in Adaptive Window

A trained GRBM model, which can reconstruct the test data, is used as a benchmark to judge whether there is an obvious change between the adjacent data block. The Kullback–Leibler divergence and the confidence interval are introduced to measure this change. By comparing if the probability distribution difference between test data and reconstructed data is within the confidence interval, generate adjustment instructions of the sliding window.

2.1 Gaussian Restricted Boltzmann Machines

The network structure of GRBM is shown in Fig. 1.

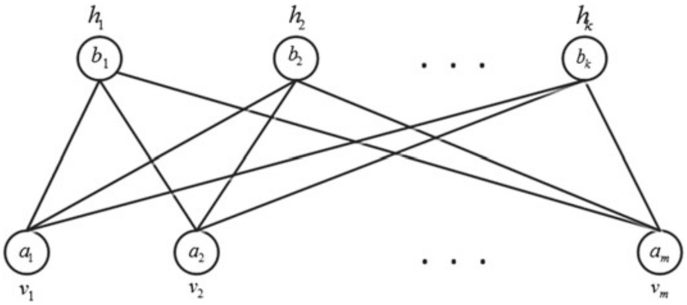


Fig. 1. Structure of GRBM

The energy function of GRBM is as follows [4]

$$E(v,h|\theta) = \sum_{i=1}^{m} \frac{(v_i - a_i)}{2\sigma_i^2} - \sum_{j=1}^{k} b_j h_j - \sum_{i=1}^{m} \sum_{j=1}^{k} \frac{v_i}{\sigma_i} W_{ij} h_j \tag{1}$$

where σ_i is the Gaussian noise standard deviation corresponding to the visible node v_i; $\theta = \{W, a, b\}$ is the parameter of the GRBM network parameter; W_{ij} represents the connection weight between the visible node v_i and the hidden layer node h_j; a_i, b_j are biases of the visible layer and the hidden layer. The joint probability can be obtained when the states of a group of visible layer nodes and hidden layer nodes (v, h) are known [5, 6]

$$p(v,h|\theta) = \frac{\mathrm{e}^{-E(v,h|\theta)}}{Z(\theta)} \tag{2}$$

where $Z(\theta)$ is the normalization factor and expressed by the following formula

$$Z(\theta) = \sum_{v,h} \mathrm{e}^{-E(v,h|\theta)} \tag{3}$$

In the GRBM model, when the state of the nodes in the visible layer is known, the activation conditions of the nodes in the hidden layer are independent. On the contrary, the activation state of each node in the visible layer is also conditionally independent when the state of each hidden layer node is known. The formulas are as follows [6]

$$P(h_j = 1|v,\theta) = \mathrm{sigmoid}\left(b_j + \sum_{i=1}^{m} \frac{v_i}{\sigma_i} W_{ij} \right) \tag{4}$$

$$P(v_i|h,\theta) = N\left(a_i + \sigma_i \sum_{j=1}^{k} W_{ij} h_j, \sigma^2 \right) \tag{5}$$

$$\mathrm{sigmoid}(x) = \frac{1}{1 + \exp(-x)} \tag{6}$$

2.2 Change Monitoring Under Sliding Window

KL distance is a method for describing the difference between two probability distributions. For two probability distributions X and Y, $X = \{x_1, x_2, \ldots, x_N\}$, $Y = \{y_1, y_2, \ldots, y_N\}$, and their similarity can be measured by KL distance [7].

$$h(X,Y) = \sum_{n=1}^{N} x_n \log \frac{x_n}{y_n} \tag{7}$$

The divergence satisfies three properties, hereafter referred to as the divergence properties:

1. Self-similarity: $h(X, X)$ = 0.
2. Self-identification: $h(X, Y)$ = 0 only if $X = Y$.
3. Positively: $h(X, Y) \geq 0$ for all f, g.

In this paper, the right value of the confidence interval is introduced as the threshold value. Confidence interval is an interval centered on the estimated value, which is used to determine the possible range of the true value according to the estimated value. Generally, [a, b] is used to represent the interval of the error range of the sample which could estimate the total mean value, where the specific values of a and b depend on the credibility of the result that "the area contains the total mean." The value of confidence interval lies in its ability to quantify the uncertainty of estimation, which provides a lower and upper limit and a possibility. As a separate radius measurement, the confidence interval is often referred to as the likelihood and represents estimated uncertainty by using error maps. In general, the larger the sample is estimated, the more accurate the estimate and the smaller the confidence interval.

For population $X \sim N(\mu, \sigma^2)$, assume that $X_1, X_2, \ldots, X_n$ is a sample from X, for $U = \frac{\overline{X}-\mu}{\sigma/\sqrt{n}} \sim N(0, 1)$, the confidence interval of confidence degree $1 - \alpha$ is $\left(\overline{X} - \frac{\sigma}{\sqrt{n}} u_{\alpha/2}, \overline{X} + \frac{\sigma}{\sqrt{n}} u_{\alpha/2}\right)$. In general, the larger the sample is estimated, the more accurate the estimate and the smaller the confidence interval.

2.3 Algorithm Implementation

The first thing to do to achieve the purpose of the paper is to train the restricted Boltzmann machine. The goal of training is to obtain the maximum likelihood of the input samples, so that the Gibbs distribution represented by the GRBM network is closest to the distribution represented by the sample itself. In order to obtain maximum likelihood, it is necessary to derive the parameters and maximize the logarithmic likelihood function step by step with the gradient rise method until the stopping condition is reached. The following likelihood functions should be maximized

$$L_{\theta,S} = \prod_{i=1}^{ns} P(v^i) \tag{8}$$

$$\ln L_{\theta,S} = \ln \prod_{i=1}^{ns} P(v^i) = \sum_{i=1}^{ns} \ln P(v^i) \tag{9}$$

The gradient of each parameter is calculated as follows [8, 9].

$$
\begin{aligned}
\Delta w_{ij} &= \varepsilon((v_i h_j)_{\text{data}} - (v_i h_j)_{\text{recon}}) \\
\Delta a_i &= \varepsilon((v_i)_{\text{data}} - (v_i)_{\text{recon}}) \\
\Delta b_j &= \varepsilon((h_j)_{\text{data}} - (h_j)_{\text{recon}})
\end{aligned}
\tag{10}
$$

where $()_{\text{data}}$ denotes parameters related to input data in the visible layer, $()_{\text{recon}}$ represents parameters related to reconstructed data, and ε is learning rate.

```
Algorithm1: training of GRBM
Input: training data, learning rate $\varepsilon$, number of visible $m$, number of hidden $n$,
       epochs
Initialize $\theta=\{W,a,b\}$ with 0.
for epochs=1,2,3…
    for $j = 1, 2, \ \ , k$, then
        calculate hidden layers status with formula (4)
    end for
    for $i = 1, 2, \ \ , m$
        calculate visible layers status with formula (5)
    end for
    updating $\theta$ with formula (10)
end for
```

Based on the trained GRBM model, the adjustment of the window can be judged by measuring the KL distance, as shown in Algorithm 2. In this article, the KL distance method can be used to judge whether an abnormality has occurred through determining the degree of difference between the input data and the reconstructed data. The smaller the absolute KL distance of X to Y, the closer the two distributions X and Y are, vice versa. Specifically, distribution between reconstructed and input data is identical when the KL distance of them equal to zero, and the data flow does not change at this time. When the difference between reconstructed data and input data increases, the KL distance between them increases gradually. When the KL distance is larger than the set threshold, it proves that the previously trained RBM model is no longer applicable to the current data stream, that is, the concept drift occurs. Confidence interval is introduced to provide a boundary for judging KL distance. Depending on the distribution of the obtained KL distance, this article assuming that the population sample obeys the standard normal distribution and chooses the confidence level of 95%.

```
Algorithm2: Window adaptive adjustment process
Input: testing data, batch $M$
Output: KL distance
Local Variables: $M_p$, $M_q$, $X$, $Y$
Initialize GRBM with Algorithm 1.
Initialize $1\text{-}\alpha = 95\%$
for $i = 1, 2, \ldots, 100$ then
    $M_p \leftarrow$ testing data
    $M_q \leftarrow$ reconstruct data of GRBM
    $X \leftarrow$ probability distribution of $M_p$
    $Y \leftarrow$ probability distribution of $M_q$
    KL distance $\leftarrow h(X, Y)$
    if KL distance > $z_{1-\alpha}$ then
        window adjustment instructions
    end if
end for
```

3 Experiment Results

The data set used in this experiment is radar data for detecting the recognition of human body passing through a wall, which fluctuates widely and needs pre-training. The GRBM mentioned in this article uses the sigmoid function which takes values between 0 and 1, so it is necessary to normalize radar data in advance.

In the experiment, the initial size of the sliding window is set to 100, and the data set used is the human body wall identification data. Select 10,000 unmanned radar data to train the model and select the same number of data for the experiment. In the experiment, 100 adjacent data blocks were analyzed. Radar data in the manned state behind the wall were read between the 34th and 64th data blocks, and the rest of the data are the measurement data in the unmanned state. The parameters of GRBM in the experiment are shown in Table 1.

The following figure is the data distribution map between data blocks 32–38 and 60–66. Obviously, the change of data stream cannot be observed directly in time-domain graph (Figs. 2 and 3).

Using the algorithm mentioned in this paper can get the following experimental results as shown in Fig. 4, where the straight line represents the upper limit of confidence, and the measured KL distance value is represented by the scatter point. Observing the experimental results, it can be seen that between the data blocks [34, 64], the KL distance exceeds the upper confidence limit, which represents the difference in probability distribution between input data and reconstructed data. The window needs

Table 1. Parameters of GRBM

Number of visible	100
Number of hidden	10
Epochs	1000
Learning_rate	0.1
Weight_decay	1
cd_steps	1
Momentum	0.5

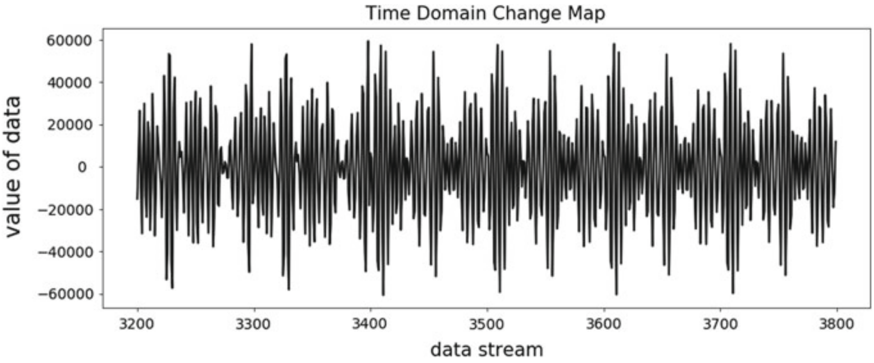


Fig. 2. Data graph at change of data stream (between data blocks 32 and 38, where each data block contains 100 data points)

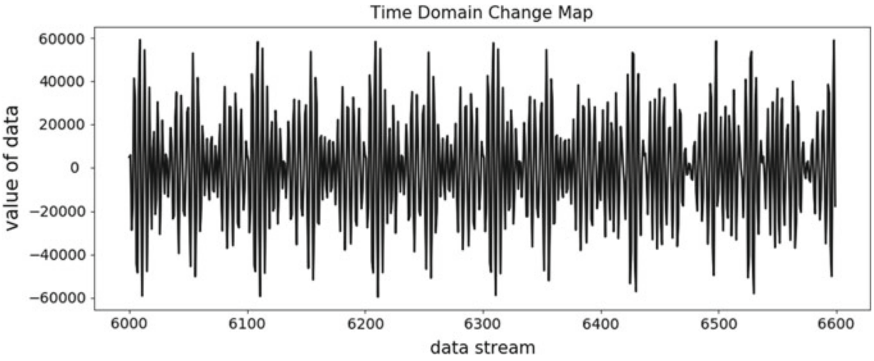


Fig. 3. Data graph at change of data stream (between data blocks 60 and 66, where each data block contains 100 data points)

to be adjusted when the KL distance exceeds the confidence limit for that the difference between training and the current data block model is large. The KL distance of other data blocks is close to 0, that is, there is no change between data blocks.

Therefore, the proposed algorithm is of great significance for anomaly detection and adaptive data flow partitioning.

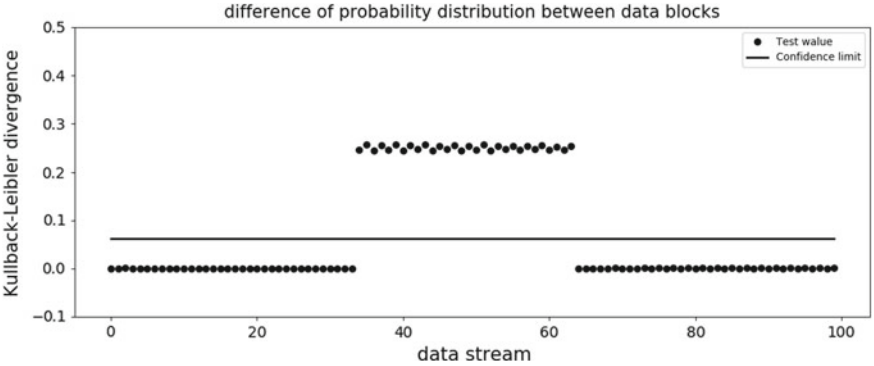


Fig. 4. KL distance measured by experiments

4 Conclusion

Aiming at reconstruction characteristics of the restricted Boltzmann machine, this paper proposes a window adaptive adjustment method based on KL distance. The KL distance algorithm is used to compare the difference of probability distribution between input data and reconstructed data, and then to judge whether there is concept drift between adjacent data blocks and realize adaptive adjustment of sliding window. However, this paper only judges the data of a single source, and on this basis, the next step should be to study the adaptive partitioning of multi-source heterogeneous data.

Acknowledgements. This paper is supported by Natural Youth Science Foundation of China (61501326), the National Natural Science Foundation of China (61731006).

References

1. Bifet A, Gavalda R (2007) Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining, pp 443–448
2. Dalmazo BL, Vilela JP, Curado M (2014) Online traffic prediction in the cloud: a dynamic window approach. In: 2014 International conference on future internet of things and cloud. IEEE, pp 9–14
3. Salakhutdinov R, Hinton G (2009) Deep boltzmann machines. In: Artificial intelligence and statistics, pp 448–455
4. Larochelle H, Mandel M, Pascanu R et al (2012) Learning algorithms for the classification restricted boltzmann machine. J Mach Learn Res 13(Mar):643–669
5. Hinton GE (2002) Training products of experts by minimizing contrastive divergence. Neural Comput 14(8):1771–1800
6. Hinton GE (2012) A practical guide to training restricted Boltzmann machines. In: Neural networks: tricks of the trade, pp 599–619
7. Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22(1):79–86

8. Tieleman T (2008) Training restricted Boltzmann machines using approximations to the likelihood gradient. In: Proceedings of the 25th international conference on machine learning, pp 1064–1071
9. Milias-Argeitis A, Lygeros J (2011) Efficient stochastic simulation of metastable Markov chains. In: 2011 50th IEEE conference on decision and control and European control conference. IEEE, pp 2239–2244

Intrusion Detection Based on Convolutional Neural Network in Complex Network Environment

Yunfeng Zhou[1], Xuezhang Zhu[1], Su Hu[1(✉)], Di Lin[1], and Yuan Gao[1,2]

[1] University of Electronic Science and Technology of China, Sichuan 611731, China
Husu@uestc.edu.cn

[2] Academy of Military Science of PLA, Beijing 100090, China

Abstract. In this paper, we propose an intrusion detection method based on a convolutional neural network (CNN) for intrusion detection systems. In designing a deep intrusion detection model, mainstream deep learning techniques such as dropout, Adam, and Softmax classifier are used, respectively. Firstly, plain text data is dimensionally corrected and converted into grayscale images. Secondly, the CNN model obtains the feature map by learning features. Finally, the feature map is input to the Softmax classifier to obtain the detection results. The method is implemented on TensorFlow and tested on KDDCUP'99 data set. The results show that the model proposed can obtain high detection accuracy rapidly and satisfy the real-time detection requirements of complex network systems.

Keywords: Intrusion detection · Convolutional neural network · Deep learning · Complex network

1 Introduction

In recent years, due to the rapid development of the Internet and its wide application in various fields of society, the problem of network security has become increasingly serious. Intrusion detection systems play a critical role in the maintenance of network security systems because it can generate an alarm in the event of an attack when monitoring network traffic [1–4]. The essence of intrusion detection can be summarized as the classification of normal data and attack data [5–7], and machine learning algorithm is one of the most effective algorithms in classification problems [8].

Classification algorithms in machine learning include support vector machines (SVM), K-means clustering, decision tree, etc. SVM is sensitive to parameter setting when dealing with large sample data. K-means clustering is an unsupervised algorithm, so the known labels in the sample cannot be used.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 229–238, 2020.
https://doi.org/10.1007/978-981-15-0187-6_26

Decision tree is prone to overfitting problems and has poor performance in the face of complex continuous features. These traditional machine learning methods mostly have weak generalization ability and limited ability to express complex functions, so they cannot deal with complex classification problems well.

In recent years, deep learning method has been widely used in face recognition, speech recognition, image recognition, and other fields, which make it become a hot topic in machine learning. At the same time, deep learning also has a good application in network security intrusion detection. By comparing the intrusion detection technology based on deep learning with various traditional intrusion detection technologies, it is found that the intrusion detection technology based on deep learning achieved better results in terms of accuracy and false positive rate.

In this paper, an intrusion detection algorithm based on TensorFlow is proposed by using CNN, combining with mainstream deep learning techniques such as dropout, Adam, and Softmax classifier. And KDDCUP'99 data set is used to verify the feasibility of the proposed algorithm. This paper compares different activation functions used in the proposed CNN model and then designs a five-category test experiment to test the performance of the proposed method for intrusion detection.

The paper is organized as follows: Sect. 2 presents an intrusion detection algorithm based on deep convolutional neural network. The experimental results and analysis are discussed in Sect. 3. Finally, the paper is concluded with a discussion of future work in Sect. 4.

2 Implementation of the Detection

Deep learning can automatically learn useful features from raw data to improve classification accuracy. The basic architecture of machine learning is the combination of feature extraction and classification modules. The proposed architecture of deep learning intrusion detection method based on CNN is shown in Fig. 1. It mainly consists of three modules: data preprocessing module, CNN modeling module, and classifier module.

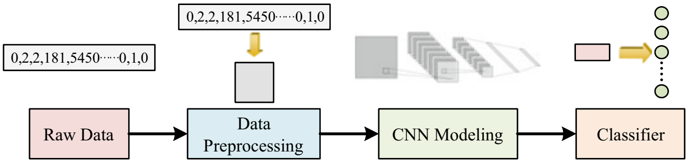


Fig. 1. Intrusion detection architecture

2.1 Data Preprocessing

KDDCUP'99 data set is a standard data set used for the research on intrusion detection systems. Each record in KDDCUP'99 data set consists of 41 features and a label representing the type.

Due to non-numerical features and large differences in the dimensions of KDDCUP'99 data set, it is difficult to directly apply the CNN algorithm for intrusion detection classification. Therefore, as shown in Fig. 2, the data set needs to be preprocessed, and the main data preprocessing includes numerical processing, dimensionality reduction, and one-hot encoding.

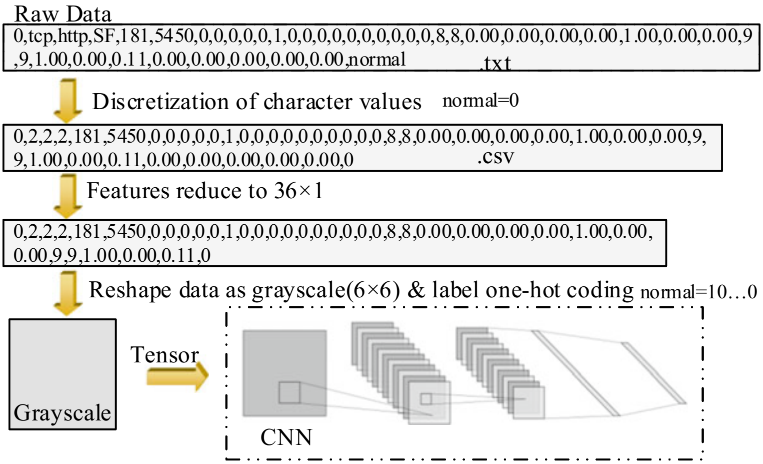


Fig. 2. Example of data processing for KDDCUP'99 data set

Numerical Processing Features 2 (protocol_type), 3 (service), 4 (flag) and the label of each connection in KDDCUP'99 data set are not numeric types, which need to be converted to numeric types to unify the format of data. In this paper, the values of feature 2 are represented by $[0,2]$; the values of feature 3 are represented by $[0,69]$; the values of feature 4 are represented by $[0,10]$; the values of the label are represented by $[0,22]$.

Dimensionality Reduction Feature 1 (duration) represents the duration of the network connection. Feature 7 (land) is 1 when the connection comes from/delivers to the same host/port, otherwise it is 0. Feature 9 (urgent) indicates the number of expedited packages. Feature 21 (is_hot_login) is 1 when the login belongs to the 'host' list, otherwise, it is 0. It can be found that features 1, 7, 9, and 21 are almost 0 in KDDCUP'99 data set. Feature 20 (num_outbound_cmds), indicating the number of outbound connections in an

FTP session, is all 0 in KDDCUP'99 data set. Therefore, the features mentioned above can be eliminated.

In addition, the original intrusion data is usually one-dimensional vector data, but CNN is generally used to process two-dimensional image data. It is necessary to convert the original one-dimensional data into two-dimensional data. After eliminating features 1, 7, 9, 20, and 21 during dimensionality reduction, the remaining 36 features are reshaped into a 6 × 6 form in this paper.

One-Hot Encoding In the numerical processing mentioned above, the values of the label are represented by $[0, 22]$. To make the data sparse, this paper uses one-hot encoding to convert the integers from $[0, 22]$ into a 23-bit binary vector, in which all the digits are 0 except that the index bit corresponding to the integer is 1.

2.2 CNN Modeling

The basic structure of CNN consists of an input layer, a convolutional layer, a pooling layer, a fully connected layer, and an output layer. Usually, in CNN structure, the deeper the network depth is, the larger the number of feature maps is, the greater the feature space that the network can represent and the stronger the network learning ability will be, while which will also make the calculation of the network more complex and lead to overfitting problems. Therefore, in practical applications, the network depth, the number of feature faces, the size of the convolution kernel, and the sliding step of convolution should be selected appropriately, so that a good model can be obtained during training, and the training time can be reduced.

This paper uses a simplified CNN model, as shown in Fig. 3, which mainly consists of a convolutional layer, a max pooling layer, two fully connected layers, and a Softmax classifier.

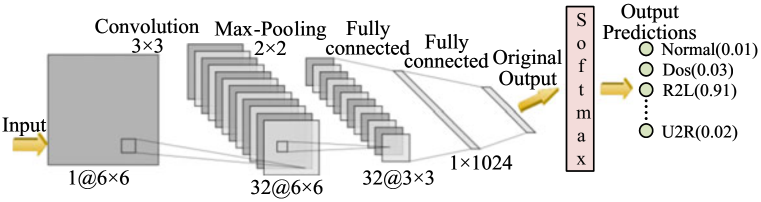


Fig. 3. Basic structure of the convolutional network

Dropout Learning Dropout is used to prevent parameters from over-reliance on training data effectively and improve the ability of parameters to generalize data sets. During model training, some hidden nodes are actively and temporarily discarded with a probability p, thus, the network size is greatly reduced. CNN is allowed to learn features in these incomplete networks, so as to ensure better generalization ability and reduce the possibility of overfitting effectively.

In the prediction, the results of all sub-models are averaged to increase the capacity and generalization of the model. The authors in [9,10] insist that dropout works best and generates the most abundant network structure when $p = 0.5$.

Activation Functions For CNN that evolved from traditional neural networks, different activation functions will lead to different expression abilities of CNN, especially between multi-layer connections. At present, nonlinear functions are generally used as activation functions due to that nonlinear traits can be used to improve the network's limited approximation ability effectively. Common activation functions are sigmoid, tanh, rectified linear unit (ReLU), and the improved activation functions based on ReLU, such as Leaky-ReLU, P-ReLU, and ReLU6. The comparison between different activation functions is shown in Table 1.

Table 1. Comparison between different activation functions

Activation function	Characteristics
sigmoid	(1) Gradient disappearance problem
	(2) The output is not zero-centered
	(3) Sigmoid contains power operation in the analytic formula, which is time-consuming to solve
tanh	(1) The output is zero-centered
	(2) Gradient disappearance problem still exist
ReLU	(1) Gradient disappearance problem is solved in the positive range
	(2) ReLU converges much faster than sigmoid and tanh
	(3) The output is not zero-centered
	(4) Some neurons may never be activated, resulting in the corresponding parameters that can never be updated

Optimization Algorithms The optimization algorithms for deep learning mainly include GD, SGD, Momentum, RMSProp, and Adam algorithms. Among them, Momentum is an optimization algorithm based on the gradient-based moving index weighted average. RMSProp calculates the differential squared weighted average for the gradient, which is beneficial to correct the swing amplitude and speed up the convergence. The Adam algorithm combines the Momentum algorithm and the RMSProp algorithm, thus, it has the advantages of both algorithms.

At the beginning of the training, gradient cumulates ($v_{d\omega}$ and v_{db}) and squared cumulates ($s_{d\omega}$ and s_{db}) are initialized as:

$$v_{d\omega} = v_{db} = s_{d\omega} = s_{db} = 0. \tag{1}$$

Assume that in the tth training, the parameter updates for Momentum and RMSProp are calculated by:

$$\begin{aligned} v_{d\omega} &= \beta_1 v_{d\omega} + (1-\beta_1)\, dW, \quad v_{db} = \beta_1 v_{db} + (1-\beta_1)\, db, \\ s_{d\omega} &= \beta_2 s_{d\omega} + (1-\beta_2)\, dW^2, \quad s_{db} = \beta_2 s_{db} + (1-\beta_2)\, db^2. \end{aligned} \tag{2}$$

In these equations, β_1 and β_2 are gradient cumulative indexes, which are usually 0.9 and 0.999, respectively; dW and db are the gradients of the weight (W) and bias (b). Since the exponential moving average will cause a large difference from the initial value in early iteration, gradient cumulates, and squared cumulates need to be corrected by:

$$v_{d\omega}^c = \frac{v_{d\omega}}{1-\beta_1^t}, \quad v_{db}^c = \frac{v_{db}}{1-\beta_1^t}, \quad s_{d\omega}^c = \frac{s_{d\omega}}{1-\beta_2^t}, \quad s_{db}^c = \frac{s_{db}}{1-\beta_2^t}. \tag{3}$$

In these equations, $v_{d\omega}^c$, v_{db}^c, $s_{d\omega}^c$ and s_{db}^c are the corrections of $v_{d\omega}$, v_{db}, $s_{d\omega}$ and s_{db} respectively. W and b can be updated by:

$$W^{(n)} = W^{(n-1)} - \alpha \frac{v_{d\omega}^c}{\sqrt{s_{d\omega}^c} + \varepsilon}, \quad b^{(n)} = b^{(n-1)} - \alpha \frac{v_{db}^c}{\sqrt{s_{db}^c} + \varepsilon}. \tag{4}$$

In these equations, ε is the smooth term, which is usually 10^{-8}; α is the learning rate, which needs to be fine-tuned during training.

2.3 Classifier

In this paper, Softmax classifier, one of the most commonly used classifier in CNN, is used as the classification module. Softmax classifier maps the output of multiple neurons to the interval of $(0, 1)$, which can be understood as a probability problem, given as follows:

$$p\left(y^{(i)} = j | x^{(i)}; \theta\right) = \frac{e^{\theta_j} x^{(i)}}{\sum_{l=1}^{k} e^{\theta_l} x^{(i)}}. \tag{5}$$

In this equation, θ_j is the j^{th} weight vector; $x^{(i)}$ is the ith data sample; $y^{(i)}$ is the output of the ith neuron; k is the total number of possible cases of the classification. In this paper, cross entry is used as the loss function.

Table 2. Details of KDDCUP'99 data set

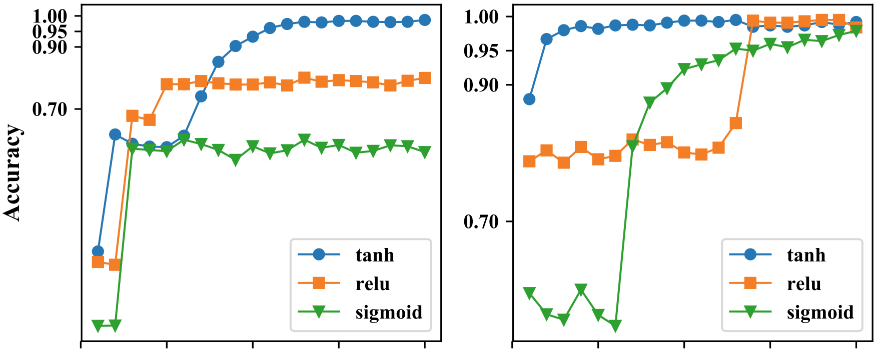
Data type	Full set	Train set	Test set
Normal	972,781	97,278	60,593
Dos	3,883,370	391,458	223,298
Probing	41,102	4107	2377
R2L (Remote to Local)	1126	1126	5993
U2R (User to Root)	52	52	39
Total	4,898,431	494,021	292,300

3 Experimental Results and Analysis

In order to verify the actual detection effect of the proposed CNN model, this paper implements the code based on TensorFlow to test KDDCUP'99 data set. The details of KDDCUP'99 data set are shown in Table 2.

Firstly, this paper designs experiments to compare the performance of different activation functions on the proposed model. Secondly, the optimal activation functions are selected to complete the whole model. Then, we design a five-category experiment to test the performance of the proposed method for intrusion detection.

There are two layers in the proposed CNN model that need to use activation functions: the first fully connected layer and the convolutional layer. Figure 4 shows the performance of the proposed model using different activation functions in the first fully connected layer. It can be found that tanh not only converges faster than other functions but also has the highest accuracy.



(a) Iteration:40 (b) Iteration:200

Fig. 4. Performance comparison of different activation functions in the first fully connected layer: **a** the result of the model's 40-step iteration, **b** the result of the model's 200-step iteration

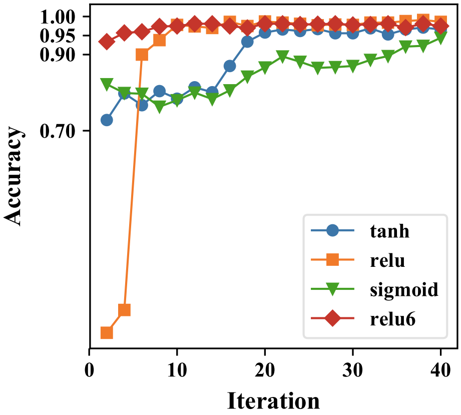


Fig. 5. Performance comparison of different activation functions in the convolutional layer

Figure 5 shows the performance of the proposed method using different activation functions in the convolutional layer. It can be found that ReLU6, an improved activation function of ReLU, has the best performance for the proposed model, which can achieve more than 95% accuracy within five iterations. After comparing the performance of different activation functions, ReLU6 and tanh are used in the convolution layer and the first fully connected layer, respectively, in this paper. The parameter settings of the CNN model are shown in Table 3.

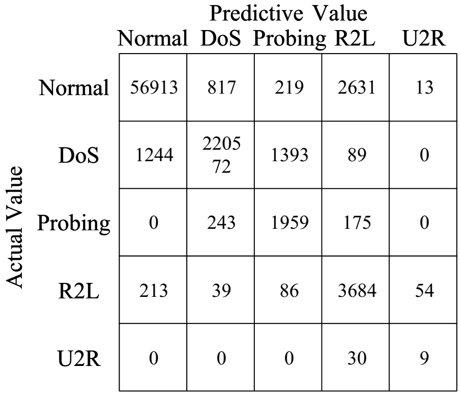
Table 3. Parameter settings of the proposed model

Model	Activation function	Optimization algorithms	Learning rate	β_1	β_2	$\mathcal{E}$ ε
CNN	ReLU6 + tanh	Adam	1×10^4	0.9	0.999	1×10^{-8}

Table 4 shows the results of the five-category test experiment and Fig. 6 shows the confusion matrix of test results. It can be found that the proposed method can achieve considerable accuracy, especially for DoS, which can achieve 98.7% accuracy. However, due to the unbalanced sample categories of the training set, the accuracies of the proposed method for U2R and R2L, which are 61.5 and 23.1%, are not satisfactory.

Table 4. Results of the five-category test experiment

Type	Normal	DoS	Probing	U2R	R2L
Training set	97,278	391,458	4107	1126	52
Test set	60,593	223,298	2377	5993	39
Accuracy(%)	93.9	98.7	82.4	61.5	23.1



		Predictive Value				
		Normal	DoS	Probing	R2L	U2R
Actual Value	Normal	56913	817	219	2631	13
	DoS	1244	220572	1393	89	0
	Probing	0	243	1959	175	0
	R2L	213	39	86	3684	54
	U2R	0	0	0	30	9

Fig. 6. Confusion matrix of test results in the five-category test experiment

4 Conclusion

In this paper, the intrusion detection data set is converted to grayscale through a series of processing, after which the data is trained and tested by CNN. The implementation of the whole model, based on TensorFlow, can achieve considerable detection accuracy rapidly and satisfy the real-time detection requirements of online systems.

However, the accuracy of the proposed method in the five-classification experiment of KDDCUP'99 data set still needs to be improved, especially for the classification results of different attack types.

In addition, the intrusion detection data set used in this paper is still preprocessed manually. In the future work, we will try to extract the corresponding features from the original network traffic, so that the proposed method can have better applicability.

Acknowledgements. This work was supported in part by the National Key R&D Program of China under Grant 2018YFC0807101, in part by the National Natural Science Foundation of China under Grant 61701503/61571082, and in part by the Ministry of Science and Technology of China (MOST) Program of International S&T Cooperation under Grant 2016YFE0123200. The work of S. Hu was supported in part by National Natural Science Foundation of China through the Research Fund for International Young Scientists under Grant 61750110527.

References

1. Almseidin M, Alzubi M, Kovacs S et al (2017) Evaluation of machine learning algorithms for intrusion detection system. In: 15th IEEE international symposium on intelligent systems and informatics. IEEE, pp 277–282
2. Amudha P, Karthik S, Sivakumari S (2014) Classification techniques for intrusion detection-an overview. Int J Comput Appl 76(16):33–40
3. Godin F, Degrave J, Dambre J, De Neve W (2018) Dual rectified linear units (DReLUs): a replacement for Tanh activation functions in quasi-recurrent neural networks. Pattern Recogn Lett 116:8–14
4. Samrin R, Vasumathi D (2018) Hybrid weighted K-means clustering and artificial neural network for an anomaly-based network intrusion detection system. J Intell Syst 27(2):135–147
5. Divyasree TH, Sherly KK (2018) A network intrusion detection system based on ensemble CVM using efficient feature selection approach. Procedia Comput Sci 143:442–449
6. Hu J, Zhang J, Zhang C, Wang J (2016) A new deep neural network based on a stack of single-hidden-layer feedforward neural networks with randomly fixed hidden neurons. Neurocomputing 171:63–72
7. Mingyu Q, Ming L, Yanming F (2015) Research on network intrusion detection using machines based on principal component support vector analysis. Netinfo Secur 15(2):15–18
8. Ludwig SA (2019) Applying a neural network ensemble to intrusion detection. J Artif Intell Soft Comput Res 9(3):177–188
9. Liang Y, Niu D, Hong WC (2019) Short term load forecasting based on feature extraction and improved general regression neural network model. Energy 166:653–663
10. Srivastava N, Hiton GE, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Application of Neural Network in Performance Evaluation of Satellite Communication System: Review and Prospect

Shaochi Cheng(✉), Yuan Gao, Jiang Cao, Yang Guo, Yanchang Du, and Su Hu

PLA Academy of Military Science, Beijing 100091, China
csc04@tsinghua.org.cn

Abstract. Satellite communication has become an indispensable means of communication. As satellite communication constellations become more and more complex, performance evaluation for satellite design, networking and applications grows more and more important. This paper summarizes current research on performance evaluation of satellite communication system and proposes application prospect of neural network in performance evaluation of satellite communication system. Identifying key parameters, adjusting evaluation model adaptively and comparing different satellite constellations' performance are supposed to be three key application areas.

Keywords: Neural network · Performance evaluation · Satellite communication

1 Introduction

Nowadays, satellite communication has become an indispensable means of communication because it has longer communication distance, more extensive coverage and higher quality of transmission than ground communication. There are totally 2062 operating satellites which include 773 communication satellites as of November 30, 2018 [1]. These communication satellites are playing an increasingly important role in agriculture, fisheries, transportation, disaster relief and national defense. In recent years, many advanced satellite communication programs have been put forward. For instance, SpaceX has planned to launch nearly 12,000 communication satellites by the mid-2020s [2], and Amazon said that it was planning to launch thousands of communication satellites to achieve global coverage in April 2019 [3]. These satellite constellations consist of a surprising number of satellites and highly complex satellite communication links, which would need performance evaluation for satellite design, networking and applications.

Neural network is a basic algorithm for deep learning. Its development history has influenced the development of deep learning, thus affecting the history of artificial intelligence. Recently, the improvement of computing power and diversity of data feature provided by big data has created favorable conditions for the rapid development of neural network. Neural network could classify, cluster, and predict data, which

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 239–244, 2020.
https://doi.org/10.1007/978-981-15-0187-6_27

shows great potential for processing unstructured data such as text, image, audio, video and time series. New types of neural network structures such as convolutional neural network, recurrent neural network, and generative adversarial network have sprung up.

BP neural network is currently the most commonly used neural network. Its algorithm could be described as follows:

(1) There are N sample data:

$$(X_k, Y_k), X_k = (x_{1k}, x_{2k}, \ldots, x_{nk}), Y_k = (y_{1k}, y_{2k}, \ldots, y_{mk}), \quad k = 1, 2, \ldots, N.$$

(2) There are L layers in neural network, $L \geq 3$.

(3) There are n^l nodes in layer $l, l = 1, 2, \ldots, L$. The input layer has $|X_k| = n$ nodes, and the output layer has $|Y_k| = m$ nodes.

(4) $f(x)$ indicates the activation function of nodes.

(5) The learning rate and error precision of neural network are denoted as $\eta \in (0, 1)$ and ε.

(6) The number of iterations t is initialized as 1.

(7) X_k is input into neural network, the output of every node is O^l_{jk}:

$$\begin{aligned} I^l_{jk} &= \sum_{i=1}^{n^{(l-1)}} w^{(l-1)}_{ij} O^{(l-1)}_{ik} \\ O^l_{jk} &= f(I^l_{jk}) \end{aligned}$$

(8) I^l_{jk} and O^l_{jk} are the input and output of node j in layer l, w^l_{ij} is the weight between node i in layer l and node j in layer $l+1$.

(9) The output layer error is E_{jk}, $j = 1, 2, \ldots, m$

$$E_{jk} = \frac{1}{2}\left(y_{jk} - O^L_{jk}\right)^2$$

(10) The weights are modified as follows:

$$\left[w^{(l-1)}_{ij}\right]^* = w^{(l-1)}_{ij} - \eta \frac{\partial E^l_{jk}}{\partial w^{(l-1)}_{ij}}$$

(11) $k = k + 1$, the iteration number $t = t + 1$, iterations continue until the output layer error or iteration number meets the requirements.

2 Current Research on Performance Evaluation of Satellite Communication System

Performance evaluation of satellite communication system could be divided into three categories: analytical method, software simulation method and index system method.

2.1 Analytical Method

Analytical method does mathematical modeling of satellite constellation topology, communication channel and information transmission. Mathematical model could be used to evaluate a certain interesting satellites' performance parameter with respect to other parameters and find out key impact factors.

Yonghwa Lee et al. analyzed the conventional and general channel model of satellite system in the mobile environment, and studied the adversary impact of long propagation delay with respect to various user mobility and Doppler-shifted carrier frequency [4]. Charilaos I. Kourogiorgas and Athanasios D. Panagopoulos applied stochastic differential equations to describe mathematical framework of rain attenuation and did numerical calculation of outage probability of satellite communication with respect to rain attenuation [5]. Younes Seyedi et al. provided a statistical model for coverage of mobile low earth orbit satellite communication, and studied coverage time with respect to satellite orbit parameters [6].

Analytical method is useful to improve satellites' communication performance by determining and adjusting key impact factors. However, analytical method usually evaluates a single performance parameter such as time lag and targets a single satellite. The overall performance of satellite constellation could not be described by analytical method accurately. Plus, in order to get mathematical model and do numerical analysis, analytical method ignores many impact factors and reserves limited parameters, which could be inconsistent with real satellite communication scenario.

2.2 Software Simulation Method

Software simulation method models satellite constellation by orbit parameters and sets communication performance parameters such as power, elevation angle and rain attenuation. This method could simultaneously evaluate multi-parameters of satellite communication system and demonstrate the dynamic changes of satellite communication links.

A group of Chinese scientists used STK software to do dynamic modeling and simulation for low earth orbit satellite networks, and evaluate Quality of Service of the constellation by comparing users' location update rate with various satellite spot beams [7]. Five Italian researchers applied MATLAB/Simulink software to model and simulate forward, return and mesh communication links of GEO satellites, and provided guidelines for the trade-off between efficiency and robustness according to mission performance requirements [8]. In [9], OPNET software was used to evaluate double-edge satellite terrestrial networks' resource allocation efficiency and service time delay.

Software simulation method could do more varieties of performance evaluation of satellite communication system than analytical method could. However, software simulation needs researchers to set some parameter values by experience before evaluation. If some predetermined parameter values are inconsistent with reality, simulation results would not be able to evaluate real performance of satellite communication system.

2.3 Index System Method

Index system method is a three-step process. Firstly, the index system of satellite communication system is established by seeking for experts' opinions. The second step is to quantify indexes according to indexes' characteristics. Finally, the performance is evaluated by weighted sum of quantitative indexes. Index system method could provide a quantitative result of the overall performance of satellite communication system.

In [10], index system construction of satellite communication system was based on four domains: physical domain, information domain, recognition domain and social domain. Three principles which were needed to be followed to determine indexes were proposed in [11]. Satellite performance indexes influenced each other. Index system was supposed to be constructed from the holistic perspective. Plus, indexes needed to be layered and took account of actual satellite communication scenario. Liu et al. [12] divided indexes into coverage, transmission, processing and protection according to satellite communication process, and then quantified indexes through numerical calculation and experts' scores. Index weights were calculated by experts' scores, and performance was evaluated by indexes' weighted sum.

Index system method is greatly influenced by experts' subjective judgment. For instance, index weights are determined by experts' experience. However, satellite communication system has various application scenarios. Different scenarios have different index weights. Scenario design would use up much labor power, material resources and financial capacity, and index weights could not be adjusted automatically. Furthermore, once a certain real application scenario is not considered in advance, performance evaluation targeting this scenario would be omitted.

3 Application Prospect of Neural Network in Performance Evaluation of Satellite Communication System

Using neural network to evaluate the performance of satellite communication system has broad prospects, such as identifying key parameters, adjusting evaluation model adaptively and comparing different satellite constellations' performance.

3.1 Identifying Key Parameters

In the future, a satellite communication constellation would commonly contain more than 1000 satellites. Furthermore, a satellite communication constellation would consist of various orbits' satellites, inter-satellite communication links would also exist in large numbers, the number of mobile satellite communication terminals would increase dramatically. As a result, evaluation parameters will grow in magnitude, the interaction between evaluated parameters will grow more and more complex, thus existing evaluation methods are difficult to identify key parameters.

Neural network method can firstly set initial weights of evaluation parameters, and then train the neural network to correct these weights through the actual communication data. The parameters that could get larger weights can be considered as key parameters, which are less affected by the subjective experience of experts.

3.2 Adjusting Evaluation Model Adaptively

In the performance evaluation of existing satellite communication system, the communication scenario is fixed in advance, and weights of evaluation parameters are set for this communication scenario. For example, the transmission capability evaluation typically assumes interference only comes from noise. However, in an actual satellite communication scenario, the rain attenuation may have a serious impact on the transmission capacity. As a result, the assessment results are difficult to match the actual situation. It is necessary to modify the communication scenario and evaluation parameters' weights artificially. There are many kinds of satellite communication scenarios, thus artificial modification requires a lot of manpower, material and financial resources, and it is difficult to operate.

The neural network can adaptively adjust weights of the evaluation parameters according to different evaluation scenarios. For example, a neural network model for evaluating terrestrial mobile communication scenarios can be used to evaluate maritime mobile communication scenarios. Although land mobile communication focuses on signal attenuation caused by building occlusion, maritime mobile communication focuses on rain attenuation, as long as there is actual communication data, neural network can adaptively adjust the evaluation model through training data, and manual adjustment is rare.

3.3 Comparing Different Satellite Constellations' Performance

Different communication needs require different satellite constellations. For example, more coverage needs more GEO satellites, less delay needs more LEO satellites. In order to find out the optimal constellation to meet a certain communication need, performance comparison of various constellation orbits becomes a must. Existing methods set constellation orbits by experience, and performance comparison is done by computer simulation. This method is greatly influenced by experts' subjective judgment, and only covers a small number of orbits.

Neural network could classify or cluster constellation orbit parameters based on satellite constellation simulation results when there is a communication need. When the classifying or clustering model of constellation orbit parameters is established, performance comparison of most constellation orbits is feasible. Computer simulation can further verify the correctness of the comparison, and later guide the improvement of neural networks.

4 Conclusion

This work summarizes current research on performance evaluation of satellite communication system and proposes application prospect of neural network in performance evaluation of satellite communication system. In the future work, a neural network based on actual satellite communication scenario will be given to verify the feasibility of performance evaluation.

Acknowledgements. The work of Su Hu was jointly supported by the MOST Program of International S&T Cooperation (Grant No. 2016YFE0123200), National Natural Science Foundation of China (Grant No. 61471100/61571082/61701503), and Science and Technology on Electronic Information Control Laboratory (Grant No. 6142105040103). The author would also like to thank all the reviewers, their suggestions help improve the work a lot.

References

1. Union of Concerned Scientists (2019) UCS_Satellite_Database_4-1-2019. https://www.ucsusa.org/nuclear-weapons/space-weapons/satellite-database
2. WIKIPEDIA (2019) Starlink (satellite constellation). https://en.wikipedia.org/wiki/Starlink_(satellite_constellation)
3. Sheetz M (2019) Here's why Amazon is trying to reach every inch of the world with satellites providing internet. https://www.cnbc.com/2019/04/05/jeff-bezos-amazon-internet-satellites-4-billion-new-customers.html
4. Lee Y, Choi JP (2018) Performance evaluation of high-frequency mobile satellite communications. Citation information: https://doi.org/10.1109/access.2019.2909885, IEEE Access
5. Kourogiorgas CI, Panagopoulos AD (2014) Application of stochastic differential equations for the performance evaluation of broadband satellite communication systems. In: 2014 XXXIth URSI general assembly and scientific symposium (URSI GASS). IEEE Conferences
6. Seyedi Y, Safavi SM (2012) On the analysis of random coverage time in mobile LEO satellite communications. J IEEE Commun Lett 16(5):612–615
7. Feng S, Zhu H, Li G (2008) Dynamic modeling and simulation for LEO satellite networks. In: 2008 11th IEEE international conference on communication technology. IEEE Conferences
8. Di Iorio E, Ruini R, De Nicola V, Miglietta A, Winkler R (2012) End-to-end system performance evaluation in a Forward and Return satellite communications link. In: 2012 IEEE First AESS European conference on satellite telecommunications (ESTEL). IEEE Conferences
9. Wang P, Zhang J, Zhang X, Liu L, Wang Y, Ouyang L (2018) Performance evaluation of double-edge satellite terrestrial networks on OPNET platform. In: 2018 IEEE/CIC international conference on communications in China (ICCC Workshops). IEEE Conferences
10. Xu X-L, Hu X-F, Qin Y-G (2015) A type of effectiveness evaluation assessment indicator system for satellite communication system based on the "Four-Domain". J Command Control 1(2):220–222
11. Liu X, Kong D, Hu J, XU G (2013) Research into the communication capability index system of military satellite communication system. J Shipboard Electron Countermeasure 36 (1):73–75
12. Liu X, Kong D, Li H, Tao Y (2012) An effectiveness evaluation method of satellite communication system. J Aerosp Electron Warfare 28(5):30–33

Construction of Marine Target Detection Dataset for Intelligent Radar Application

Hao Ding[1], Ningbo Liu[1,2](✉), Wei Zhou[1], Yonghua Xue[1],
and Jian Guan[1]

[1] Naval Aviation University, YanTai, People's Republic of China
hao3431@tom.com, lnb198300@163.com
[2] No. 23 Institute of the Second Academy, CASIC, Beijing,
People's Republic of China

Abstract. With the development of artificial intelligence technology, intelligent processing technology, which is represented by deep learning theory, has been widely used in radar community. The precondition of its application is the support of a large number of training dataset. However, for the technical direction of marine target detection, open dataset is insufficient to meet application requirements. To this end, framework of marine target detection dataset for intelligent radar application is constructed in this paper. It consists of three core modules, namely the detection equipment and data acquisition system, data management and processing, and integrated display. At present, sea clutter property cognition dataset, clutter suppression and target detection dataset, target characteristics dataset and detection performance evaluation dataset are preliminarily obtained. Dataset management and property analysis software are also developed, so as to serve applications more conveniently.

Keywords: Sea clutter · Target detection · Intelligent radar · Dataset

1 Introduction

For marine radars operating in high sea states and complex meteorological conditions, it is very difficult to detect targets in the presence of complex sea clutter, especially for targets with low grazing angles, slow speed and small RCS. Such target signals are embedded in sea clutter in both time domain and Doppler domain, resulting in a significant detection performance degradation [1–6].

In recent years, with the development and application of artificial intelligence technology, such as machine learning and deep learning theory, radar signal processing is also expected to break through the original theoretical framework and steps into the era of intelligence. In fact, in the past two years, intelligent processing technology, which is represented by deep learning theory, has been widely applied in radar. It is a more efficient intelligent signal processing method with the advantages of automatic extraction of deep features with higher accuracy [7]. However, the precondition for deep learning application is the support of sufficient training dataset. At present, open

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 245–253, 2020.
https://doi.org/10.1007/978-981-15-0187-6_28

dataset for SAR image recognition is already available, namely the MASTER datasets, but for marine target detection, open dataset is insufficient to meet application requirements [8].

Aiming at this problem, framework of marine target detection dataset for intelligent radar application is constructed in this paper, on the basis of reviewing existing dataset. Supported with previous measurement experiments, several datasets such as sea clutter property cognition dataset, clutter suppression and target detection dataset, target characteristics dataset and detection performance evaluation dataset are preliminarily obtained. At present, these datasets are in the stage of accumulation and enrichment stage, which still need the support of measured data with diverse experimental conditions such as sea areas or sea states. When the datasets are nearly sufficient, they will be gradually opened up and shared for public use in radar community.

2 Description of Existing Marine Target Detection Dataset

2.1 IPIX Radar Dataset

The dataset was acquired by McMaster University in Canada during experiments with the IPIX radar and was managed and maintained by a team led by Professor Haykin. The data were stored in complete netCDF format. It contains 1993 and 1998 dataset, which are typical X-band low grazing angle data [9]. The 1993 dataset was measured in Dartmouth, southern Nova Scotia, Canada. The cooperative target is a spherical block of styrofoam, wrapped with wire mesh. It has a diameter of one meter. The average SCR varies in the range 0–6 dB. The 1998 dataset was collected at Grimsby, Ontario. The radar site was located at east of the Place Polonaise at Grimsby, Ontario, looking at Lake Ontario from a height of 20 m.

2.2 CSIR Radar Dataset

In 2006 and 2007, the Defence, Peace, Safety, and Security Unit of the Council for Scientific and Industrial Research (CSIR) in Pretoria, South Africa, conducted two sea clutter measurement trials to address some of the limitations of the IPIX dataset [10]. During the trail, various boats (Fig. 1) were deployed with conditions ranging from calm to rough seas. The boats sailed a number of maneuver at different ranges and azimuth angles. The ground truth tracks of the boats were estimated using a differential processing GPS receiver. In addition, datasets were recorded for a large variety of non-cooperative boats of opportunity [11].

Fig. 1. Cooperative target boats [11]

2.3 DSTO Radar Dataset

From 1993 to 2008, the Defence Science and Technology Organization (DSTO) in Australia conducted six shore-based and airborne radar measurement trails with three radar bands (L, S and X bands), and datasets were constructed covering various data types [12–18]. DSTO dataset supported several task background effectively. For example, the L-band dataset was in support of SEA1448 [14], the ANZAC Anti-Ship Missile Defence (ASMD) program. The S-band dataset was in support of the ANZAC Anti-Ship Missile Defence (ASMD) upgrade project [13].

2.4 UCL Radar Dataset

In 2010 and 2011, trails were conducted within the collaboration between University College London (UCL) and the University of Cape Town (UCT) to collect and analyze data on multistatic sea clutter and maritime targets, using the S-band netted radar system NetRAD. The system consists of three distinct, but essentially identical radar nodes. GPS disciplined oscillators that have been developed by the University of Cape Town were used to synchronize the nodes both in time and frequency. The marine environment information during the trail was provided by the weather station nearby [19, 20].

3 Construction of Marine Target Detection Dataset

3.1 Construction Framework

The overall framework of dataset construction can be decomposed into three closely related and mutually supported modules (as shown in Fig. 2).

The first is detection equipment and data acquisition system, which mainly solves the problem of data sources and storage. This module mainly includes radar detection equipment (including multi-type experimental radars), satellite ground station and photoelectric equipment, AIS and ADS-B equipment, high-speed data acquisition and storage equipment, etc. What's more, weather instrument, compass, GPS, laser rangefinder, wave buoy, multi-type standard reflector, sailing boat, rubber boat and other auxiliary experimental facilities are also equipped.

The second is data management and processing module, including radar database, auxiliary sensor database, and supported management and analysis software, so as to manage various types of data more efficiently. In addition, a signal processing and data processing platform composed of high-performance computers is also constructed in this module to support the development and verification of algorithms, such as pulse compression, STC, clutter suppression, CFAR, TBD, track initiation, data association and track filtering.

The third is integrated display module, which mainly completes radar data display, auxiliary sensor data display and comprehensive situation data display functions. It can display AIS, ADS-B, electronic chart and other auxiliary information superimposed on radar video, as well as the comprehensive situation of radar, photoelectricity, AIS and other targets detected by multi-sensors.

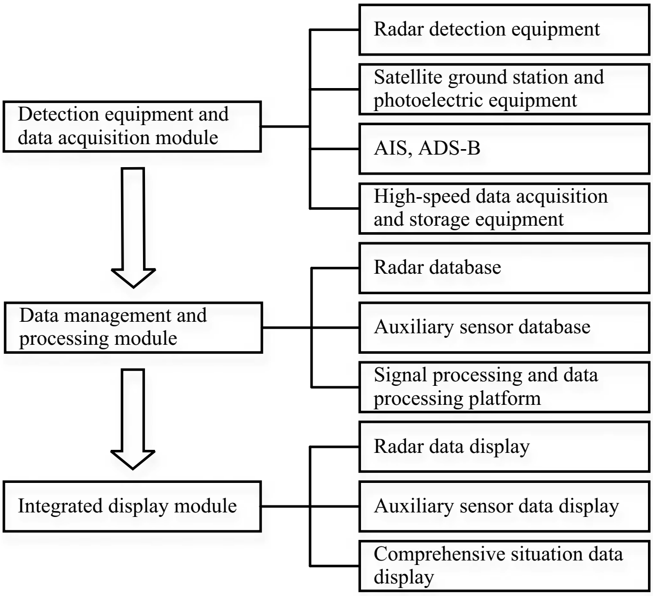

Detection equipment and data acquisition module
Data management and processing module
Integrated display module
Radar detection equipment
Satellite ground station and photoelectric equipment
AIS, ADS-B
High-speed data acquisition and storage equipment
Radar database
Auxiliary sensor database
Signal processing and data processing platform
Radar data display
Auxiliary sensor data display
Comprehensive situation data display

Fig. 2. Overall framework of the constructed datasets

Many types of data are managed in the database, including radar return data, target feature data, AIS and ADS-B data, weather/hydrological data and infrared data, etc. In order to store these data more effectively, a special data and computing center server is set up, and a schematic diagram of the stored data types is shown in Fig. 3.

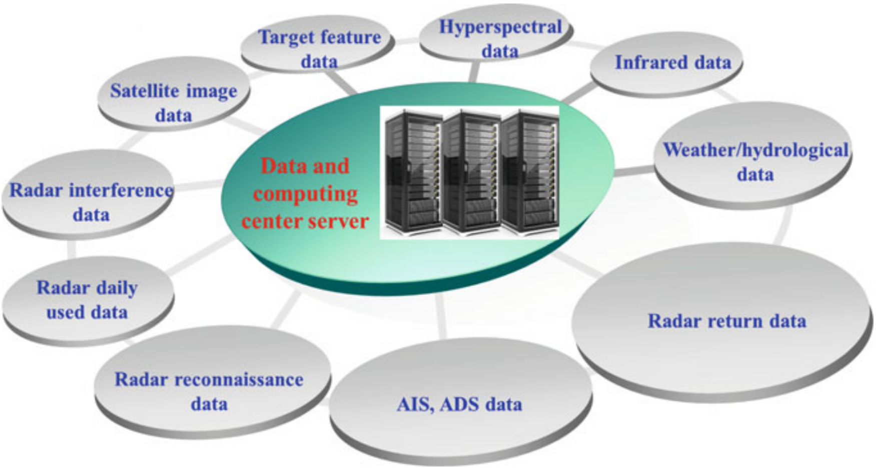


Fig. 3. Schematic diagram of data types

At present, measurement trials have been jointly organized and carried out with industrial departments such as No. 23 Institute of the Second Academy CASIC, No. 38 and No. 22 CETC, AVIC LEIHUA electronic technology research institute. Preliminary marine target detection dataset with various platforms, sea areas, sea states, and target types, are constructed, including sea clutter property cognition dataset, clutter suppression and target detection dataset, target characteristics dataset and detection performance evaluation dataset. In the future, the trials will continue to enrich the experimental data in a large diversity of conditions such as sea areas, sea states, radar working frequencies and many others, so as to make it more practical for intelligent radar applications.

3.2 Dataset Management

The massive datasets obtained through long-term and systematic trials are various and inconvenient to query. In order to manage them more effectively, management software based on Windows architecture server and SQL database version is developed, so as to provide a more efficient data support for intelligent radar. In the software, all radar data and auxiliary sensor data are associated and stored, and detailed information related to the trials is recorded. Data classification and retrieval function is also realized. The software can be connected with SQL database to manage off-line radar data. It can also be connected with radar data acquisition system through data format conversion and data transmission protocol, thus storing the collected data to the database in real time.

The software system design follows the principle of modular, structured and reusability. According to the data processing procedure, the system can be divided into five layers.

The first layer is information import section. It mainly includes five sub-layers information, namely the trial information, radar information, weather information, sea state information and target information. These sub-layers are used to record various levels of data related to the trial, respectively, so as to facilitate the association with subsequent radar data.

The second layer is data import section, which takes the radar data as main index. When importing radar data files, the information layer data, target position, AIS, optical, infrared and satellite image data as well as program documents and processing results are synchronously imported for time synchronization and associated storage with the main index.

The third layer is query setting section, which realizes the separate or joint retrieval according to the test date, radar model, scanning mode, weather, sea state, target type and other categories to improve the retrieval efficiency.

The fourth layer is display of query results. According to the query conditions set in the third layer, the list of all data meeting the conditions is displayed, and the auxiliary sensor data, program documents and existing analysis results associated with the main index (radar data) are displayed synchronously.

The fifth layer is data preview and export section. After the specific item is selected in the query result, the frame header details of radar echo data will be displayed, and the export menu will be added in the right-click menu for the export of all data associated with the current item.

This structured design can enhance the independence of each functional module, improve system maintainability and stability, and facilitate software function upgrade, maintenance and future extension. The overall diagram of software is shown in Fig. 4.

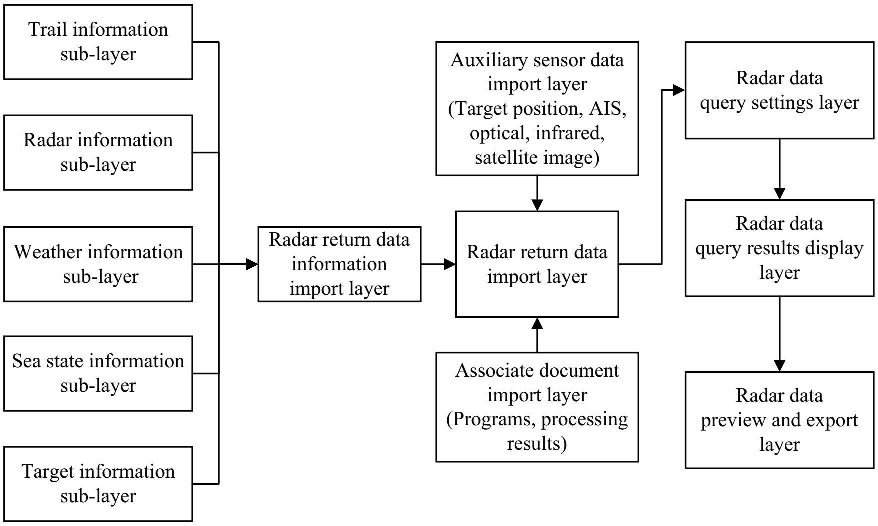

Trail information sub-layer
Radar information sub-layer
Weather information sub-layer
Sea state information sub-layer
Target information sub-layer
Radar return data information import layer
Auxiliary sensor data import layer (Target position, AIS, optical, infrared, satellite image)
Radar return data import layer
Associate document import layer (Programs, processing results)
Radar data query settings layer
Radar data query results display layer
Radar data preview and export layer

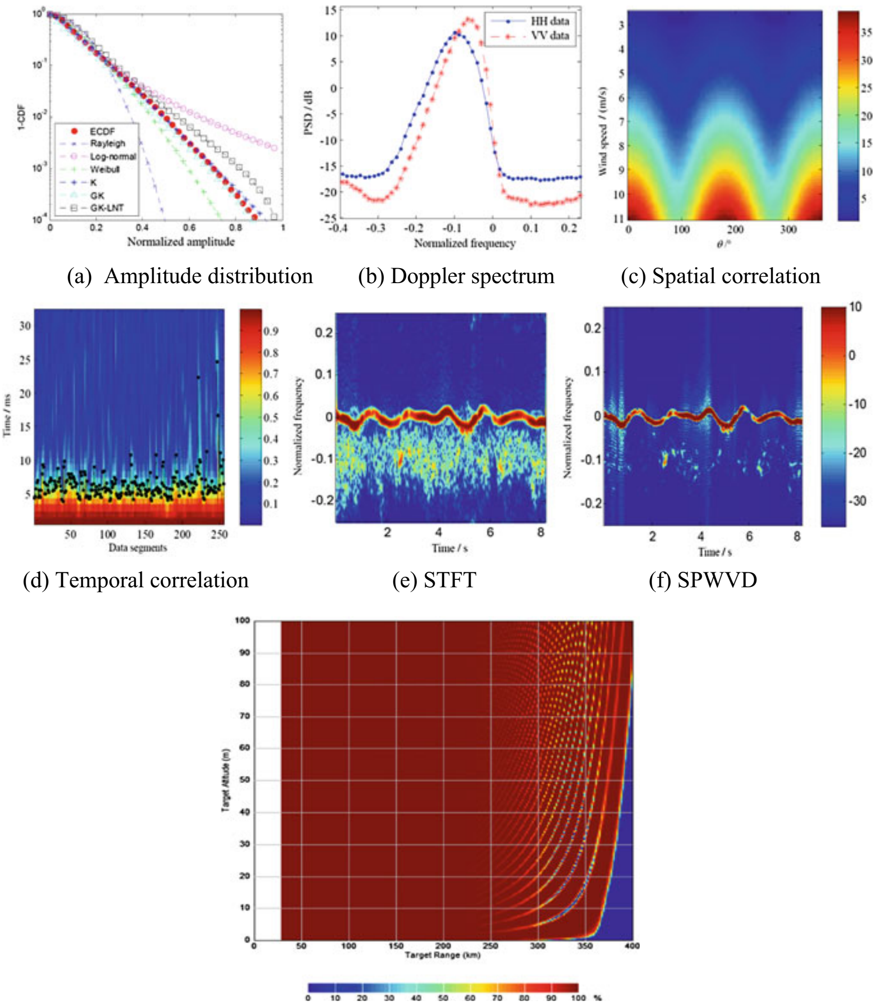
Fig. 4. Overall diagram of software

3.3 Dataset Property Analysis

In order to serve applications more conveniently, analysis methods for sea clutter property cognition dataset, clutter suppression and target detection dataset, target characteristics dataset and detection performance evaluation dataset are researched, and data analysis and processing software is developed. For the online real-time data or offline data queried in the database, the analysis and processing can be completed efficiently and conveniently, and the analysis results can be output in parameter list or visualization form.

The software integrates amplitude distribution modeling, Doppler spectrum analysis and modeling, spatial correlation analysis and modeling, FRFT domain characteristic analysis, nonlinear fractal characteristic analysis and other modules. It can perform functions such as experimental data preprocessing, existing model analysis and evaluation, performance verification of new models, sea clutter and target difference feature extraction, feature space construction, clutter suppression and target detection, detection performance evaluation, which will provide important support for the optimization design, parameter selection and performance prediction of the target detection algorithms.

Take the time/frequency domain characteristics analysis and modeling module as an example, in Fig. 5a–g, typical visualization results of analysis results are shown. Among them, Fig. 5g is a schematic diagram of detection performance evaluation based on data set and specific radar operating parameters.



(a) Amplitude distribution (b) Doppler spectrum (c) Spatial correlation

(d) Temporal correlation (e) STFT (f) SPWVD

(g) Schematic diagram of detection performance evaluation

Fig. 5. Visualization of the analysis results

4 Conclusions

In this paper, existing marine target detection dataset is briefly introduced firstly, including IPIX dataset, CSIR dataset, DSTO dataset and UCL dataset. Then the framework of marine target detection datasets for intelligent radar application is constructed, which consists of detection equipment and data acquisition system module, data management and processing module, and integrated display module. Special data and computing center server is set up to store various types of data, such as radar return data, target feature data, AIS and ADS-B data, weather/hydrological data and infrared data. Dataset management and property analysis software are also developed. The datasets are scheduled to be released gradually for public use in radar community in the near future.

Acknowledgements. This work was supported in part by the China Postdoctoral Science Foundation (2017M620862) and the National Natural Science Foundation of China (61871392, 61531020, 61871391).

References

1. Ward K, Tough R, Watts S (2013) Sea clutter: scattering, the K-distribution and radar performance, 2nd edn. The Institution of Engineering and Technology, London
2. Ding H, Liu N, Dong Y et al (in press) Overview and prospects of radar sea clutter measurement experiments. J Radars
3. Ding H, Dong Y, Liu N, Wang G, Guan J (2016) Overview and prospects of research on sea clutter property cognition. J Radars 5(5):499–516
4. Ding H, Guan J, Liu N et al (2015) New spatial correlation models for sea clutter. IEEE Geosci Remote Sens Lett 12(9):1833–1837
5. Ding H, Guan J, Liu N et al (2016) Modeling of heavy tailed sea clutter based on the generalized central limit theory[J]. IEEE Geosci Remote Sens Lett 13(11):1591–1595
6. Shi S, Shui P (2018) Sea-surface floating small target detection by one-class classifier in time-frequency feature space. IEEE Trans Geosci Remote Sens 56(11):6395–6411
7. Wang J, Zheng T, Lei P, Wei S (2018) Study on deep learning in radar. J Radars 7(4): 395–411
8. Su N, Chen X, Guan J, Mou X, Liu N (2018) Detection and classification of maritime target with micro-motion based on CNNs. J Radars 7(5):565–574
9. Drosopoulos A (1994) Description of the OHGR database. Defence research establishment Ottawa, Technical Note, pp 94–14
10. De Wind HJ, Cilliers JC, Herselman PL (2010) Sea clutter and small boat radar reflectivity databases. IEEE Signal Process Mag 32(2):145–148
11. Herselman PL, Baker CJ, deWind HJ (2008) An analysis of X-band calibrated sea clutter and small boat reflectivity at medium-to-low grazing angles. Int J Navig Obs 1–14
12. Antipov I (1998) Analysis of sea clutter data. Technical Report, DSTO-TR-0647
13. Dong Y, Merrett D (2008) Statistical measures of S-band sea clutter and targets. Technical Report, DSTO-TR-222
14. Dong Y, Merrett D (2010) Analysis of L-band multi-channel sea clutter. Technical Report, DSTO-TR-2455
15. Antipov I (2001) Statistical analysis of northern Australian coastline sea clutter data. Technical Report, DSTO-TR-1236
16. Crisp DJ, Stacy NJS, Goh AS (2006) Ingara medium-high incidence angle polarimetric sea clutter measurements and analysis. Technical Report, DSTO-TR-1818
17. Watts S, Rosenberg L, Bocquet S et al (2016) Doppler spectra of medium grazing angle sea clutter part 1: characterisation. IET Radar Sonar Navig 10(1):24–31

18. Watts S, Rosenberg L, Bocquet S et al (2016) Doppler spectra of medium grazing angle sea clutter part 2: model assessment and simulation. IET Radar Sonar Navig 10(1):32–42
19. Ritchie M, Stove A, Woodbridge K et al (2016) NetRAD: monostatic and bistatic sea clutter texture and Doppler spectra characterization at S-Band. IEEE Trans Geosci Remote Sens 54 (9):1–11
20. Fioranelli F, Ritchie M, Griffiths H et al (2017) Analysis of polarimetric bistatic sea clutter using the NetRAD radar system. IET Radar Sonar Navig 10(8):1356–1366

Pedestrian Retrieval Using Valuable Absence Augmentation

Xiaolong Hao[1,2], Shuang Liu[1,2](✉), Zhong Zhang[1,2], and Tariq S. Durrani[3]

[1] Tianjin Key Laboratory of Wireless Mobile Communications
and Power Transmission, Tianjin Normal University, Tianjin, China
shuangliu.tjnu@gmail.com

[2] College of Electronic and Communication Engineering,
Tianjin Normal University, Tianjin, China
haoxiaolong17@gmail.com, zhang8848@gmail.com

[3] Department of Electronic and Electrical Engineering,
University of Strathclyde, Glasgow, Scotland, UK
t.durrani@strath.ac.uk

Abstract. In this paper, we propose a novel data augmentation method named valuable absence augmentation (VAA) in order to alleviate the overfitting and evaluate the influence of the pedestrian valuable parts for the network performance. Specifically, we first train a base convolutional neural network model and obtain the attention map of the pedestrian. Then, we use the attention map to generate new samples. Finally, original samples and new samples are combined to fine-tune the base network model. We conduct experiments on a large-scale pedestrian retrieval database, i.e., Market-1501. Experimental results show that the pedestrian valuable part has a crucial influence for the network performance and that the proposed method achieves better performance than other state-of-the-art methods.

Keywords: Convolutional neural networks · Pedestrian retrieval · Data augmentation

1 Introduction

Pedestrian retrieval aims at retrieving a specified pedestrian from a large-scale image pool (Gallery) that consists of images taken by different cameras at different times [1,2]. It is widely used in many applications including crowd counting, multi-camera tracking, video surveillance, etc. [3–6], and gradually becomes a hot research focus. Although pedestrian retrieval has been studied for many years, it still confronts many challenges such as background clutter, different illumination, various body poses, and so on.

With the help of the convolutional neural networks (CNNs), pedestrian retrieval has achieved remarkable performance [7–10]. However, when an exces-

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 254–261, 2020.
https://doi.org/10.1007/978-981-15-0187-6_29

sively complex CNN model is trained, the overfitting might occur, which degenerates the performance of pedestrian retrieval. In order to alleviate the overfitting, the most straightforward method is to increase the number of training samples. However, it is very expensive to obtain a large number of labeled samples. Data augmentation can extend the training set through transforming the original training samples, and it is widely used in deep learning, especially in training deep CNN model. The most common methods of data augmentation include rotation, adding noises, flipping, cropping, etc. Krizhevsky et al. [11] randomly flipped samples in the training set to train the deep CNN model. In order to obtain various training samples, Simonyan et al. [12] randomly cropped training samples and extracted sub-patch of samples to expand the training set. In [13], random erasing was proposed as a data augmentation approach, which randomly selects a rectangle region on the sample and assigns random values. Xiao et al. [7] combined several databases to jointly train the deep CNN model. The above-mentioned methods only focus on increasing the training sample number to relieve the overfitting, but it only increases the generalization ability of the network model to some extent. This is because in the view of the network model, there is only a subtle difference between original samples and increasing samples, and that is to say, these increasing samples cannot force the network model to learn different features.

In this paper, we propose a novel data augmentation method named valuable absence augmentation (VAA) to increase the number of training samples and meanwhile utilize these increasing samples to train the network in order to force the network to learn variable features. Specifically, we first train a base network model using the original training set. Next, we obtain the attention map extracted from convolutional activation maps of the trained base network model. Afterward, we take valuable parts of the attention map and project these parts to the original sample and set the corresponding position of the original sample to zero. As a result, we can obtain a large number of images without valuable parts for the base network model. Finally, we combine these samples with original samples to fine-tune the base network model so that the network model can learn new discriminative features from valuable absence parts of the sample. Hence, the generalization ability of the network model can be improved.

We evaluate the proposed VAA on a large-scale pedestrian retrieval database (Market-1501), and extensive experiments show that the proposed VAA is effective for reducing the risk of overfitting and improving the generalization ability of the network model. In addition, during the test stage, we also evaluate the impact of removing valuable parts with different proportions from the original sample. Experimental results show that the performance of pedestrian retrieval decreases with the increase of removing valuable parts ratio.

In summary, the contribution of this paper is: (1) We propose VAA to increase the number of training set in order to mitigate the risk of overfitting; (2) we utilize these increasing samples and original samples to fine-tune the base network, which can improve the generalization ability of the network model; and (3) we prove the effectiveness of VAA on a large-scale database.

2 Approach

In this section, we present details of the proposed VAA for training the deep CNN model. This method mainly consists of three phases. First, we train a base network model till convergence using original samples. Afterward, we obtain the attention map form the trained base network model and set the value of valuable parts on the original sample to zero in order to obtain new samples. Finally, we utilize these new samples and original samples to fine-tune the trained base network model.

With the development of deep learning, many CNN models have been applied to pedestrian retrieval such as CaffeNet, VGG, ResNet, and so on. We adopt the ResNet-50 model [14] as the base network model and set the neuron number of fully connected (FC) layer to the identity number of pedestrian retrieval database, i.e., 751 for Market-1501. The structure of the base network model is shown in Fig. 1. During training, we treat pedestrian retrieval as a classification task and use cross-entropy loss to fine-tune the network model. During testing, we remove the FC layer and utilize the remaining layer to extract the feature of pedestrian sample.

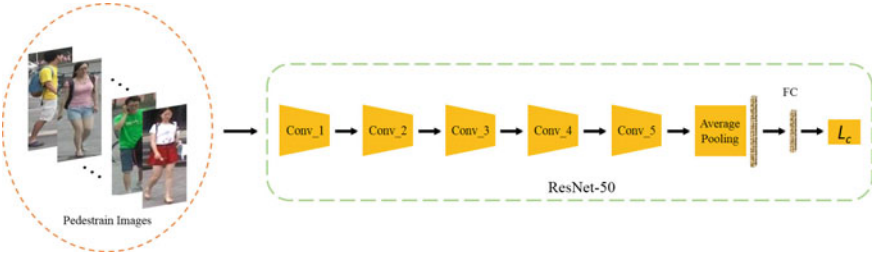


Fig. 1. Structure of the base network model.

After training the base network model, the next phase is to find valuable parts on the pedestrian sample and set the pixel value of these parts to zero in order to obtain new samples. Since there is no location annotation when training the network model, we cannot directly find valuable parts of pedestrian sample. Considering that the attention map can reflect the part of network model concern, we can locate valuable parts of the pedestrian sample. Inspired by this, we use the attention map to find valuable parts of the pedestrian sample. The attention map AM_n obtained at the nth layer of the network model is represented as follows:

$$AM_n = \sum_i (FM_i)^2, \tag{1}$$

where FM_i represents the ith convolutional activation map in this layer. It is worth mentioning that the operation is between elements.

Afterward, we adjust the size of the attention map to the size of the original pedestrian sample and record valuable parts on the original pedestrian sample according to the attention map. After that, we set the pixel value of these parts to zero. As a result, we obtain new samples with valuable part absence for the base network model. We list some new and original pedestrian samples in Fig. 2. Note that we assign the same identity label with the original sample to the new one.

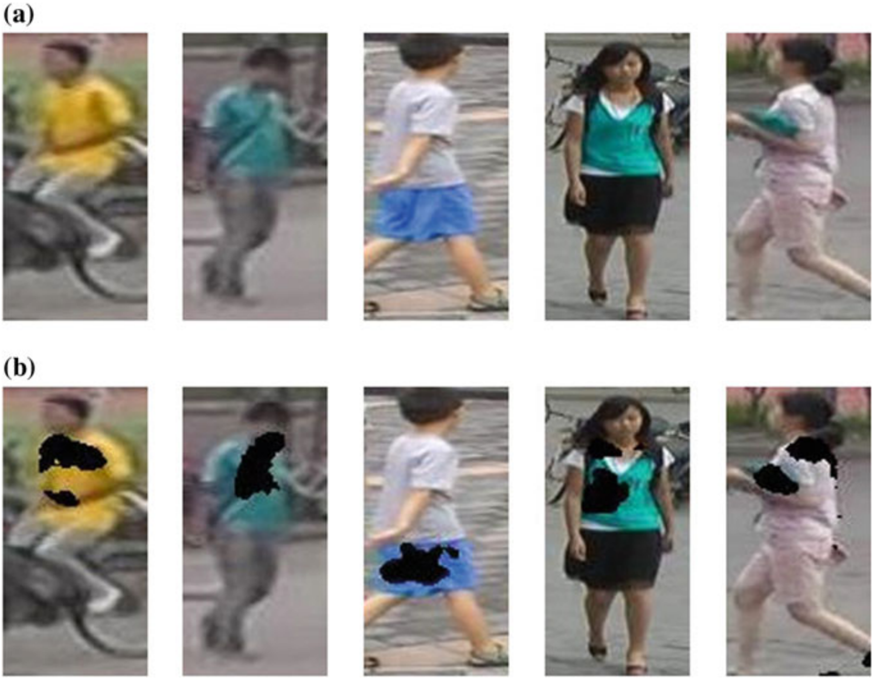


Fig. 2. a Original pedestrian samples and **b** new pedestrian samples obtained by VAA.

Finally, we fine-tune the base network model using these new samples. Nevertheless, we find it is difficult to set the learning rate when fine-tuning the network model. On the one hand, the learning rate of fine-tuning network model should be smaller than that of training from scratch. On the other hand, using a small learning rate may cause the network model to fall into a partial optimization. At the same time, we find that new samples can force the network model to learn new discriminative features from other regions of the pedestrian sample. Therefore, we combine original samples and new samples to train the base network model by using the same settings as the first phase. The final results show that combining original samples and new samples to train the network can produce better results.

3 Experiments

In this section, we conduct extensive experiments on a large-scale pedestrian retrieval database (Market-1501 [15]). We first evaluate the influence of the pedestrian valuable parts for network performance. Then, we evaluate the proposed VAA on this database. During the evaluation process, the cumulative match characteristic and the mean average precision (mAP) are treated as criteria.

3.1 Market-1501 Database

Market-1501 is a large-scale pedestrian retrieval database. It is collected from six different cameras and contains 32,668 pedestrian images with 1501 identities. This database has 12,936 pedestrian images of 751 pedestrian in the training set, 19,732 pedestrian images of 750 pedestrian in the testing set, and 3368 pedestrian images as the query set. Each pedestrian has an average of 17.2 images, and each pedestrian image is detected by the Deformable Parts Model (DPM) detector [16], which is close to the real environment. We report the Rank-1 accuracy and mAP for the sake of fair comparison with other methods.

3.2 Implementation Details

When training the base network model, we utilize random horizontal flipping as a data augmentation method and resize the size of pedestrian image to 256×128 as the input of the base network model. We set the batch size to 32 and the number of epoch to 60 and use stochastic gradient descent (SGD) to optimize this model's parameters. As for the learning rate, we set the last layer (FC) to 0.1 and other layers to 0.01, and the learning rate gradually decreases at the rate of 0.1 after 40 epochs. In the second phase, we select convolutional activation maps from the output of Conv_2 and utilize Eq. 1 to obtain the attention map. We project the first 5% valuable parts of the attention map to original pedestrian image and set corresponding positions to zero. As a result, we obtain new samples. In the last phase, all parameter settings are the same as in the first phase. During testing, we extract the 2048-dim feature vector from the output of pooling5 to represent pedestrians and compute the similarity of features by the Euclidean distance.

3.3 Evaluation of Valuable Parts for Network Performance

We evaluate the influence of the pedestrian valuable parts for the network performance in this subsection, and experiment results are listed in Table 1. We can see that the performance of pedestrian retrieval becomes worse and worse as the proportion of removing valuable parts increases. Specifically, when 5% valuable parts are removed, the performance of pedestrian retrieval decreases by only 7% in the Rank-1 accuracy, but when 20% valuable parts are removed, the performance of pedestrian retrieval drops drastically and only reaches the result of 52.2% Rank-1 accuracy. This indicates that the valuable part has a crucial influence on the performance of the network model.

Table 1. Influence of valuable parts of the pedestrian on the network performance, where C expresses the proportion of removing valuable parts.

C (%)	0 (Baseline)	5	10	15	20
Rank-1 (%)	85.2	78.2	70.1	61.3	52.1

3.4 Evaluation of the Proposed VAA for the Performance of Pedestrian Retrieval

We evaluate the proposed VAA on the Market-1501 database in this subsection. In order to fully present the effectiveness of the proposed VAA, we first train the base network model (Baseline) and achieve the result of 85.2% Rank-1 accuracy and 68.3% mAP. As shown in Table 2, we can see that the proposed VAA achieves the best results compared with the other methods. Specifically, compared with the Baseline, the proposed VAA obtains improvements with the result of 0.9% Rank-1 accuracy and 1.0% mAP. This mainly includes two reasons. On the one hand, the proposed VAA can produce new samples, which can mitigate the overfitting and improve the generalization ability of the network model. On the other hand, these new samples can force the network model to learn new discriminative features from other pedestrian regions.

Table 2. Comparison of the proposed VAA with the state-of-the-art methods on the Market-1501 database.

BoW [15]	34.4	14.1
SSDAL [17]	39.4	19.6
LOMO+XQDA [1]	43.8	22.2
Gated [18]	65.9	39.6
IDE [19]	73.7	51.5
PAR [20]	81.0	63.4
SVDNet [21]	82.3	62.1
LSRO [22]	84.0	66.1
APR [23]	84.3	64.7
TriNet [24]	84.9	69.1
Baseline	85.2	68.3
Ours	**86.1**	**69.3**

4 Conclusion

In this paper, we have evaluated the influence of the pedestrian valuable parts for the network performance and have proposed the VAA to increase samples. Specifically, we first train a base network model using the original training set.

Afterward, we employ the attention map to remove valuable parts from the original pedestrian sample and therefore obtain a large number of samples with valuable part absence. Finally, we combine original pedestrian samples and new pedestrian samples to fine-tune the base network model so that the network model can learn new discriminative features from valuable absence parts of the sample. Experimental results have shown that the proposed method achieves better performance than other state-of-the-art methods.

Acknowledgements. This work was supported by National Natural Science Foundation of China under Grant No. 61501327 and No. 61711530240, Natural Science Foundation of Tianjin under Grant No. 17JCZDJC30600, the Fund of Tianjin Normal University under Grant No.135202RC1703, the Open Projects Program of National Laboratory of Pattern Recognition under Grant No. 201800002, and the Tianjin Higher Education Creative Team Funds Program.

References

1. Liao S, Hu Y, Zhu X, Li S (2015) Person re-identification by local maximal occurrence representation and metric learning. In: IEEE conference on computer vision and pattern recognition, pp 2197–2206
2. Liu S, Hao X, Zhang Z (2018) Pedestrian retrieval via part-based gradation regularization in sensor networks. IEEE Access 6:38171–38178
3. Zhang Z, Wang C, Xiao B, Zhou W, Liu S, Shi C (2013) Cross-view action recognition via a continuous virtual path. In: IEEE conference on computer vision and pattern recognition, pp 2690–2697
4. Zhang Z, Wang C, Xiao B, Zhou W, Liu S (2014) Cross-view action recognition using contextual maximum margin clustering. IEEE Trans Circ Syst Video Technol 24(10):1663–1668
5. Li A, Wu Z, Lu H, Chen D, Sun G (2018) Collaborative self-regression method with nonlinear feature based on multi-task learning for image classification. IEEE Access 6:43513–43525
6. Li A, Chen D, Wu Z, Sun G, Lin K (2018) Self-supervised sparse coding scheme for image classification based on low rank representation. PLOS One 13(6):1–15
7. Xiao T, Li H, Ouyang W, Wang X (2016) Learning deep feature representations with domain guided dropout for person re-identification. In: IEEE conference on computer vision and pattern recognition, pp 1249–1258
8. Zheng Z, Zheng L, Yang Y (2017) A discriminatively learned CNN embedding for person re-identification. ACM Trans Multimedia Comput Commun Appl 14(1):1–20
9. Sun Y, Zheng L, Yang Y, Tian Q, Wang S (2018) Beyond part models: person retrieval with refined part pooling (and a strong convolutional baseline). In: European conference on computer vision, pp 1–17
10. Zhong Z, Zheng L, Li S, Yang Y (2018) Generalizing a person retrieval model hetero-and homogeneously. In: European conference on computer vision, pp 172–188
11. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105

12. Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: International conference on learning representations, pp 135–153
13. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2017) Random erasing data augmentation (Online). arXiv:1708.04896
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, pp 770–778
15. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: a benchmark. In: IEEE international conference on computer vision, pp 1116–1124
16. Felzenszwalb PF, Girshick RB, Mcallester DA, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32:1627–1645
17. Su C, Zhang S, Xing J, Gao W, Tian Q (2016) Deep attributes driven multi-camera person re-identification. In: European conference on computer vision, pp 475–491
18. Varior RR, Haloi M, Wang G (2016) Gated siamese convolutional neural network architecture for human re-identification. In: European conference on computer vision, pp 791–808
19. Zheng L, Yang Y, Hauptmann AG (2016) Person re-identification: past, present and future (Online). arXiv:1610.02984
20. Zhao L, Li X, Wang J, Zhuang Y (2017) Deeply-learned part-aligned representations for person re-identification. In: IEEE conference on computer vision and pattern recognition, pp 3219–3228
21. Sun Y, Zheng L, Deng W, Wang S (2017) SVDNet for pedestrian retrieval. In: IEEE international conference on computer vision, pp 3820–3828
22. Zheng Z, Zheng L, Yang Y (2017) Unlabeled samples generated by GAN improve the person re-identification baseline in vitro. In: IEEE international conference on computer vision, pp 3754–3762
23. Lin Y, Zheng L, Zheng Z, Wu Y, Yang Y (2017) Improving person re-identification by attribute and identity learning (Online). arXiv:1703.07220
24. Hermans A, Beyer L, Leibe B (2017) In defense of the triplet loss for person re-identification (Online). arXiv:1703.07737

An End-to-End Neural Network Model for Blood Pressure Estimation Using PPG Signal

Cuicui Wang, Fan Yang, Xueguang Yuan(✉), Yangan Zhang,
Kunliang Chang, and Zhengyang Li

State Key Laboratory of Information Photonics and Optical Communications,
Beijing University of Posts and Telecommunications, Beijing 100876, China
yuanxg@bupt.edu.cn

Abstract. With the increasing number of hypertension patients, the monitoring of blood pressure information becomes an important task. In this study, an end-to-end approach is proposed to estimate blood pressure from the pulse wave signal. In this approach, a normalized single pulse wave is the input of a neural network, which consists of the convolutional layers and the recurrent layers, then outputs the corresponding blood pressure. The convolutional layers consist of one-dimensional convolutional layers and depth-separable convolutional layers. The gated recurrent unit (GRU) is used in the recurrent layer. Finally, a dense layer is used to output estimated values of blood pressure. In comparison with previous approaches, the proposed method does not require complicated feature extraction. It is only necessary to input a single pulse wave into the neural network and blood pressure can be estimated. The proposed approach is tested in the multi-parameter intelligent monitoring in intensive care (MIMIC) dataset, and the average absolute error is 3.95 mmHg for systolic blood pressure and 2.14 mmHg for diastolic blood pressure. This result fulfills the international standard of blood pressure measurement, which shows the proposed approach is simple and effective. In practice, the proposed method is designed to obtain blood pressure information from pulse waves.

Keywords: Photoplethysmography · Blood pressure measurement · Neural network · Transfer learning

1 Introduction

Blood pressure (BP) is the pressure of circulating blood on the walls of blood vessels. Monitoring of human blood pressure can reflect well the function of the cardiovascular system of the human body, so it is a great way to judge the disease and observe the therapeutic effect clinically. According to the association for the advancement of medical instrumentation (AAMI), the mean and deviation absolute error between the device and the mercury standard sphygmomanometer should be lower than 5 ± 8 mmHg [1].

With the development of deep learning, many scientists are committed to taking advantages of the neural network to discover new blood pressure measurement

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 262–272, 2020.
https://doi.org/10.1007/978-981-15-0187-6_30

methods using Photoplethysmography (PPG) signals, which can achieve a simple and accurate blood pressure measurement. Teng et al. [2] established a linear regression model using four extracted features of PPG signals for BP estimation: a width of 2/3 pulse amplitude, a width of 1/2 pulse amplitude, systolic upstroke time, and diastolic time. Kurylyak [3] has done a lot more in the characteristics of the interval, a total of 21 parameters as the input of the BP network, in order to predict blood pressure. Gao et al. [4] developed a method for BP estimation using the regression support vector machine (RSVM) method, with RBF kernel and discrete wavelet transform, and obtained better performance. Wang et al. [5] used MTM method for feature extraction, and an artificial neural network (ANN) is used for BP estimation. Wu [6] used SWT to decompose the pulse wave and extracted ten characteristic parameters as the input of ANN network. Wen et al. [7] used the sum of two Gaussian functions as the model to fit the pulse wave envelope, and the parameters of the Gaussian function after fitting were the input, and the determination of systolic and diastolic pressure was performed through two feed-forward neural networks. The methods mentioned above all need to perform feature extraction on the original pulse wave data. The operation of the feature extraction is burdensome and inevitably added in human subjective consciousness, so that the nonlinear relation between the pulse wave and the blood pressure cannot be extracted very well.

This paper presents an end-to-end method to obtain blood pressure. Using a single complete pulse wave as the input of the neural network, the estimated blood pressure values can be obtained. In this study, we extract the signal from the MIMIC database [8] for network training and testing. Finally, the average absolute error on 15 datasets is 3.95 ± 4.38 mmHg for systolic blood pressure (SBP) and 2.14 ± 2.40 mmHg for diastolic blood pressure (DBP) when using 6000 transfer learning data, fulfilling the international standard of blood pressure measurement, and the work avoids the complicated feature extraction work, which shows the proposed method is simple, efficient, and accurate. In summary, the main contribution of this paper is proposing an end-to-end neural network for the measurement of blood pressure which proved to be effective.

In the rest of this paper, Sect. 2 introduces the MIMIC database, the details of the method are recommended in Sects. 3 and 4 introduce the implementation and the results, a discussion is given in Sect. 5 finally summarize the article.

2 Data Source

The MIMIC database contains data recorded on 90 ICU patients. The data for each case includes the signals obtained from the bedside monitor and periodic measurements and clinical data obtained from the patient's medical records. The length of these records varies, with almost all records being at least 20 h and many being 40 h or longer. In short, the database contains nearly 200 days of real-time signals and related data.

The data for each case is kept in one folder, usually containing hundreds of separate files, each containing a 10-minute recording signal, and then it can be assembled without clearance and forming a continuous record. The types of information contained in the records of different cases are slightly different, including roughly arterial blood pressure (BP), heart rate (HR), PPG, pulmonary arterial pressure (PAP), etc. [9, 10].

Because the MIMIC database contains a large variety of physiological information data with high quality, it has been widely used in blood pressure prediction research. In this experiment, PPG and BP signals are used, and the specific signal waveforms are shown in Fig. 1.

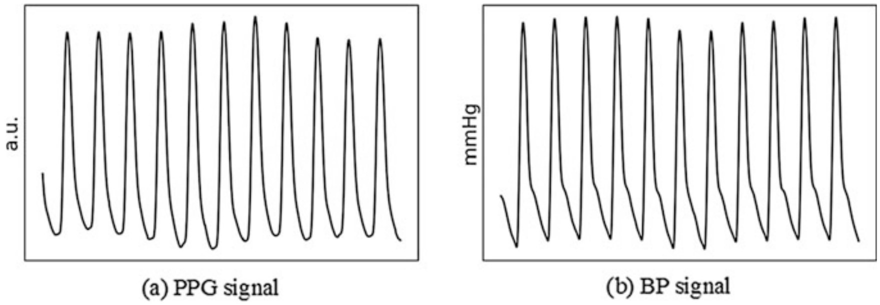


Fig. 1. PPG and BP signal of subject 230

3 Proposed Method

The main flow of the proposed method is as follows. Firstly, a single complete pulse wave is extracted and normalized by an interpolation method. Then the normalized pulse wave would be used as the input of the neural network, and finally, the estimations of blood pressure value can be obtained. Figure 2 shows the schematic illustration of presented BP estimation frameworks.

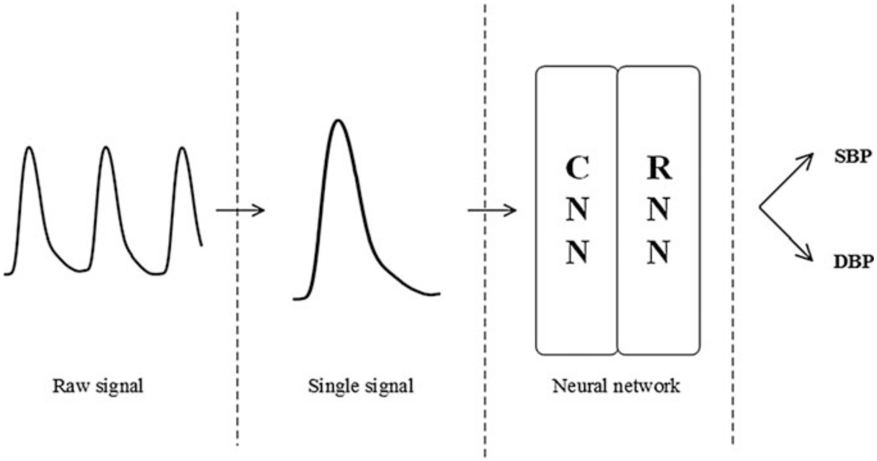


Fig. 2. Schematic illustration of presented BP estimation frameworks. A single pulse wave was cut from raw pulse wave and normalized to 100 points. It is then fed into a neural network consisting of a convolutional network and a recurrent network, and finally, the estimated blood pressure values, namely SBP and DBP, are obtained

3.1 Data Processing

The MIMIC dataset contains several different kinds of physiological information for volunteers, such as HR, PPG, and BP signals. However, the types of physiological information contained in each sample are slightly different, and only those samples that contain both the PPG signal and the BP signal are used. Due to the fact that there are some time periods that no information was collected in the original data, which are of no value to the experiment. So, the first step is to eliminate it. After discarding the data segments that do not contain information, the original data is cut off, eventually resulting in several consecutive signal segments of varying length. Then, a single waveform cut is made for the successive signal fragments. As the segmentation is done, the single waveform is finally normalized to 100 points by an interpolation method.

The raw data is given as a blood pressure waveform, and what needed is the blood pressure value, that is SBP and DBP in a single corresponding waveform period. SBP refers to the maximum blood pressure value in systole and DBP refers to the blood pressure value in end-diastole. After cutting the single pulse waveform, the corresponding blood pressure waveform can be obtained which represents a complete blood pressure cycle. The peak point value of blood pressure wave is the ground-truth of SBP, and the end point of blood pressure wave is the ground-truth of DBP.

3.2 Model Structure

This section introduces the module and structure of the model. A convolutional neural network (CNN) is a feed-forward neural network which uses a variation of multilayer perceptrons designed to require minimal preprocessing. Its artificial neurons can respond to a part of the surrounding cells in the coverage area and have excellent performance for large-scale image processing [11–13]. A CNN includes input and output layer, and a plurality of hidden layers. The hidden layer of CNN generally includes convolutional layers, pooling layers, fully connected layers, and normalization layers. The convolutional layer is the core building block of a CNN. The parameters of this layer consist of a set of learnable filters that have a small receive field but extend through the entire depth of the input volume. During the forward pass, each filter is convolved over the width and height of the input volume, a dot product is calculated between the input and the items of the filter, and a two-dimensional activation map of the filter is generated. The specific calculation formula is as follows.

$$Z^{l+1}(i,j) = [Z^l \otimes w^l](\mathrm{i},\mathrm{j}) + b = \sum_{k=1}^{K_l}\sum_{x=1}^{f}\sum_{y=1}^{f}\left[Z_k^l(s_0 i + x, s_0 j + y)w_k^{l+1}(x,y)\right] + b$$

$$(\mathrm{i},\mathrm{j}) \in \{0,1,\ldots,L_{l+1}\} \quad L_{l+1} = \frac{L_l + 2p - f}{s_0} + 1$$

where b is the amount of deviation, Z^l and Z^{l+1} represent the convolution input and output of the $1+1$th layer, also known as the feature map, L_{l+1} is the size of Z^{l+1}, $\mathrm{Z}(\mathrm{i},\mathrm{j})$ corresponds to the pixel of the feature map, and K is the channel of the feature map number, f, s_0, and p are the convolution layer parameter, corresponding to the convolution kernel size, convolution step, and padding layer. The depth-separable convolutional layer is slightly different from the ordinary convolutional layer. It uses different filters to convolve each channel to obtain the output of the corresponding channel and then blends the information. The depth-separable convolutional layer reduces the required parameters over ordinary convolutional layers. The ordinary convolution layer considers the channel and the area simultaneously, while the depth-separable convolution layer considers only the area first, and then considers the channel to achieve the separation of the channel and the area.

GRU is a type of recurrent neural network (RNN). This structure appeared in 2014 [14] and its performance on polyphonic music modeling and speech signal modeling [15–17] is comparable to that of long short-term memory (LSTM) [18], but it uses fewer parameters as lacking an output gate [19, 20]. Since the three gates in LSTM contribute differently to improve their learning ability, the contribution of small gates and their corresponding weights can be omitted, which can simplify the neural network structure and improve its learning efficiency. GRU is an algorithm based on the above concept, and its corresponding loop unit contains only two gates: the update gate and the reset gate, wherein the function of the reset gate is similar to the input gate of the LSTM unit, and the update gate simultaneously implements the function of the forgetting gate and the output gate. The GRU is updated as follows:

$$\begin{gathered}
h^{(t)} = g_u^{(t-1)} h^{(t-1)} + (1 - g_u^{t-1}) f_h [w h^{(t-1)} + (u X^{(t)}) g_r^{(t)} + b] \\
g_u^{(t)} = \mathrm{sigmoid}(w_u h^{(t-1)} + u_u X^{(t)} + b_u) \\
g_r^{(t)} = \mathrm{sigmoid}(w_r h^{(t-1)} + u_r X^{(t)} + b_r)
\end{gathered}$$

where X is the input sequence, b is the amount of deviation, h is called the system state of the cyclic neural network, w and u are the weight of the loop node, the former is called the state–state weight, the latter is called the state-input weight, and f_h is the excitation function of the system state, g is the gating updated with time, the foot u and r means the update door and the reset door.

Considering that the PPG signal is a sequence signal, the proposed method uses CNN to extract features and uses RNN to extract time-related information. The model structure of the proposed model is shown in Fig. 3, and the specific parameters are shown in Table 1. The processed pulse wave signal is unified to 100 points, and then it is fed into the feature extraction layer composed of convolutional layers and max pooling layers. The convolution kernel's size is 3 for the convolutional layers and 2 for max pooling layers which is aim to reduce the amount of computation and network size. By concatenating several one-dimensional convolutional layers to continuously increase the nuance of the convolutional kernel, a larger range of feature information can be extracted. The dropout layer [21] is added at the end of CNN to prevent overfitting and improve the robustness of the model. The information output by the convolutional network is normalized by the BN layer [22] and sent to GRU to obtain time-related feature information. Finally, the fully connected layer is used to integrate the feature information to obtain the estimated blood pressure values. The convolutional layers used in this model are the one-dimensional convolutional layer and the one-dimensional depth-separable convolutional layer. The one-dimensional convolutional layer extracts different aspects of the original data characteristics, and the depth-separable convolutional layer can reduce the model parameters while preserving the effect, so that the model is simpler. It is also based on this consideration that GRU layer is chosen.

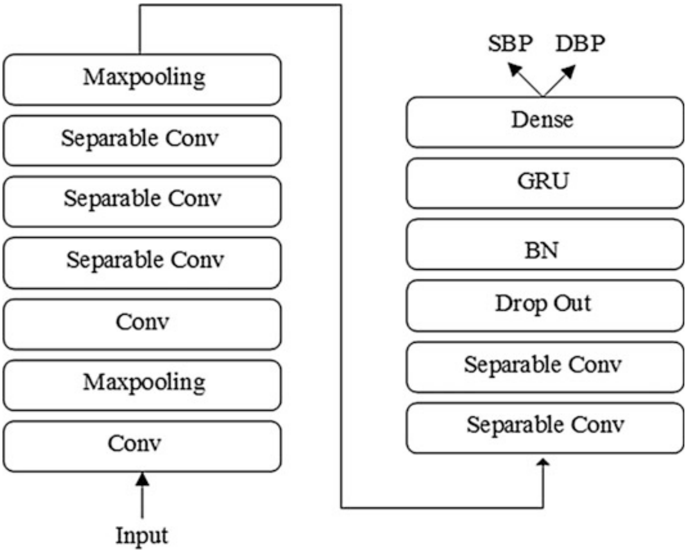


Fig. 3. Proposed method. The normalized single pulse wave is used as input. The convolutional network consists of the one-dimensional convolutional layer and the one-dimensional depth-separable convolutional layer. The GRU is used to reduce the computational complexity of the model while remaining accurate. Finally, the model outputs the estimated blood pressure, which is SBP and DBP

4 Experimental Results

In the experimental session, we did two tests. The first test was to train and test for single-sample data. The second is to use data from all samples to train a basic model, and then use a small amount of data from the new sample for transfer learning.

In the first test, subject 230 is used. After eliminating the useless data segment, the single complete waveform is segmented and normalized, generating 37,600 valid data. In this experiment, the validation set and test set are randomly selected according to the proportion of 20% and 15%, respectively, left the training set maintain relative order. That is, 24,440 data is used to train the model. The Adam optimization function is applied, which has the advantage of fast convergence. It is found that the rectified linear units (Relu) function and sigmoid function are approximately equal, but the convergence speed of Relu function is much fast, so Relu is finally selected as the activation function. Learning rate is divided, 0.001 for front 300 epochs, 0.0001 for 300–500 epochs, and a total of 500 rounds are trained. The batch size is set to 128.

Table 1. Parameters of the proposed model

Type	Kernel	Stride	Outputs	Parameters
Conv.	3	1	512	2048
Pooling	/	2	512	0
Conv.	3	1	256	393,472
Sep Conv.	3	1	256	66,560
Sep Conv.	3	1	128	33,664
Sep Conv.	3	1	128	16,896
Pooling	/	2	128	0
Sep Conv.	3	1	64	8640
Sep Conv.	3	1	64	4352
Dropout	/	/	64	0
BN	/	/	64	64
GRU	/	/	64	24,768
Dense	/	/	1	65
Total parameters: 550,529				
Trainable parameters: 550,497				
Non-trainable parameters: 32				

The absolute error is used as an evaluation criterion in this experiment. The calculation formula is:

$$e = |\mathrm{BP}_{\mathrm{est}} - \mathrm{BP}| \tag{1}$$

where $\mathrm{BP}_{\mathrm{est}}$ is the estimated SBP or DBP using the proposed method, and BP is the reference value obtained from MIMIC database. The Bland-Altman plot [23] is another evaluation indicator to test the consistency between ground-truth and estimates. It shows the difference between each estimate and the associated ground-truth against their average. In this analysis limit of agreement (LOA) is also calculated, which is defined as $[\mu - 1.96\sigma, \mu + 1.96\sigma]$, where μ is the average of the difference and σ is the standard deviation. Pearson correlation between ground-truth and estimated SBP is also used as one of the performance indices.

The average absolute error on the test set of subject 230 is 3.81 ± 3.90 mmHg for SBP, and 1.74 ± 2.28 mmHg for DBP, which both are meeting the requirement. Figure 4 presents the results of several estimated SBP on the test set of subject 230 using the proposed method. Tests are performed on all test samples and 5640 test values are obtained. In order to be more intuitive in the figures, the obtained test values are sorted, and then one point is taken at intervals of 120 points. Finally, 47 points are taken and plotted in the figure. The corresponding point of ground-truth SBP is also plotted in the figure. It can be seen that most of the estimates are close to the ground-truth, showing the proposed method effective.

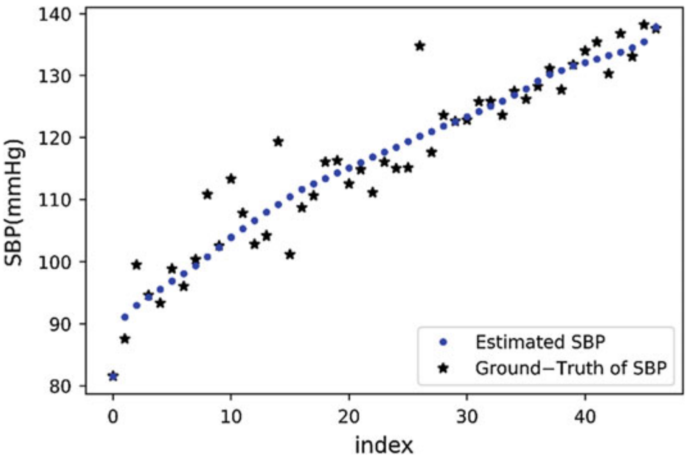


Fig. 4. SBP estimation results on subject 230. Tests and random sampling were performed on the 230 test set, and 47 pairs of estimated values and ground-truth values were obtained. Blue solid dots in the figure represent model estimates and black stars represent ground-truth values. In order to be more intuitive, the values of the 47 points are sorted according to the ground-truth values. The index refers to the serial number of each pair of estimated values and real values

Figure 5 shows the histograms of the errors on subject 230, calculated as the difference between ground-truth of SBP/DBP and the output of the proposed method. The Bland-Altman plot between ground-truth and estimates of SBP on subject 230 is given in Fig. 6a. In this case, the limit of agreement (LOA) was [−9.8053, 10.0505] mmHg and 94.911% of all differences were inside this range. Figure 6b shows the scatter plot between the ground-truth of SBP and the associated estimates on subject 230. It also shows that the fitted line was $Y = 0.8918X + 12.73$, where X denotes the ground-truth of SBP, and Y indicates the associated estimate. The Pearson coefficient in our framework was 0.9344.

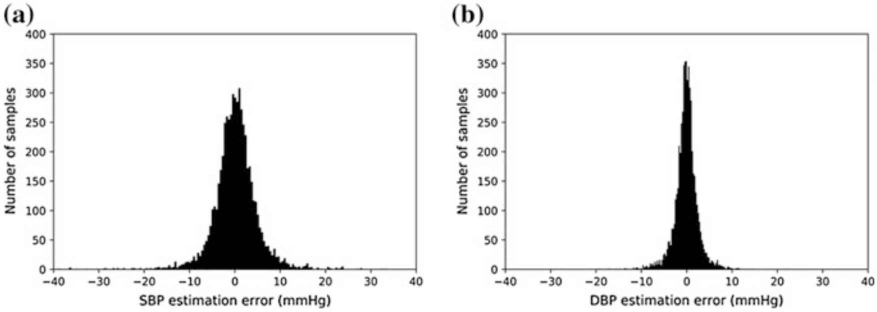
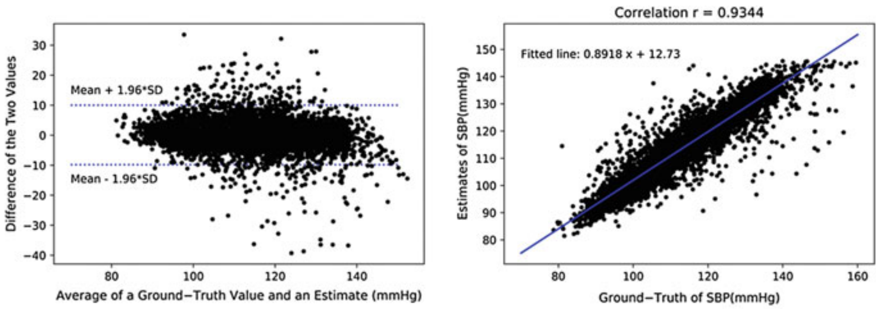


Fig. 5. Histograms of error on subject 230: **a** SBP estimation error and **b** DBP estimation error



(a) The Bland-Altman plot (b) Scatter plot

Fig. 6. Estimation results on subject 230 of SBP

Using single-sample data to make predictions after training, the accuracy is very high, but the generality is insufficient. So we did the second test. The goal is to use a small amount of data for transfer learning to achieve better result. First, all data is divided into training set and test set by sample, and the data of all samples in the training set are used for preliminary training to obtain a basic model. Then, for each test sample in the test set, a small amount of data is taken out in order to perform a transfer learning. After the completion, blood pressure estimation can be performed for the test sample. Table 2 shows the test results.

Table 2. Performance results based on transfer learning measurements

Number of data	4000		6000	
Samples	SBP	DBP	SBP	DBP
Subject 041	2.42 ± 2.69	1.78 ± 1.98	2.43 ± 2.64	1.72 ± 1.91
Subject 216	2.64 ± 3.77	1.86 ± 2.44	2.52 ± 3.64	1.78 ± 2.38
Subject 221	4.49 ± 4.15	2.47 ± 2.42	4.38 ± 3.92	2.39 ± 2.33
Subject 230	3.67 ± 3.92	2.02 ± 2.16	3.52 ± 3.91	1.96 ± 2.14
Subject 237	4.84 ± 5.05	3.05 ± 3.37	4.78 ± 5.02	3.02 ± 3.35
Subject 253	5.18 ± 6.50	3.28 ± 4.03	4.95 ± 6.38	3.14 ± 3.95
Subject 254	4.80 ± 6.05	2.56 ± 2.94	4.64 ± 5.86	2.47 ± 2.96
Subject 276	5.20 ± 4.92	2.33 ± 2.47	5.32 ± 4.96	2.40 ± 2.49
Subject 414	4.40 ± 4.18	2.51 ± 2.38	4.27 ± 4.05	2.47 ± 2.36
Subject 437	5.23 ± 5.31	2.57 ± 2.50	4.97 ± 5.19	2.52 ± 2.49
Subject 449	2.61 ± 3.76	1.28 ± 1.86	2.43 ± 3.64	1.21 ± 1.89
Subject 452	3.27 ± 3.44	1.41 ± 1.36	3.16 ± 3.43	1.36 ± 1.35
Subject 453	3.99 ± 4.18	1.81 ± 1.83	3.88 ± 4.02	1.75 ± 1.76
Subject 474	4.36 + 4.78	2.56 + 2.93	4.24 + 4.78	2.49 + 2.93
Subject 476	3.84 + 4.48	1.54 + 1.75	3.76 + 4.28	1.49 + 1.74
Mean	4.06 + 4.47	2.20 + 2.42	3.95 + 4.38	2.14 + 2.40

Table 3 shows the performance results on the test sets for the linear regression, RSVM-based method, ANN-based method, and the proposed method, presented as mean and standard deviation of absolute error among reference SBP/DBP and estimated values.

Table 3. Performance of different methods [1, 3, 4]

Method	SBP	DBP
Linear regression	9.80 ± 8.09	5.88 ± 5.11
RSVM-based	4.32 ± 3.59	4.31 ± 3.83
ANN-based	4.02 ± 2.79	2.27 ± 1.82
Proposed method	3.95 ± 4.38	2.14 ± 2.40

5 Discussion

As can be seen from Table 2, with 4000 data for transfer learning, the average error of SBP is 4.06 ± 4.47 mmHg, and the error of DBP is 2.20 ± 2.42 mmHg. When using 6000 data, the average error of SBP is 3.95 ± 4.38 mmHg, and the error of DBP is 2.14 ± 2.40 mmHg. This result proves that it is feasible and effective to use transfer learning to improve the original system. And the more transfer learning data used, the better the effect.

It can be seen from Table 3 that the approach in this paper is comparable to the existing best results in accuracy, meeting the criteria for international blood pressure measurement, while do not require complex feature extraction.

There are still some deficiencies in this model. According to past experience, an end-to-end neural network is mostly data-driven. This means that to train such an effective model, there must be a sufficient amount of training data. However, with the development of computer technology and the big data era, the amount of data is no longer a problem. The second problem is that in order to solve the problem of insufficient model universality, we need to do transfer learning, but we can see from the experiment that the more transfer learning data used, the better the effect. So, the next research direction is how to get better results with as little transfer learning data as possible.

6 Conclusion

In this paper, an end-to-end neural network model for blood pressure estimation using PPG signals is proposed. A single complete pulse wave is normalized to be used as input, and a neural network consisting of convolutional layers and recurrent layers is applied to obtain the estimated values of blood pressure. Compared with other methods, the proposed method is better for its accuracy and its simplicity.

We have only made a small part of the attempt, and hope that this work can provide some reference for researchers interested in this work.

References

1. Association for the Advancement of Medical Instrumentation (1996) American national standard for electronic or automated sphygmomanometers. In: ANSI/AAMISP10-1992/A1, Arlington,VA, USA
2. Teng XF, Zhang YT Continuous and noninvasive estimation of arterial blood pressure using a photoplethysmographic approach. In: Presented at the 25th annual international conference of the IEEE engineering in medicine and biology society
3. Kurylyak Y, Lamonaca F, Grimaldi D (2013) A neural network-based method for continuous blood pressure estimation from a PPG signal. In: Presented at 2013 IEEE I2MTC
4. Gao SC, Wittek P, Zhao L, Jiang WJ Data-driven estimation of blood pressure using photoplethysmographic signals. In: Presented at IEEE EMBC annual international conference
5. Wang L, Zhou W, Xing Y et al (2018) A novel neural network model for blood pressure estimation using photoplethesmography without electrocardiogram. J Healthc Eng 1:1–9
6. Wu YD, Zhong SC, Shen YC (2017) Noninvasive continuous blood pressure measurement method based on SWT and ANN. Inf Med Equip 5(32):22–27
7. Wen L, Li ZB, Chen JP et al (2014) Blood pressure measurement algorithm of neural network based on Gaussian fitting. Transducer Microsyst Technol 4(33), 132–134 + 138
8. Goldberger AL, Amaral LA, Glass L et al (2000) PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation 101(23):E215–E220
9. Moody GB, Mark RG (1996) A database to support development and evaluation of intelligent intensive care monitoring. Comput Cardiol 23:657–660
10. Krizhevsky A, Sutskever I, Hinton GE ImageNet classification with deep convolutional neural networks. In: Presented at international conference on NIPS
11. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Comput Sci
12. He K, Zhang X, Ren S et al Deep residual learning for image recognition. In: Presented at IEEE conference on CVPR
13. Cho K, Van Merrienboer B, Gulcehre C et al (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation. Comput Sci
14. Amodei D, Anubhai R, Battenberg E et al (2015) Deep speech 2: end-to-end speech recognition in English and Mandarin. Comput Sci
15. Graves A (2013) Generating sequences with recurrent neural networks. Comput Sci
16. Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks. In: Presented at the 31st international conference on ML
17. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
18. Jozefowicz R, Zaremba W, Sutskever I An empirical exploration of recurrent network architectures. In: Presented at international conference on machine learning
19. Chung J, Gulcehre C, Cho KH et al (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. Comput Sci
20. Zeiler MD (2012) ADADELTA: an adaptive learning rate method. Comput Sci
21. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
22. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. Comput Sci
23. Altman DG, Bland JM (1983) Measurement in medicine: the analysis of method comparison studies. The Statistician 32(3):307–317

Concept Drift Detection Based on Kolmogorov–Smirnov Test

Zhixiong Wang and Wei Wang(✉)

Tianjin Key Laboratory of Wireless Mobile Communications and Power
Transmission, Tianjin Normal University, Tianjin 300387, China
weiwang@tjnu.edu.cn

Abstract. With the advancement of information society, a large amount of data, which is in the form of stream, has been produced in many fields. As a result of its extensive application in the fields of sensor networks, banking and telecommunications, data stream mining is obtaining more attention. One of the most challenging steps to learn from data stream is to react to concept drift, as most of the existing data stream algorithms only deal with abrupt or gradual concept drifts. The existing work of detecting concept drift is mostly based on the changing of error rate of single window, making it difficult to be universally applied to different types of concept drifts. A method of detecting concept drift is proposed in this paper based on Kolmogorov–Smirnov test (K–S test).

Keywords: Data stream · Concept drift · Kolmogorov–Smirnov test

1 Introduction

With the development of economy and society, a large amount of data, which is in the form of stream, has been continuously produced in fields such as financial data analysis, network monitoring, sensor network data processing, credit card fraud monitoring, weather forecasting and electricity price forecasting. This fast-reaching, real-time, continuous, and unbounded data sequence is called data stream. It makes data classification, is based on data stream system, which becomes a research focus. On one hand, because it's fast, infinite, and unreproducible, data mining technology in a relatively mature and static system can't be directly transferred to data stream system. On the other hand, due to the data stream system and application, the concepts in data stream are dynamically changing or even full of noise; as a result, the distribution of data will change with time, which aggravates the complexity of data stream classification. For example, the rules of weather forecast may change with seasons; Methods of Customer online shopping preference analysis may change with customer groups, seller reputation, service type, and other factors; industrial electricity consumption will change periodically as seasons alternate. Therefore, research on detecting concept drift is being become more and more important in data streams. According to the changing speed, we can subdivide concept drift into gradual drift and abrupt drift. If the data distribution in data stream is suddenly replaced by another completely different data distribution in a relatively short period of time, then we can think that abrupt concept drift occurs. Gradual concept drift, however, can only be observed after a rather long period of time

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 273–280, 2020.
https://doi.org/10.1007/978-981-15-0187-6_31

(such as a sensor's gradual failure) and there are more or less similarities among the concepts before and after the concept drift occurs.

At present, many researchers in related fields have proposed some effective algorithms to detect concept drifts of data streams. A constraint penalty regression combiner is used to track concept drifts in Document Ref. [1]. It is the first time that integrated classification technology has been introduced into data stream classification and SEA algorithm has been proposed in Ref. [2]. Gama et al. proposed the DDM algorithm to detect whether the concept drift occurred in the data through the current model error rate and standard deviation statistics, and statistically divide the data state into three states: normal, warning and conceptual drift [3]. However, the DDM method does not have a good effect on the detection of the gradient drift. In order to solve this problem, Baena-Garcia et al. proposed the EDDM [4] algorithm to improve the detection effect of the gradient concept drift. Information entropy is also often used as a detection indicator for drift detection algorithms [5]. Ross et al. applied the exponentially weighted moving average charts (EWMA) to the detection of concept drift, and proposed the ECDD algorithm [6], which uses the exponentially weighted moving average control chart to monitor the error rate when the error rate when a certain threshold is exceeded, it is considered that the concept drift is detected. The drift detection method of comparing data distribution is also a very common method. Recently, Frias. Blarco et al. used moving averages and weighted moving averages to measure changes in the data stream, and experimentally proved that these two statistics are applicable to abrupt drift and gradual drift, respectively [7], Farid proposed Changes in data distribution such as CUSUM cumulative sum and exponential weighted moving average method emphasize the problem of metric determination during detection [8]. Literature [9] compares two data distributions that can continuously expand the window, and then selects different window strategies for detection [9]. Sakthithasan proposed the SeqDrift detection algorithm, which uses reservoir sampling for memory management and improves the sensitivity of conceptual drift detection under slowly varying conditions [10, 11].

This paper proposed a detection method to concept drift by detecting the differences of frequency distribution between data blocks, which is introduced in Sect. 2. According to the method, In Sect. 3 we did related experiments and results analysis. Finally, the conclusions are given according to the results analysis.

2 Concept Drift Detection Based on Kolmogorov–Smirnov Test

2.1 Relevant Definitions

In order to facilitate the elaboration of relevant contents of this paper, the following relevant definitions are given at first:

Definition 1 Concept drift: setting that X_n^t is a sample and y_n^t is sample's class label; the joint probability distribution of the sample generated by the data source changes with time. Supposing that every time t arrives at a data block D_t, $D_t = \{(X_1^t, y_1^t), (X_2^t, y_2^t),$

$(X_3^t, y_3^t), \ldots, (X_n^t, y_n^t)\}$, X_n^t's corresponding distribution is $P_t(X, y)$, if $P_t(X, y) \neq P_{t-1}(X, y)$ then concept drift occurs.

Definition 2 Independent increment process: As the name suggests, it means that the increments are independent of each other. Strictly defined as follows: set $\{X(j), j \geq 0\}$ be a random process. For $0 \leq s \leq j$, the random variable $X(j) - X(s)$ is increment of the random process in the interval $[s, j]$, if any positive integer $n \geq 2$ and any $0 \leq j_0 < j_1 < j_2 < j_3 < \cdots < j_n$, n increments are independent of each other, and $\{X(j), j \geq 0\}$ is autocephalous increment process.

Definition 3 Wiener process: given the process $\{W(j), j \geq 0\}$, if it satisfies a stationary independent increment, $W(0) = 0$, and for any $0 \leq s < j$, $\{W(j) - W(s)\} \sim N(0, \sigma^2(j - s))$.

Definition 4 Brown Bridge: a special Wiener process, is strictly defined as follows: $B(t) := (W(t)|W(T) = 0)$, $t \in [0, T]$.

Definition 5 Kolmogorov distribution is defined as

$$K = \sup_{t \in [0,1]} |B(t)| \tag{1}$$

sup is the smallest upper bound of a set, by finding the distribution of random variables that are bounded by the Brownian motion, its cumulative distribution function can be written as:

$$P(K \leq x) = 1 - 2\sum_{k=1}^{\infty} (-1)^{(k-1)} \mathrm{e}^{-2k^2x^2} = \frac{\sqrt{2\pi}}{x} \sum_{k=1}^{\infty} \mathrm{e}^{\frac{-(2k-1)^2\pi^2}{8x^2}} \tag{2}$$

2.2 Kolmogorov–Smirnov Test

In statistical test, K–S test is used to test whether an empirical distribution conforms to a theoretical distribution or whether there is a significant difference between data distributions. Because it is sensitive to parameters of the data distribution function between both samples, K–S test is used to compare two samples. Kolmogorov–Smirnov test finds the upper bound of the difference between the cumulative probability of experience and the cumulative probability of target distribution at each data point, and lists the following formulas:

$$D_n = \sup_x\{|F_0(x) - F_n(x)|\} \tag{3}$$

sup is the supremum of distance, which is minimum upper bound of absolute value of $[F_n x - F_0(x)]$ under the original assumption $[F_n(x) = F_0(x)]$ that the conditions are true, and the cumulative distribution function of the observed values of random samples is represented by $F_n(x)$, and the cumulative probability distribution function of the theoretical distribution is represented by $F_0(x)$.

$$F_n(x) = \frac{1}{n}\sum_{1}^{n} I_{[-\infty,x]}(X_i) \tag{4}$$

$I_{[-\infty,x]}$ denotes indicator function:

$$I_{[-\infty,x]}(X_i) = \begin{cases} 1, & X_i \le x \\ 0, & X_i > x \end{cases} \tag{5}$$

$$D_n = \max_{1 \le k \le n} \{|F_n(x_k) - F_0(x_k)|, |F_n(x_{k+1}) - F_0(x_k)|\} \tag{6}$$

the single sample detection statistics is Z:

$$Z = \sqrt{n}D_n = \sqrt{n} \max_{1 \le k \le n} \{|F_n(x_k) - F_0(x_k)|, |F_n(x_{k+1}) - F_0(x_k)|\} \tag{7}$$

When the null hypothesis is true, the Z conforms to the Kolmogorov distribution.

k is the ordinal number when ascending order is adopted in the sample. It is seen from the formula that D_n is the largest difference between the adjacent two empirical distribution functions and the target cumulative probability distribution function. Based on the above theory, the two-sample test simply replaces the other distribution of the test statistics with the empirical distribution of the other sample. Assuming that the sample sizes of the two samples are m and n, respectively, $F_m(x)$ and $F_n(x)$ represent the cumulative empirical distribution function of both samples, in the case of $D_j = F_m(x_j) - F_n(x_j)$, the test statistic is approximately normal distribution, and the expression is

$$Z = \max_j |D_j| \sqrt{\frac{mn}{m+n}} \tag{8}$$

The original hypothesis is H_0 that the two data distributions are consistent or the data conforms to the theoretical distribution.

The two-sample K–S test steps:

(1) Assumption: H_0: $F_m(x) = F_n(x)$.
(2) Calculate the absolute difference between the sample's cumulative frequency and the theoretical distribution's cumulative probability, so that the largest absolute difference is $D_{m,n} = \max|D_j|$.
(3) Find the critical value $D(m, n, \alpha)$ by sample size n and significant level α.
(4) p-value $= P(D_{m,n} < D(m, n, \alpha))$.
(5) If p-value ≥ 0.05, that is, the two samples conform to the same distribution.

2.3 Model Description

The detection model based on K–S test is as follows: $D_1, D_2, D_3, \ldots, D_i$, D_{i+1} (each data block contains n samples) denotes that the data stream arriving at t time is cut into data blocks. Setting $\alpha = 0.05$ (the significant level) based on experience, when $n \in [1, i]$ and n is an integer, the significant level of the difference between the two data blocks is measured by p-value. When this p-value is smaller than 0.05, we can draw the conclusion that there are obvious differences between two data blocks, that is, concept drift occurs. When p-value is greater than 0.05, we accept the original assumption to draw the conclusion that there are no obvious differences between two data blocks. We accept the initial hypothesis, it means there is no concept drift.

3 Experimental Process and Results

3.1 Experimental Process

Two data sets are used in experiments, one is simulation data set and the other is radar data set. For the simulation dataset and real dataset experiments, experiments cut the dataset into 12 data blocks, labels them in order, and calculates p-value according to the above model method.

The noise-free simulation data set that has been generated by python is named CIRCLES. The samples are labeled based on a circular function: if a sample is within the circle, then its label is positive; if a sample is beyond the circle, then its label is negative. Concept drift occurs when shifting the center and increasing the size of the circle. Each data block contains 30 samples and each sample contains 2 values. This data set is defined by three circular functions as shown in Table 1.

Table 1. Simulation data description

Center	(0.2, 0)	(0.4, 0)	(1, 0)
Radius	0.2	0.3	0.5

The radar data set is based on data that have been measured by radar in our laboratory in three different states when there is no person, one person, and two persons. The three different states, respectively, represent three different classes: 0, 1, 2. Each data block contains 30 samples and each sample contains 100 values. The category distribution in each data block is shown in Table 2.

Table 2. Radar data description

Data block number	Class	Data block number	Class
0	0, 1	6	0, 2
1	0, 1, 2	7	0
2	0, 1	8	0, 2
3	0, 1, 2	9	0, 1, 2
4	0, 1, 2	10	0, 1
5	0, 1, 2	11	0, 1

3.2 Experimental Results

In the simulation data set experiment, the detection model based on Kolmogorov–Smirnov test is used to detect whether the two data blocks obey the same potential distribution. The horizontal red line indicates that the significant level of α is 0.05. The results show that whether the difference between the data blocks is less than 0.05 or greater than 0.05. If the p-value between the data blocks is greater than 0.05, we will accept the initial assumption that the data blocks before and after conform to the same potential distribution and it means no concept drift occurs between the two data blocks. But if the p-value is less than 0.05, we will turn down the initial assumption as the data blocks before and after do not conform to the same potential distribution, which states clearly that there is an obvious difference between the data blocks before and after, that is, concept drift has occurred. By comparing the significance level α and p-value of simulation data, we can clearly see that the model can accurately detect whether there is a significant difference between the data blocks before and after (Fig. 1).

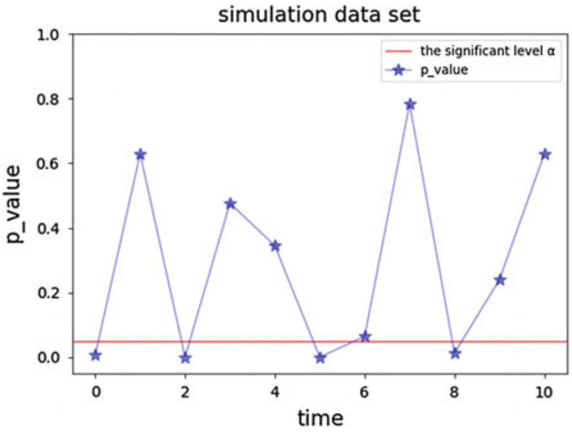


Fig. 1. Simulation data result

In terms of radar data that contains some noise, by comparing Fig. 2 and result plots, we can see that errors occurred at 5 and 7 when testing the significance level, but as for its overall results, the K–S test can accurately detect whether there is concept drift between adjacent data blocks. And when α is 0.05, the result is highly reliable, therefore K–S detection can be used to detect whether concept drift occurs in the data stream.

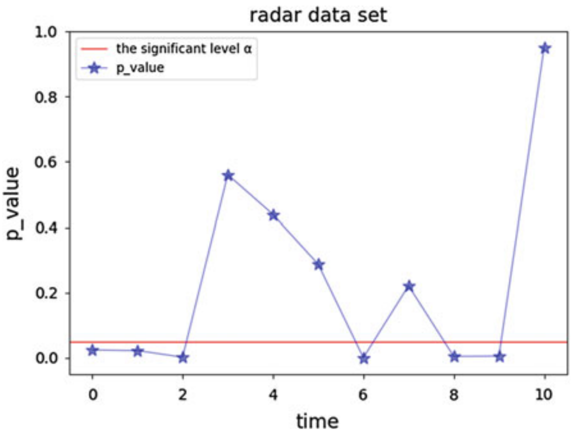


Fig. 2. Radar data result

4 Conclusions

By comparing the above two experiments, we can clearly see that in the case of stable data stream, the model based on K–S test can effectively detect whether there are obvious differences between the two data blocks. It is also effective to be used to detect whether concept drift occurs between blocks. Most of all, the results also prove it, thus the model algorithm can be used to detect the concept drift.

Acknowledgements. This paper is supported by Natural Youth Science Foundation of China (61501326), the National Natural Science Foundation of China (61731006).

References

1. Stiglic G, Kokol P (2011) Interpretability of sudden concept drift in medical informatics domain. In: 2011 IEEE 11th international conference on data mining workshops, pp 609–613
2. Sun J, Li H, Adeli H (2013) Concept drift-oriented adaptive and dynamic support vector machine ensemble with time window in corporate financial risk prediction. IEEE Trans Syst Man Cybern: Syst 43(4):801–813
3. Gama J, Medas P, Castillo G (2004) Learning with drift detection. In: Brazilian symposium on artificial intelligence, pp 286–295
4. Baena-Garcıa M, del Campo-Ávila J, Fidalgo R (2006) Early drift detection method. In: Fourth international workshop on knowledge discovery from data streams, pp 77–86
5. Vorburger P, Bernstein A (2006) Entropy-based concept shift detection. pp. 113–118
6. Ross GJ, Adams NM, Tasoulis DK et al (2012) Exponentially weighted moving average charts for detecting concept drift. Pattern Recogn Lett 33(16): 191–198
7. Frias-Blanco I, Campo-Avila JD, Ramos-Jimenez G et al (2015) Online and non-parametric drift detection methods based on hoeffding's bounds. IEEE Trans Knowl Data Eng 27 (3):810–823

8. Farid DM, Li Z, Hossain A et al (2013) An adaptive ensemble classifier for mining concept drifting data streams. Expert Sys Appl 40(15):5895–5906
9. Ada I, Berthold MR (2014) EVE: a framework for event detection. Evolving Sys 4(1):1–10
10. Sakthithasan S, Pears R, Yun SK (2013) One pass concept change detection for data streams. Adv Knowl Discov Data Min
11. Pears R, Sakthithasan S, Yun SK (2014) Detecting concept change in dynamic data stream. Mach Learn 97(3):259–293

Millimeter-Wave Beamforming of UAV Communications for Small Cell Coverage

Weizhi Zhong[1(✉)], Lei Wang[1], Qiuming Zhu[2(✉)], Xiaomin Chen[2],
and Jianjiang Zhou[2]

[1] College of Astronautics, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China
zhongwz@nuaa.edu.cn

[2] College of Electronic and Information Engineering, NUAA,
Nanjing 210016, China
zhuqiuming@nuaa.edu.cn

Abstract. The unmanned aerial vehicle (UAV) communications in the millimeter-wave (mmWave) band have found a wide range of concerns recently. In order to improve the communication capacity of UAV cellular networks, a low-complexity beam optimization method for the hybrid beamforming system with uniform planar arrays (UPAs) is proposed in this paper. First, the target cellular cell is quantified in spatial domain and the equivalent channel model of quantified region is established. Then, the data rate is formulated and an ideal precoding vector is achieved in terms of the expected beam gain. At last, a hybrid precoding method based on dynamic dictionary learning orthogonal matching pursuit (DDL-OMP) algorithm is introduced for producing an optimized beam. Simulation results demonstrate that our proposed method outperforms the traditional ones in achieving considerable capacity with faster convergence speed.

Keywords: UAV · mmWave communications · UPA · Beamforming · Beam coverage

1 Introduction

UAVs have found a wide range of applications during recent years, i.e., environmental monitoring, post-disaster rescue, and communication relay [1, 2]. Under these scenarios, a UAV acts as an aerial base station (BS) and serves multiple ground users. Due to the mobility of UAVs, the service quality of cellular cells can be improved and the business of hotspot communities can be diverted [3].

The mmWave band has been adopted in UAV communications to achieve larger capacity, but the severe path loss restricts its practical applications [4]. In order to overcome this shortcoming, mmWave communication systems are usually equipped with large-scale array antennas for beamforming [5].

Different from traditional terrestrial communications, UAV-based communication systems face some new challenges. For example, more efficient beamforming training

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 281–288, 2020.
https://doi.org/10.1007/978-981-15-0187-6_32

and tracking are needed to accommodate the movement of UAVs [6]. Good beam shape is also desired to improve the data rate of covered region.

In [7], a novel capacity enhancement method for small cells utilizing the flexible three-dimensional (3D) beamforming facilitated by adoption of the active antenna system (AAS) at base stations (BSs) is proposed. In contrast to conventional microcell networks, significant gain of capacity has been achieved by using this novel method. However, this dynamic and flexible 3D beamforming with narrow beamwidth is only feasible in the small cell layer. In [8], the authors took the coverage and rate performance of mmWave cellular networks into account, which leads to higher peak rates while didn't improve the cell edge rates. To guarantee a good tradeoff between the rates of center and edge, a novel wide beam design method combined with orthogonal matching pursuit (OMP) algorithm was proposed in [9]. The work [10] proposed an optimized beam by combining the quantification of spatial angle and the improved OMP algorithm. Literature [11] proposed solution relies on the gradient pursuit (GP) algorithm to fast approximation the optimal digital precoding. However, all the methods mentioned above had high system complexity.

This paper aims to fill these gaps. The major contributions and novelties of this paper are summarized as follows:

(1) To achieve better beam coverage performance, the expression of the equivalent channel is derived by using the principle of singular value decomposition (SVD).
(2) To obtain the ideal hybrid precoding vector according to the expected beam gain, the target region is quantified and the sparse state matrix is established.
(3) A precoding algorithm based on DDL-OMP is presented. The new method can reduce the system complexity while improving the communication capacity.

2 System Model

In this paper, a downlink UAV-BS with mmWave hybrid beamforming system is considered. Both the transmitter and receiver are equipped with UPA of $M_{\mathrm{MS}} = M_{\mathrm{BS}} = M_h \times M_v$ antennas. The array is laid out in a grid pattern with M_h columns and M_v rows [9], where the distance between antenna elements is spaced uniformly with separations d_h and d_v, respectively [12].

A block fading channel is adopted and the received signal after processing at the receiver can be expressed as

$$y = \sqrt{\rho}\boldsymbol{w}^H\mathbf{H}\boldsymbol{c}s + \boldsymbol{w}^H\boldsymbol{n} \tag{1}$$

where ρ is the average receiving power, $\boldsymbol{c} = \mathbf{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}} \in C^{M\times 1}$ is the beamformer of the transmitter formed by a combination of a RF precoding matrix $\mathbf{F}_{\mathrm{RF}}=[\boldsymbol{f}_1,\boldsymbol{f}_2,\ldots,\boldsymbol{f}_{M_{\mathrm{RF}}}] \in C^{M\times M_{\mathrm{RF}}}$ and a baseband precoding vector $\boldsymbol{v}_{\mathrm{BB}}$. Similarly, $\boldsymbol{w}$ is the combiner of the receiver and it can be written in a similar form. $\mathbf{H} \in C^{M\times M}$ is the mmWave channel matrix, s is the transmit symbol subject to the constraint of $E[|s|^2] \leq 1$, and $\boldsymbol{n}$ is the additive white Gaussian noise (AWGN) with zero mean and variance $\sigma^2 = 1$. In this paper, we assumed that both of the transmitter and receiver have the same RF chains which is limited due to the constraints of the system cost and power consumption.

The mmWave channel between UAVs can be modeled as [13]

$$\mathbf{H} = \sqrt{\frac{M_{\mathrm{MS}}M_{\mathrm{BS}}K}{1+K}}\alpha_0\mathbf{a}_{\mathrm{MS}}(\theta_h,\theta_v)\mathbf{a}_{\mathrm{BS}}^H(\psi_h,\psi_v) \tag{2}$$

where M_{MS} and M_{BS} denote the antenna numbers of receiver and transmitter, respectively. K is the Ricean factor, and α_0 is the complex channel gain. Moreover, $\mathbf{a}_{\mathrm{BS}}(\psi_h,\psi_v)$ is the array response vector of transmitter, which can be described as

$$\mathbf{a}_{\mathrm{BS}}(\psi_h,\psi_v) = \mathbf{a}_{\mathrm{BS}M_h}(\psi_h)\otimes\mathbf{a}_{\mathrm{BS}M_v}(\psi_v) \tag{3}$$

where $\mathbf{a}_{\mathrm{BS}M_h}(\psi_h)$ and $\mathbf{a}_{\mathrm{BS}M_v}(\psi_v)$ can be given by

$$\begin{aligned}\mathbf{a}_{\mathrm{BS}M_h}(\psi_h) &= \frac{1}{\sqrt{M_h}}\left[1, e^{jkd_h\sin\psi_h\cos\psi_v},\ldots,e^{jk(M_h-1)d_h\sin\psi_h\cos\psi_v}\right]^T\\ \mathbf{a}_{\mathrm{BS}M_v}(\psi_v) &= \frac{1}{\sqrt{M_v}}\left[1, e^{jkd_v\sin\psi_v},\ldots,e^{jk(M_v-1)d_v\sin\psi_v}\right]^T\end{aligned} \tag{4}$$

where $k = 2\pi/\lambda$, λ is the signal wavelength, d_h and d_v are both set as $\lambda/2$, $\psi_{h(v)}$ is the angle of departure (AoD). h and v denote horizontal and vertical domains, respectively. $\mathbf{a}_{\mathrm{MS}}(\theta_h,\theta_v)$ is the array response vectors of receiver and it can be written in a similar form with (4).

3 Improved Beam Design

3.1 Equivalent Spectral Efficiency

In order to evaluate the performance of beam, we need to derive the expression of equivalent spectral efficiency. Based on the mmWave system model, the spectral efficiency achieved can be expressed by [14]

$$R = \log_2\left(\left|\mathbf{I} + \rho\mathbf{R}_n^{-1}\boldsymbol{w}_{\mathrm{BB}}^H\mathbf{W}_{\mathrm{RF}}^H\mathbf{H}\mathbf{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}}\boldsymbol{v}_{\mathrm{BB}}^H\mathbf{F}_{\mathrm{RF}}^H\mathbf{H}^H\mathbf{W}_{\mathrm{RF}}\boldsymbol{w}_{\mathrm{BB}}\right|\right) \tag{5}$$

where $\mathbf{R}_n = \sigma^2\boldsymbol{w}^H\boldsymbol{w}$ is the noise covariance matrix. We focus on the design of precoder/combiner to improve spectral efficiency. However, it is difficult to obtain global optimal value. To simplify this problem, we design the precoding vector $\boldsymbol{c} = \mathbf{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}}$ by using singular value decomposition (SVD) of channel as $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^H$. Then, (5) can be simplified as

$$R = \log_2(1+\rho\left|\boldsymbol{h}^H\boldsymbol{c}\right|^2) \tag{6}$$

where $\boldsymbol{h} = \boldsymbol{\Sigma}_1\mathbf{V}_1$ is the equivalent channel state coefficient.

3.2 Optimal Beam Pattern

Since the receiver has the same structure, we only focus on $\boldsymbol{c}_{\text{opt}}$. According to [9], the reference gain can be defined as

$$G(\psi_{h_i}, \psi_{v_j}, \boldsymbol{c}) = \boldsymbol{c}^H \boldsymbol{h}(\psi_{h_i}, \psi_{v_j}) \tag{7}$$

where $\boldsymbol{h}(\psi_{h_i}, \psi_{v_j})$ is the equivalent array response corresponding to the quantized angles (ψ_{h_i}, ψ_{v_j}). The ideal beam should have a non-zero constant reference gain in the coverage region $U_{m,n}$ and zero reference gain for the rest of area. In the condition, the reference gain $G(\psi_{h_i}, \psi_{v_j}, \boldsymbol{c})$ can be expressed as

$$G(\psi_{h_i}, \psi_{v_j}, \boldsymbol{c}) = \begin{cases} b & \text{if } (\psi_{h_i}, \psi_{v_j}) \in U_{m,n} \\ 0 & \text{if } (\psi_{h_i}, \psi_{v_j}) \notin U_{m,n} \end{cases} \tag{8}$$

where b is a non-zero constant. The beam precoding vector $\boldsymbol{c}$ can be rewritten as the solution of the equation

$$\boldsymbol{D}^H \boldsymbol{c} = \eta \boldsymbol{G}_{(Z_h, Z_v)} \tag{9}$$

where η is a normalized constant that satisfies $\|\boldsymbol{c}\|_2 = 1$, $\boldsymbol{D} \in \boldsymbol{C}^{M \times (Z_h \times Z_v)}$ is a set of equivalent channel state coefficient corresponding to the quantized angles (ψ_{h_i}, ψ_{v_j}), which can be expressed as

$$\boldsymbol{D} = \left[\boldsymbol{h}(\psi_{h_1}, \psi_{v_1}), \boldsymbol{h}(\psi_{h_1}, \psi_{v_2}), \ldots, \boldsymbol{h}(\psi_{h_{Z_h}}, \psi_{v_{Z_v}}) \right] \tag{10}$$

Meanwhile, the set of reference gain $\boldsymbol{G}_{(Z_h, Z_v)} \in \boldsymbol{C}^{(Z_h \times Z_v) \times 1}$ also can be written as

$$\boldsymbol{G}_{(Z_h, Z_v)} = \left[G(\psi_{h_1}, \psi_{v_1}, \boldsymbol{c}), G(\psi_{h_1}, \psi_{v_2}, \boldsymbol{c}), \ldots, G(\psi_{h_{Z_h}}, \psi_{v_{Z_v}}, \boldsymbol{c}) \right]^T \tag{11}$$

where $\boldsymbol{G}_{(Z_h, Z_v)}$ is a sparse matrix. Thus, the optimal precoding vector can be obtained by

$$\boldsymbol{c}_{\text{opt}} = \eta (\boldsymbol{D}\boldsymbol{D}^H)^{-1} \boldsymbol{D} \boldsymbol{G}_{(Z_h, Z_v)} \tag{12}$$

3.3 Beam Design Based on DDL-OMP Algorithm

As mentioned above, the optimal digital precoder can be achieved by (12). However, in actual beamforming system, the RF terminals cannot directly generate $\boldsymbol{c}_{\text{opt}}$. It is necessary to approximate the value with RF matrix $\boldsymbol{F}_{\text{RF}}$ and baseband vector $\boldsymbol{v}_{\text{BB}}$, thus the problem of precoder design can be expressed as

$$(\boldsymbol{F}_{\mathrm{RF}}^{\mathrm{opt}}, \boldsymbol{v}_{\mathrm{BB}}^{\mathrm{opt}}) = \arg\min \left\| \boldsymbol{c}_{\mathrm{opt}} - \boldsymbol{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}} \right\|_F$$
$$\boldsymbol{F}_{\mathrm{RF}} \in \boldsymbol{f}_{\mathrm{RF}}, \quad \|\boldsymbol{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}}\|_F^2 = 1 \tag{13}$$

where $\boldsymbol{f}_{\mathrm{RF}}$ is the set of feasible RF precoders, i.e., the set of $M_{\mathrm{BS}} \times M_{\mathrm{RF}}$ matrices with constant magnitude entries. Each column $\boldsymbol{f}_{\mathrm{RF}}$ is obtained from a dictionary set of $\boldsymbol{\kappa}_Q$, which can be defined as

$$\boldsymbol{\kappa}_Q = [\mathbf{a}_{\mathrm{BS}}(\psi_{h,1}, \psi_{v,1}), \ldots, \mathbf{a}_{\mathrm{BS}}(\psi_{h,Q_h}, \psi_{v,Q_v})] \tag{14}$$

where Q_h and Q_v are the quantization times in the corresponding direction for a single beam coverage area. When each phase shifter is controlled by 2^B, the elements of the matrix can be written as $e^{jk_Q 2\pi/2^B}$ for $k_Q = 0, 1, 2, \ldots, 2^B - 1$.

Unfortunately, the complex non-convex nature of feasible set $\boldsymbol{f}_{\mathrm{RF}}$ makes the problem of finding such a projection both analytically and algorithmically intractable. In order to provide near-optimal solutions to the problem in (13) and reduce the system complexity, a precoding method based on DDL-OMP algorithm [15] is introduced in this paper. The process of the algorithm is shown in Table 1.

Table 1. Spatial sparse precoding based on DDL-OMP algorithm

Stage	Operation
1	$\boldsymbol{F}_{RF} = [\ ]$
2	$\boldsymbol{F}_{\mathrm{res}} = \boldsymbol{c}_{\mathrm{opt}}$
3	$\boldsymbol{\kappa}_Q = [\boldsymbol{\kappa}_Q, \tau(\boldsymbol{c}_{\mathrm{opt}})]$
4	**for** $i \leq M_{\mathrm{RF}}$ **do**
5	$\mathrm{T} = \boldsymbol{\kappa}_Q^H \boldsymbol{F}_{\mathrm{res}}$
6	$t = \underset{l=1,2,\ldots,B}{\arg\max} (\boldsymbol{T}\boldsymbol{T}^*)_{l,l}$
7	$\boldsymbol{F}_{\mathrm{RF}} = [\boldsymbol{F}_{\mathrm{RF}} \quad \boldsymbol{\kappa}_Q(:,t)]$
8	$\boldsymbol{v}_{\mathrm{BB}} = (\boldsymbol{F}_{\mathrm{RF}}^H \boldsymbol{F}_{\mathrm{RF}})^{-1} \boldsymbol{F}_{\mathrm{RF}}^H \boldsymbol{c}_{\mathrm{opt}}$
9	$\boldsymbol{F}_{\mathrm{res}} = \boldsymbol{c}_{\mathrm{opt}} - \boldsymbol{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}}$
10	$\boldsymbol{\kappa}_Q(:,t) = \tau(\boldsymbol{F}_{\mathrm{res}})$
11	**end for**
12	$\boldsymbol{F}_{\mathrm{RF}} = \boldsymbol{F}_{\mathrm{RF}},\ \boldsymbol{v}_{\mathrm{BB}} = \frac{\boldsymbol{v}_{\mathrm{BB}}}{\|\boldsymbol{F}_{\mathrm{RF}}\boldsymbol{v}_{\mathrm{BB}}\|_F}$

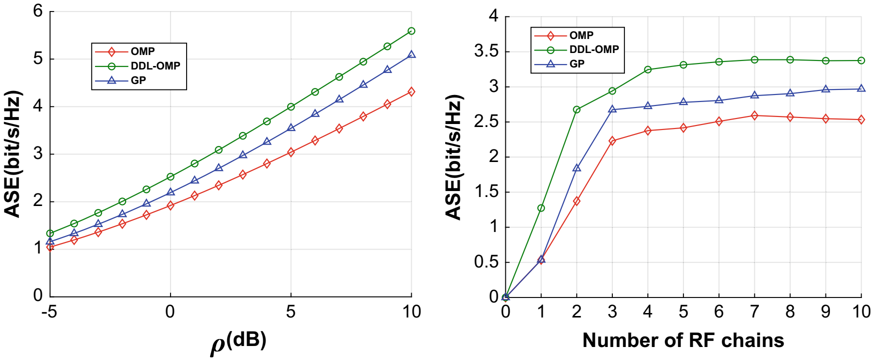
4 Simulation

In this section, the performance of our proposed beam design method is verified by using numerical simulations. In the simulations, the carrier frequency is $f_c = 60\,\mathrm{GHz}$. The phase shifter of beam is controlled by $B = 3$ and the Ricean factor K is set as 13.5 dB [9]. It is remarkable that we adopt a computer of AMD A4-5300B APU with Radeon(tm) HD Graphics 3.40 GHz to carry out all of the simulations.

4.1 Average Spectrum Efficiency

In order to assess the communication quality of the coverage area, the average spectrum efficiency (ASE) indicator is adopted. The simulation parameters are set as $M = 12 \times 8$, $Z_h = 8$, $Z_v = 6$, $M_{\mathrm{RF}} = 8$, and $Q_h = Q_v = 8$.

As shown in Fig. 1a, for the same number of RF chains, our proposed beam design method can obtain highest ASE compared to the other methods. The reason is that our proposed method has approximately uniform gain in the coverage region which is the characteristic of optimal beam.



(a) Relationship between the ASE and the SNR

(b) Relationship between the ASE and the number of RF chains

Fig. 1. Comparison of ASE

The ASEs of different methods with different values of RF chains are also compared in Fig. 1b, it can be observed that the convergence speed of the proposed method is faster than others for the same signal-to-noise ratio (SNR). The reason is that the DDL-OMP based method adopts a dynamic learning method to update the dictionary set by continuously approximating the optimal solution.

4.2 Iterative Efficiency and Complexity

To evaluate the iterative efficiency of different methods, the ratio of the residual to the ideal precoding vector is used as an indicator of the convergence speed, which can be expressed as

$$\eta = \frac{\|\boldsymbol{F}_{\mathrm{res}}\|_2}{\|\boldsymbol{c}_{\mathrm{opt}}\|_2} \tag{15}$$

where $\boldsymbol{F}_{\mathrm{res}}$ is the residual corresponding to the number of iterations. The simulation parameters are set as $M = 8 \times 8$, $Z_h = 8$, $Z_v = 6$, and $Q_h = Q_v = 8$.

By using (15), the convergence speed is calculated and shown in Table 2. As we can see that as the number of iterations increases, the proposed method has higher convergence speed and better iterative performance than the traditional methods of OMP and GP. When achieving the same iteration accuracy, our method requires much fewer RF chains, which means that it can greatly reduce the system complexity.

Table 2. Comparison of iterative efficiency

M_{RF}	OMP	GP	DDL-OMP
1	0.8312	0.8288	0.7296
2	0.7700	0.7093	0.3910
3	0.6619	0.6354	0.2617
4	0.5990	0.5692	0.1596
5	0.5365	0.5067	0.1119
6	0.5151	0.4633	0.0684
7	0.4969	0.4314	0.0484
8	0.4867	0.4033	0.0352
9	0.4774	0.3771	0.0265
10	0.4694	0.3566	0.0204

5 Conclusion

In this paper, a novel beam design method for mmWave based UAV cellular networks has been proposed. In the method, the SVD decomposition of channel and the quantization of the target area are introduced to achieve the ideal precoding vector. On this basis, a DDL-OMP algorithm is proposed to obtain the final optimized beam. Simulation results have shown that our proposed method outperforms traditional schemes in achieving considerable capacity with lower system complexity. Although the new method has improved the communication quality while reducing the system overhead, some problems such as complex inversion calculation at the iterative process of the algorithm, need further research.

Acknowledgements. This work was supported by Aeronautical Science Foundation of China (2017ZC52021), the Major Program of National Natural Science Foundation of China (61827801), and the Foundation of Graduate Innovation Center in NUAA (kfjj20181503).

References

1. Xiao Z, Xia P, Xia XG (2016) Enabling UAV cellular with millimeter-wave communication: potentials and approaches. IEEE Commun Mag 54(5):66–73
2. Zeng Y, Zhang R, Lim TJ (2016) Wireless communications with unmanned aerial vehicles: opportunities and challenges. IEEE Commun Mag 54(5):36–42
3. Shen X (2018) Research on coverage and resource allocation of UAV mounted airborne base station. Beijing University of Posts and Telecommunications, Beijing

4. Yang J, Ai B, Guan K et al (2018) A geometry-based stochastic channel model for the millimeter-wave band in a 3GPP high-speed train scenario. IEEE Trans Veh Technol 67 (5):3853–3864
5. Alkhateeb A, Mo J, Gonzalez P-N et al (2014) MIMO precoding and combining solutions for millimeter-wave systems. IEEE Commun Mag 52(12):122–131
6. Li B, Zhou Z, Zou W et al (2013) On the efficient beam-forming training for 60 GHz wireless personal area networks. IEEE Trans Wireless Commun 12(2):504–515
7. Yu B, Yang L, Ishii H (2014) 3D beamforming for capacity improvement in macrocell-assisted small cell architecture. In: IEEE global communications conference. IEEE Press, Austin, pp 4833–4838
8. Kulkarni M-N, Singh S, Andrews J-G (2014) Coverage and rate trends in dense urban mmWave cellular networks. In: IEEE global communications conference. IEEE Press, Austin, pp 3809–3814
9. Song J, Choi J, Love D-J (2017) Common codebook millimeter wave beam design: designing beams for both sounding and communication with uniform planar arrays. IEEE Trans Commun 65(4):1859–1872
10. Zhong WZ, Xu L, Zhu Q et al (2019) mmWave beamforming for UAV communications with unstable beam pointing. China Commun 16(1):37–46
11. Kaushik A, Thompson J, Yaghoobi M (2016) Sparse hybrid precoding and combining in millimeter wave MIMO systems. In: Radio propagation and technologies for 5G. IET Press, Durham, pp 1–7
12. Hansen R-C (2009) Phased array antennas, 2nd edn. Wiley, Hoboken, NJ
13. Rappaport T-S, Ben-Dor E, Murdock J-N et al (2012) 38 GHz and 60 GHz angle-dependent propagation for cellular & peer-to-peer wireless communications. In: IEEE international conference on communications. IEEE Press, Ottawa, pp 4568–4573
14. Ayach O-E, Rajagopal S, Abu-Surra S et al (2013) Spatially sparse precoding in millimeter wave MIMO systems. IEEE Trans Wireless Commun 13(3):1499–1513
15. Donno D, Joan P-B, Giustiniano D et al (2016) Hybrid analog–digital beam training for mmWave systems with low-resolution RF phase shifters. In: IEEE international conference on communications workshops. IEEE Press, Kuala Lumpur, pp 700–705

The Feasibility Analysis of the Application of Ensemble Learning to Operational Assistant Decision-Making

Xiang Wei and Xueman Fan(✉)

Navy Submarine Academy, Qingdao 266199, China
oucfanxm@163.com

Abstract. It is urgent to develop artificial intelligence technology and extend the brain of warship commanders with intelligent machines, thus assisting battlefield situation cognition and decision making. Since the 1990s, ensemble learning has become a new research focus in the field of machine learning. Ensemble learning can improve the generalization ability of the classification algorithms, and achieve the effect of “1 + 1 > 2”. Whether the performance bottleneck of single classifier can be broken by ensemble learning, so as to improve the ability of operational assistant decision of warship? This is a topic worth to discuss. In this paper, we focus on whether the torpedo can find the warship after the warship launched a torpedo decoy, and transform it into the two classification problem in machine learning. First, the classification data set is generated through offline simulation. Secondly, select decision trees as base classification method and Bagging as the typical of ensemble learning, and then the performance improvement of ensemble learning over single classifier is analyzed under ideal conditions. Finally, label noise, small training dataset size is taken into account, and the comparison experiments between ensemble learning and single classifier are performed further. The experimental results verify the feasibility of applying the ensemble learning to operation assistant decision.

Keywords: Operational assistant decision-making · Decision tree · Ensemble learning · Bagging

1 Introduction

For warships, the threat mainly comes from heavy torpedoes such as self-guided torpedoes, wire-guided torpedoes, air-dropped torpedoes, and rocket-assisted torpedoes. Self-propelled acoustic decoy can not only simulate the radiated noise and acoustic reflection characteristics of warships but also mimic the motion characteristics of warships, which has great deception to torpedoes. Using self-propelled acoustic decoy to defend acoustic homing torpedo has become one of the main means of warship underwater defense. Self-propelled acoustic decoy has become one of the most important equipments for warship underwater defense. After the warship launches the decoy, it is very important for the subsequent decision-making of the warship to predict whether the decoy can successfully exert the jamming effect according to the relative

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 289–297, 2020.
https://doi.org/10.1007/978-981-15-0187-6_33

situation of warship, decoy and torpedo, that is, whether the torpedo can discover the warship. In this paper, the problem is transformed into a binary classification scenario by machine learning, and the feasibility of ensemble learning in warship operational assistant decision-making is studied.

Ensemble learning refers to using simple, trainable and differentiated classifiers to achieve accurate classification decisions through integrated voting [1]. Since ensemble learning was first proposed to solve classification problems, and most of the ensemble learning algorithms are used to design classifiers, thus ensemble learning is also called classifier ensemble or multi-classifier system and so on. Both the classification accuracy can be improved and the risk of relying on a single classifier can be reduced significantly by means of classifier ensemble. Therefore, it has been widely used in bionics, finance, politics, and medicine [2]. But is it feasible to apply the ensemble learning algorithm which is superior in other fields directly to the operational assistant decision-making? As the "no free lunch" theorem states [3], there are many ensemble learning algorithms, but there is no universal algorithm suitable for all data sets. It is necessary to study the adaptability of ensemble learning to warship operational assistant decision-making.

2 Data Set Description

The decoy jamming effect data set is generated by off-line simulation. In the simulation process, the relative situation of warship, torpedo, and decoy can be described as five parameters: the distance between torpedo and warship, the distance between decoy and warship, the azimuth angle between decoy and warship, the azimuth angle between torpedo and warship and the azimuth angle between warship and torpedo, which are, respectively, denoted as x_1, x_2, x_3, x_4, x_5. In this way, each sample is represented with a five-dimensional vector, i.e., $\boldsymbol{x} = (x_1, x_2, \ldots, x_5)$. The label y of sample $\boldsymbol{x}$ is 1 or −1, $y = -1$ means that in the situation corresponding to $\boldsymbol{x}$, the torpedo cannot find the warship and vice versa.

In the simulation, each feature takes 10 values at equal intervals, as a result, 10^5 samples are generated for the training and testing of the ensemble classifier.

3 Algorithms Description

3.1 Single and Ensemble Classification Algorithms

In order to verify the effectiveness of ensemble learning, it is necessary to compare the ensemble classification algorithm with typical classification algorithm. Decision tree is one of the most widely used algorithms in the field of data mining. Therefore, the decision tree was selected for comparative experiments.

In terms of ensemble learning, the Bagging algorithm [4] was selected for comparison, which used the reduced error pruning tree (REPTree) [5] as the base classification algorithm. Consider a classification problem with a set $\Omega = \{\omega_1, \omega_2, \ldots, \omega_C\}$ of class labels and feature space $X \subseteq R^n$. Let a pool of base classifiers $\boldsymbol{C} = \{C_1, C_2, \ldots, C_L\}$, called a classifier ensemble, be given and let a classifier C_i, $i = 1, 2, \ldots, L$ be a function C_i: $X \rightarrow \Omega$. When the "soft label" is used, the output of classifier C_i can be represented as an M-dimensional vector [6]

$$C_i(\boldsymbol{x}) = [d_{i,1}(\boldsymbol{x}), d_{i,2}(\boldsymbol{x}), \ldots, d_{i,M}(\boldsymbol{x})]^T \tag{1}$$

where $d_{i,j}(\boldsymbol{x})$ is the degree of support given by classifier C_i to the hypothesis that $\boldsymbol{x}$ comes from class j. Most often $d_{i,j}(\boldsymbol{x})$ is an estimation of the posterior probability $P(j|\boldsymbol{x})$, and without loss of generality $\sum_j d_{i,j}(\boldsymbol{x}) = 1$. Classification is made according to the maximum rule

$$C_i(\boldsymbol{x}) = \omega_s = s = \arg\max_j d_{i,j}(\boldsymbol{x}) \tag{2}$$

When T ($T \leq L$) classifiers are selected to participate in decision-making, the final result is obtained through a combination function such as minimum, maximum, average and product.

3.2 Confidence Evaluation Based on Non-parametric Estimation of Probability Density

In order to further improve the prediction accuracy of the algorithm, the rejection ability is required. In this paper, a confidence evaluation method based on nonparametric estimation of probability density is used, which can quantify the classification confidence, and reject low confidence samples by setting a threshold.

A good confidence metric (CM) has the following attributes:

$$\begin{cases} \mathrm{CM}(H_i) \geq 0 \\ \mathrm{CM}(H_i \cup H_j) = \mathrm{CM}(H_i) + \mathrm{CM}(H_j) \\ \sum_i \mathrm{CM}(H_i) = 1 \end{cases} \tag{3}$$

where H_i is the hypothesis that the sample belongs to ith class, and $\mathrm{CM}(H_i)$ is the confidence of hypothesis H_i.

In this paper, we use the method in literature [7] to evaluate the classification confidence, and the final assumption can be calculated by the following formula,

$$H_s \Leftrightarrow s = \arg\max_i \{\mathrm{CM}(H_i)\}, \quad i = 1,\ 2 \tag{4}$$

Set the threshold to TH, if $\mathrm{CM}(H_s) \geq$ TH, the test sample is classified as sth class, otherwise, the test sample will be rejected. The classification algorithm will have rejection ability by embedding the above process.

4 Experiments

This experiment is implemented on IDEA Java platform with the help of Weka API. Concretely, Java compiler environment is JDK 8 and data mining software is Weka 3.9 [8]. The REPTree in Weka is selected as the typical classification algorithm for comparison and the basic classification algorithm of ensemble learning. In the experiments, REPTree adopts the post pruning strategy [9] and the branching criterion based on information gain. The number of base classifiers of the Bagging algorithm is 50, and the other parameters are set by default in Weka.

In this section, we present four experiments. First, the comparison experiment of Bagging and REPTree under ideal conditions. Second, the comparison experiment under the influence of noise. Third, the comparison experiment under the influence of small sample set. Finally, the comparison experiment in consideration of rejection. The data set of the decoy jamming effect generated by simulation is used as experimental data. For each case, five times of two-fold cross-validation experiments were carried out, i.e., two-fold cross-validation was repeated five times. During two-fold cross-validation, the original data set is randomly divided into two equal parts as training set ***TR*** and test set ***TE***. ***TE*** is invisible in training process, and it is specially used to evaluate the accuracy of classification algorithm.

4.1 Comparison Experiment Under Ideal Conditions

Ideal conditions represent that there is no object outside the library and it is not affected by noise sample and small sample set. The confusion matrix is used to study the separability between targets. The concrete result of the classification algorithm can be calculated from the confusion matrix. The confusion matrices of the two classification algorithms are shown in Tables 1 and 2 respectively.

Table 1. The confusion matrix of REPTree under ideal conditions (%)

True	Predicted	
	−1	1
−1	99.33	0.66
1	0.17	99.83

Table 2. The confusion matrix of Bagging under ideal conditions (%)

True	Predicted	
	−1	1
−1	99.45	0.55
1	0.13	99.87

It can be seen that both of the two algorithms can predict the effect of decoy jamming considerably accurately. The average accuracy of Bagging and REPTree were 99.66% and 99.58%, respectively. Thus, the classification performance of Bagging is slightly better than that of REPTree under ideal conditions.

4.2 Comparison Experiment Under the Influence of Noise

In this section, the noise is injected into the training data, namely, a certain proportion of the class labels in the data are flipped to other randomly selected values to study the influence of class noise on the performance of the two classification algorithms. It should be noted that only the training data are added with noise, while the test data are not affected. Considering 11 different cases of noise ratio, i.e., 0%, 10%, …, 100%. For each case, five times of two-fold cross-validation experiments were performed to study the influence of label noise. The relationship between the classification accuracy and the noise ratio was shown in Fig. 1, where the dotted line represents the accuracy of random guess, namely 50%.

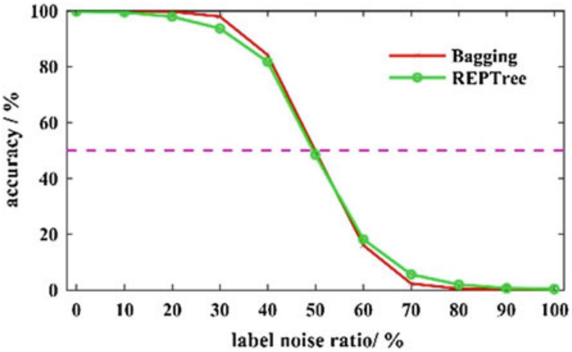


Fig. 1. Classification accuracy under different noise ratios

As can be seen from Fig. 1, when the noise ratio is less than 30%, the classification accuracy of the two algorithms decreases slowly with the increase of noise ratio, and Bagging has considerable advantages over REPTree. When the noise ratio is more than 30%, the classification accuracy of the two algorithms will deteriorate sharply with the increase of noise ratio. When the noise ratio increases to 60%, the classification accuracy of REPTree is equivalent to that of Bagging. When the noise ratio exceeds 60%, although REPTree is slightly dominant, the accuracy is far less than that of random guess, so it has no reference value. In a word, the robustness to label noise can be greatly improved by ensemble learning.

4.3 Comparison Experiment Under Influence of Small Sample Set

Small sample set refers to the problem of inadequate training samples. As the same with label noise, small sample set only influences the training stage. Set the deletion ratio of training set is 0%, 10%, …, 99%. We can't take 100% here, because 100% means that there is no training data so the classification model can't be trained. The relationship between the classification accuracy and the deletion ratio was shown in Fig. 2, where the dotted line represents the classification accuracy of random guess, namely 50%. Because when the deletion ratio of training samples exceeds 80%, the classification accuracy drops sharply to less than 50%, which has no reference value. Therefore, we focus on the situation that the deletion ratio is less than 80%, as shown in Fig. 3.

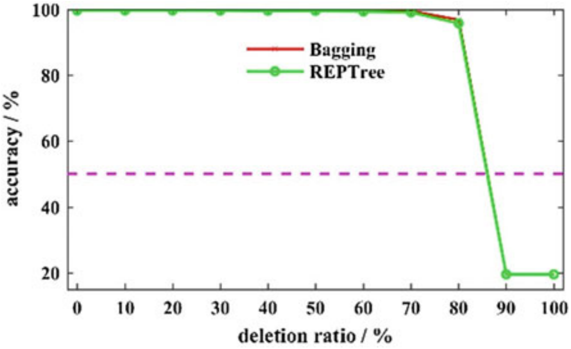


Fig. 2. Classification accuracy under different deletion ratio

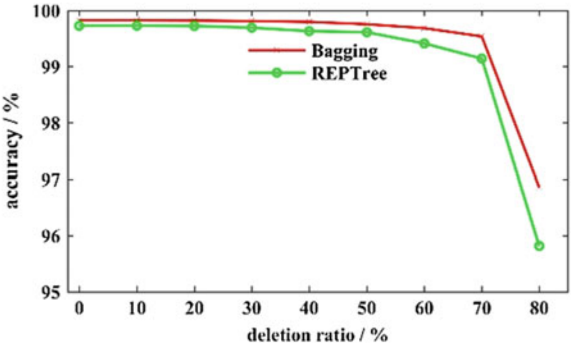


Fig. 3. The detail view of Fig. 2 (0–80%)

As can be seen from Fig. 2, when the deletion ratio is less than 80%, the classification accuracy of Bagging and REPTree changes very slowly. When the deletion ratio exceeds 80%, the classification accuracy will almost decrease linearly with the increase of the deletion ratio, and the trend of the two algorithms is very similar. As can be seen from Fig. 3, the classification accuracy of Bagging is always higher than that of

REPTree. It can be concluded that ensemble learning has some advantages over single classification algorithm in dealing with small sample sets, but advantages are not significant.

4.4 Comparison Experiment with Rejection

When considering rejection, the performance evaluation of classification algorithm becomes more complex. It is not enough to explain the problem comprehensively and deeply only by the classification accuracy. Two indicators are defined as follows, the first is the classification accuracy of the samples to be tested

$$P_{\mathrm{cc}} = 100\% \times N_{\mathrm{cc}}/N_{\mathrm{de}} \tag{5}$$

The second is the judgment rate of the sample to be tested

$$P_{\mathrm{dec}} = 100\% \times N_{\mathrm{dec}}/N_{\mathrm{all}} \tag{6}$$

where N_{all} is the total number of samples to be tested, N_{dec} is number of samples that have been classified; N_{cc} is the number of samples that have classified correctly. There is a compromise between P_{cc} and P_{dec}. Generally, the larger the P_{dec}, the smaller the P_{cc}.

When rejection experiments are carried out, the influences of label noise and small sample set are taken into consideration. Bagging and REPTree are compared with each other under the condition of 30% label noise and 70% deletion ratio of training samples. The variation curves of P_{cc} and P_{dec} with the threshold are shown in Figs. 4 and 5.

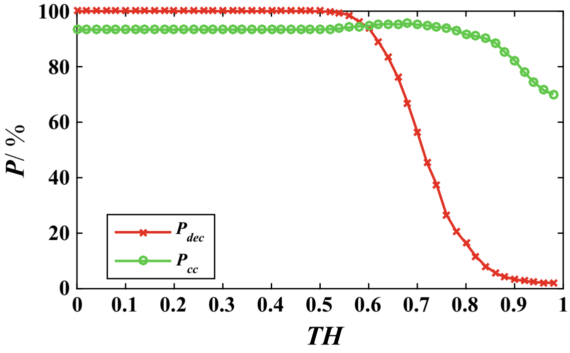


Fig. 4. Variation curves of REPTree

As can be seen from Fig. 4, for REPtree, when TH < 0.71, P_{cc} and $1 - P_{\mathrm{dec}}$ increase with the TH, and when TH $\geq$ 0.71, P_{cc} decreases with the increasing TH. Besides, it can be seen from Fig. 5, for Bagging, when TH < 0.92, P_{cc} and $1 - P_{\mathrm{dec}}$ increase with the TH, and when TH $\geq$ 0.92, P_{cc} decreases with the increasing TH.

In practical application, TH values need to be determined according to relevant requirements, such as controlling P_{cc} or P_{dec} not less than a certain value. For example,

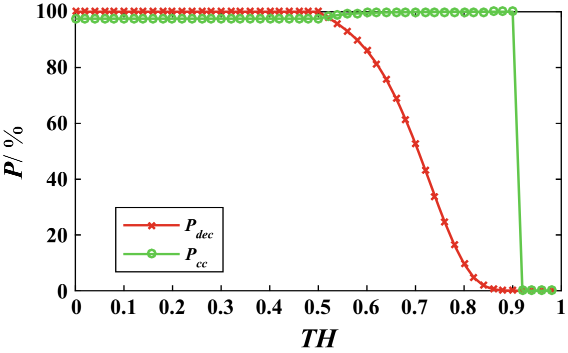


Fig. 5. Variation curves of Bagging

under the condition that P_{dec} is not less than 60% when P_{cc} is required to be as high as possible, the optimal TH of the two classification algorithms and the corresponding P_{cc} and P_{dec} are shown in Table 3. As can be seen from Table 3, Bagging has obvious advantages in P_{cc}.

Table 3. The parameters of the two algorithms ($P_{\mathrm{dec}} > 60\%$)

Algorithm	Parameters		
	TH	P_{dec} (%)	P_{cc} (%)
REPTree	0.69	63.03	95.44
Bagging	0.68	61.23	99.85

5 Conclusions

Based on the prediction of the interference effect of acoustic decoy, the feasibility of ensemble learning in warship operational assistant decision-making is studied in this paper. REPTree and Bagging are selected for comparison, their classification performance under ideal conditions, label noise, and small sample set are studied by experiments. It can be concluded that the ensemble learning is slightly better than the single classification algorithm under ideal conditions. Further, the ensemble learning has advantages in dealing with label noise and small sample set and has obvious advantages in label noise. The results show that ensemble learning is effective in predicting the effect of decoy jamming. Although this problem is only the tip of the iceberg of warship operational assistant decision-making, which can be decomposed into several similar machine learning problems. It can be inferred that it is feasible to apply ensemble learning to operational assistant decision-making.

References

1. Cruz RMO, Sabourin R, Cavalcanti GDC et al (2015) META-DES: a dynamic ensemble selection framework using meta-learning. Pattern Recognit 48(5):1925–1935
2. Fan XM, Hu SL, He JB (2017) A dynamic selection ensemble method for target recognition based on clustering and randomized reference classifier. Int J Mach Learn Cybern 3:1–11
3. Fokoue E (2015) A taxonomy of big data for optimal predictive machine learning and data mining. Imi Period 113(13):1–18
4. Collell G, Prelec D, Patil KR (2018) A simple plug-in bagging ensemble based on threshold-moving for classifying binary and multiclass imbalanced data. Neurocomputing 275:330–340
5. Belouch M, El S, Idhammad M (2017) A two-stage classifier approach using REPTree algorithm for network intrusion detection. IJACSA 8(6):389–394
6. Fan XM, Hu SL, He JB (2017) A target recognition method for maritime surveillance radars based on hybrid ensemble selection. Int J Syst 48(15):3334–3345
7. Pilcher CM, Khotanzad A (2011) Maritime ATR using classifier combination and high resolution range profiles. IEEE Trans Aerosp Electron Syst 47(4):2558–2573
8. Kotthoff L, Thornton C, Hoos HH et al (2017) Auto-WEKA 2.0: automatic model selection and hyperparameter optimization in WEKA. J Mach Learn Res 18:1–5
9. Kapoor P, Rani R (2015) Efficient decision tree algorithm using J48 and reduced error pruning. Int J Eng Res Gen Sci 3:1613–1621

A Review of Artificial Intelligence for Games

Xueman Fan(✉), J. Wu, and L. Tian

Navy Submarine Academy, Qingdao 266199, China
oucfanxm@163.com

Abstract. Artificial Intelligence (AI) has made great progress in recent years, and it is unlikely to become less important in the future. Besides, it would also be an understatement that the game has greatly promoted the development of AI. Game AI has made a remarkable improvement in about fifteen years. In this paper, we present an academic perspective of AI for games. A number of basic AI methods usually used in games are summarized and discussed, such as ad hoc authoring, tree search, evolutionary computation, and machine learning. Through analysis, it can be concluded that the current game AI is not smart enough, which strongly calls for supports coming from new methods and techniques.

Keywords: Artificial intelligence · Games · Ad hoc authoring · Tree search · Evolutionary computation · Machine learning

1 Introduction

Games are fascinating due to the effort and skills needed from people to complete them. It is the complexity and interestingness nature of games that make them a desirable problem for AI. The difficulty of the games lies in that their finite state spaces, such as the possible strategies for an agent, are usually very large. Furthermore, it is often very hard to assess the goodness of any game state properly. From a computational complexity perspective, many games are NP-hard, meaning that an algorithm for solving a particular game may run for a long time.

Game and AI have a long history, and the research on game AI mainly focuses on the construction of game agents. In academic game AI, we distinguish two main fields and corresponding research activities: board games and video games. Checkers, Chess and Go are classic board games, which were conquered by AI scientists in 1994 [1], 1997 [2] and 2016 [3], respectively. Compared with the board games, characterized by the discrete turn-based mechanics and full state of the game is visible to both players, video games are more challenging. A notable milestone of AI for video games was reached in 2014 when the game AI developed by Google DeepMind played several classic Atari 2600 video games on a superhuman skill level only input from the original pixel [4]. The real-time strategy (RTS) game such as StarCraft has become the next biggest challenge [5]. RTS is a sub-genre of strategy games where players need to build an economy and military power in order to defeat their opponents. From a theoretical point of view, the main differences between RTS games and traditional board games such as Chess are [6]:

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 298–303, 2020.
https://doi.org/10.1007/978-981-15-0187-6_34

- More than one player can take actions at the same time.
- RTS games are "real-time", which means that players have a very short time to decide the next move.
- Most RTS games are partially observable.
- Most RTS games are non-deterministic.
- Last but not least, the complexity of these games, both in terms of state-space size and in terms of number of actions available at each decision cycle is very large. For example, the state space of Go is around 10^{170}, while a typical StarCraft game has at least 10^{1685} possible states [7].

In general, to devise a satisfactory RTS game AI, a series of challenges must be solved, such as opponent modeling, spatiotemporal reasoning, and multi-agent collaboration. These problems can be summarized into three levels: strategy, tactics, and manipulation. Recently, with the rapid development of computing science, researchers invested more effort in designing more intelligent RTS game AI and proposed new intelligent algorithms to raise the game AI's intelligent ability. For the convenience of researchers' reference, basic AI methods that are commonly used in games are summarized and presented. The framework of the survey is illustrated in Fig. 1, which consists of Ad Hoc behavior authoring, tree search algorithms, evolutionary computation, machine learning.

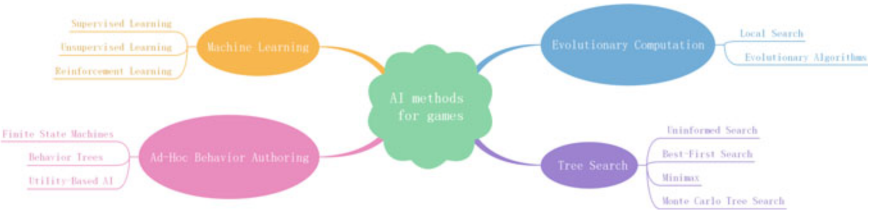


Fig. 1. Main AI methods for game AI

2 Ad Hoc Behavior Authoring

Ad Hoc behavior authoring methods are the most popular kinds of AI methods used to control the non-player characters in games. Finite state machine (FSM), behavior trees (BT), and utility-based AI are three most representative algorithms ad hoc behavior authoring methods.

FSM belongs to the field of expert system and is depicted with graphs where nodes (states) store information about a task, transitions between states represent a state change and actions that need to be executed within each state. The design, implementation, visualization and debugging of FSMs are very simple. However, FSMs have the following shortcomings, i.e., it is hard to cope with a huge game, and it is in lack of flexibility and dynamic characteristics [8].

Similar to FSM, BT also models transitions between a finite set of behaviors [9]. Compared to FSM, BTs are more flexible to design and easier to test due to the modular design concept, which promotes its successful application in games such as Halo 2 and Bioshock.

In order to eliminate the modularity limitations of FSMs and BTs, utility-based AI methods are proposed. Following this method, instances in the game get assigned a particular utility function that gives a value for the importance of the particular instance. While FSMs and BTs check one decision at a time, the utility-based AI methods check all available options, assign utilities to them, and chooses the most appropriate option [10]. In comparison with other ad hoc authoring techniques, utility-based AI has obvious advantages including modularity, extensibility, and reusability. As a result, utility-based methods have been successfully used in Kohan 2, Iron Man, Red Dead Redemption, and Killzone 2.

3 Tree Search Algorithms

It is generally accepted that most, if not all, of AI is actually just search. Almost every AI problem can be transformed into a search problem, which can be solved by finding the best plan, path, model, function, etc. Tree search algorithms can be seen as building a search tree, where the root node is the initial state, the edge of the tree represents the operation taken by the agent from one state to another, and the node represents the state. Tree need to branch because usually several different operations can be performed in a given state. The main differences between tree search algorithms are the branch position and branch order.

It's no exaggeration to say that Monte Carlo Tree Search (MCTS) is the most famous algorithm in the tree search field. MCTS have a long history within numerical algorithms and have also had significant success in various games, particularly imperfect information games such as Scrabble, Bridge and Go. As is known to all, MCTS is an indispensable part of AlphaGo [11]. The core loop of the MCTS algorithm can be divided into four steps: selection, expansion, simulation and backpropagation [4].

It is noteworthy that there are many variants of the basic MCTS methods. The flat UCB presented in [12] treats the leaves of the search tree as a single multi-armed bandit problem. Coquelin and Munos [13] present an algorithm that combines MCTS with TDL using the notion of permanent and transient memories to distinguish the two kinds of state estimation. In [14] the simulation phase of UCT is described as a single agent competing with itself rather than considering the influence of multiple agents.

4 Evolutionary Computation

Randomized optimization algorithms which include several solutions are named Evolutionary Computation (EC), inspired by Darwin's theory of evolution. On the basis of population evolution, where each individual represents a solution with a certain degree of fitness. By selecting and mutating operators, the population gradually evolves, which makes the fitness of current optimal solution more and more higher. EC attracts the attention of game AI because it can generate highly optimized solutions in various problem settings.

Marcolino and Matsubara [15] apply EC in StarCraft to optimize the build-order. Unlikely, Young and Hawes [16] directly apply the genetic algorithm (GA) to optimize more comprehensive strategy for StarCraft, including both quantity and sequence of construction. In Ref. [17], the author takes multi-objective optimization into consideration in order to establish the correct element at a certain time. For more details on multi-objective optimization using EC, see [18]. In [19], the author establishes a forward model to optimize the results of the construction sequence. In [20], EC combines with potential areas to optimize the policy parameters of micromanagement. Both Refs. [21, 22] use neuroevolution to learn micromanagement. They are mainly different in the structure of neural networks. As can be seen from [23], the balance is another factor that researchers and players focus on.

5 Machine Learning

Machine learning can be divided into supervised learning (SL), unsupervised learning (UL) and reinforcement learning (RL). In SL the training signal is provided as sample labels, while in RL it is the reward obtained from the environment. Instead, UL tries to find the correlation of inputs by searching patterns in all input features in the absence of target output as a guide.

Replay data of human can be used as valuable training data for game AI. On the basis of SL, intelligent models can be learned by the means of mining implicit knowledge in data without a large number of manual rules. Therefore, many organizations such as DeepMind and Facebook have released various replay data sets. Literature [24] uses replay data to predict the combat strategies of opponents. In [25], the authors train a probabilistic model to predict the strategy behaviors based on replay data. Deep learning (DL), one of the most popular areas of SL, has been widely used in speech recognition, natural language processing, and other scenarios. Dereszynski et al. [26] use DL to learn macro-management decisions directly from state-action pairs in StarCraft.

RL is good at sequential decision-making tasks. The combination–deep reinforcement learning (DRL), the result of combing DL with RL, can teach agents to make decisions in high-dimension state space by an end-to-end framework [27]. Zhao et al. [28] apply RL to micromanagement of StarCraft, where RL agents need to face many difficult problems including delayed rewards, large action space, and vast state space. Usunier et al. [29] resort to more effective state representation method to decompose the complexity because of the wide state space. Based on SC2LE, Shao et al. [30] propose a new value-based DRL algorithm, which learns decentralized strategies in a centralized end-to-end manner.

Instead of imitating or predicting target values, the intrinsic structure of and associations in the data are concerned for UL. Clustering and frequent pattern mining are two most usually used UL methods for games. K-means, k-medoids, and DBSCAN are common used clustering algorithms [31]. Besides, popular frequent pattern mining methods include the Apriori algorithm for itemset mining, and SPADE [32] and GSP [33] for sequence mining.

6 Conclusions

As the list in the previous section shows that games are an ideal domain for AI, which poses a large number of open problems. In this paper, we review the development of game AI. Different AI methods and their applications in games are presented. Obviously, it is very difficult to design a universal game AI with a unified framework. Different tasks represent different abstractions and involve different time spans. Therefore, a highly intelligent game AI needs the combination of various AI methods in order to learn from each other's strong points, and the emergence of notable hybrid algorithms such as neuroevolution and deep Q-learning is strong evidence.

References

1. Cruz RMO, Sabourin R, Cavalcanti GDC et al (2015) META-DES: a dynamic ensemble selection framework using meta-learning. Pattern Recognit 48(5):1925–1935
2. Schaeffer J, Lake R, Lu P et al (1996) Chinook: the world man machine checkers champion. AI Mag 17(1):21
3. Campbell M, Hoane AJ, Hsu F (2002) Deep blue. Artif Intell 134(1):57–83
4. Silver D, Huang A, Maddison CJ et al (2016) Mastering the game of Go with deep neural networks and tree search. Nature 529(7587):484–489
5. Mnih V, Kavukcuoglu K, Silver D et al (2015) Human-level control through deep reinforcement learning. Nature 518(7540):529–533
6. Tang Z, Shao K, Zhu Y et al (2018) A review of computational intelligence for StarCraft AI. In: IEEE SSCI, pp 1167–1173
7. Ontañón S, Synnaeve G, Uriarte A et al (2013) A survey of real-time strategy game AI research and competition in StarCraft. IEEE Trans Comput Intell AI 5(4):293–311
8. Ketter W, Collins J, Gini M et al (2012) Real-time tactical and strategic sales management for intelligent agents guided by economic regimes. Inf Syst Res 23(4):1263–1283
9. Hidalgo-Bermúdez RM, Rodríguez-Domingo MS, Mora AM et al (2013) Evolutionary FSM-based agents for playing super Mario game. LION. https://doi.org/10.1007/978-3-642-44973-4_39
10. Toll WGV, Iv AFC, Geraerts R et al (2012) Real-time density-based crowd simulation. Comput Animat Virtual Worlds 23(1):59–69
11. Jadon S, Singhal A, Dawn S (2014) Military simulator—a case study of behaviour tree and unity based architecture. Int J Comput Appl 88(5):26–29
12. Guillaume MJ, Mark HM, Herik HJ et al (2008) Progressive strategies for Monte-Carlo tree search. NMNC 4(03):343–357
13. Coquelin R, Munos A (2007) Bandit algorithms for tree search. In: Conference on uncertainty in artificial intelligence, pp 67–74
14. Silver D, Sutton RS, Müller M (2008) Sample-based learning and search with permanent and transient memories. In: Annual international conference on machine learning, pp 968–975

15. Marcolino LS, Matsubara H (2011) Multi-agent monte carlo Go. In: International conference on autonomous agents and multiagent systems, pp 21–28
16. Young J, Hawes N (2012) Evolutionary learning of goal priorities in a real-time strategy game. In: AIIDE-12
17. Sánchez PG, Tonda AP, Garcia AM et al (2015) Towards automatic StarCraft strategy generation using genetic programming. In: IEEE CIG, pp 284–291
18. Köstler H, Gmeiner B (2013) A multi-objective genetic algorithm for build order optimization in StarCraft II. KI 27:221–233
19. Carlos A, Gary B, David A (2007) Evolutionary algorithms for solving multi-objective problems. Springer
20. Justesen N, Risi S (2017) Continual online evolutionary planning for in game build order adaptation in StarCraft. In: GECCO, pp 187–194
21. Svendsen JB, Rathe EA (2012) Micromanagement in StarCraft using potential fields tuned with a multi-objective genetic algorithm. Master's thesis, NTNU
22. Gabriel I, Negru V, Zaharie D (2012) Neuroevolution based multi-agent system for micromanagement in real-time strategy games. In: BCI, pp 32–39
23. Santiago O, Synnaeve G, Uriarte A et al (2015) RTS AI problems and techniques. https://doi.org/10.1007/978-3-319-08234-9_17-1
24. Lu F, Yamamoto K, Nomura LH et al (2013) Fighting game artificial intelligence competition platform. In: IEEE GCCE. https://doi.org/10.1109/gcce.2013.6664844
25. Cho HC, Kim KJ, Cho SB (2013) Replay-based strategy prediction and build order adaptation for StarCraft AI bots. In: IEEE CIG, pp 1–7
26. Dereszynski E, Hostetler J, Fern A (2011) Learning probabilistic behavior models in real-time strategy games. In: AIIDE, pp 20–25
27. Justesen N, Risi S (2017) Learning macromanagement in StarCraft from replays using deep learning. In: IEEE CIG, pp 162–169
28. Zhao D, Shao K, Zhu Y et al (2016) Review of deep reinforcement learning and discussions on the development of computer Go. IET Control Theory A 33(6):701–717
29. Usunier N, Synnaeve G, Lin Z et al (2017) Episodic exploration for deep deterministic policies: an application to StarCraft micromanagement tasks. In: International conference on learning representations
30. Shao K, Zhu Y, Zhao D (2018) StarCraft micromanagement with reinforcement learning and curriculum transfer learning. In: IEEE TETCI. https://doi.org/10.1109/tetci.2018.2823329
31. Bloembergen D, Tuyls K, Hennes D et al (2015) Evolutionary dynamics of multi-agent learning: a survey. J Artif Intell Res 53(1):659–697
32. Al-Yaseen WL, Othman ZA, Nazri MZA (2015) Hybrid modified k-means with C4.5 for intrusion detection systems in multiagent systems. Sci World J. https://doi.org/10.1155/2015/294761
33. Zaki MJ (2001) SPADE: an efficient algorithm for mining frequent sequences. Mach Learn 42(1–2):31–60

Intelligent Exhaust System of Cross-Linked Workshop

Yan-ting Xu(✉) and Yongjie Yang

School of Information Science and Technology, Nan-tong University,
Nan-tong 226019, Jiang-su, China
1061463697@qq.com, yang.yj@ntu.edu.cn

Abstract. An intelligent exhaust system has been designed based on the characteristics of practical and efficient. The system takes STM32F103C8T6 single-chip microcomputer as the control core and consists of four parts: acquisition terminal nodes, routing nodes, master node, and monitoring management software. Use SHT20 sensors to collect environmental data. To realize the function of the system's data transceiver and the exhaust fan control, data exchange between STM32 and the wireless transmission module LORA has been adopted. The upper computer not only contains PC monitoring management software, but also contains WeChat small program, which makes the fan control more simple and convenient. The system has been applied in a cross-linked workshop. The practical application shows that the system is stable, accurate, and easy to operate.

Keywords: Ad hoc network · Data acquisition · LORA

1 Introduction

Nowadays, wireless sensor network has been widely used in factories, military, agriculture, and other fields due to the rapid development of wireless communication technology, embedded technology, and Internet of Things technology. In the environment of modern factory, agriculture, military and home, environmental parameters including temperature and humidity seriously affect the quality of production and life. Therefore, it becomes an important research direction to acquire these environmental parameters in time and turn on or off the exhaust fan in real time according to these parameters. In the existing intelligent exhaust systems, there are more or less disadvantages: (a) limited monitoring scope due to short transmission distance; (b) low practicability. Once a node is damaged, the whole system will collapse; (c) inaccurate environmental parameters and incomplete monitoring. Based on the above description, this paper proposes a self-organized network [1] intelligent exhaust system centering on LORA communication.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 304–312, 2020.
https://doi.org/10.1007/978-981-15-0187-6_35

2 Overall Scheme Design of the System

This paper designs an intelligent exhaust fan control system based on STM32, which is mainly composed of four parts: acquisition terminal nodes, routing nodes, master node, and monitoring management software. The system model is shown in Fig. 1. The acquisition terminal nodes are responsible for collecting temperature and humidity data, processing this data, and transmitting it to the nearby route. The acquisition terminal nodes can also control the switch of the exhaust fan according to the command issued by the monitoring management software. The routing nodes are responsible for monitoring the data of its neighboring acquisition terminal nodes, receiving and sending it, and acting as a transit station. The master node is responsible for receiving and forwarding the routing nodes or collecting the environmental data of the acquisition terminal nodes. Monitoring management software consists of two parts: PC side management software and mobile side management software. Users cannot only monitor environmental parameters on the PC side and control the switch of the exhaust fan, but also realize these functions from the mobile side at any time. It makes the monitoring and management of the exhaust fan more convenient.

The system has the characteristics of self-organization and non-centrality. The network can be reconfigured automatically without relying on the network of other devices. When the number of nodes in the network changes, the routing nodes will be automatically adjusted to realize the self-organization of the whole network. The joining or leaving of single or multiple nodes does not affect the operation of the whole network and increases the practicability of the network.

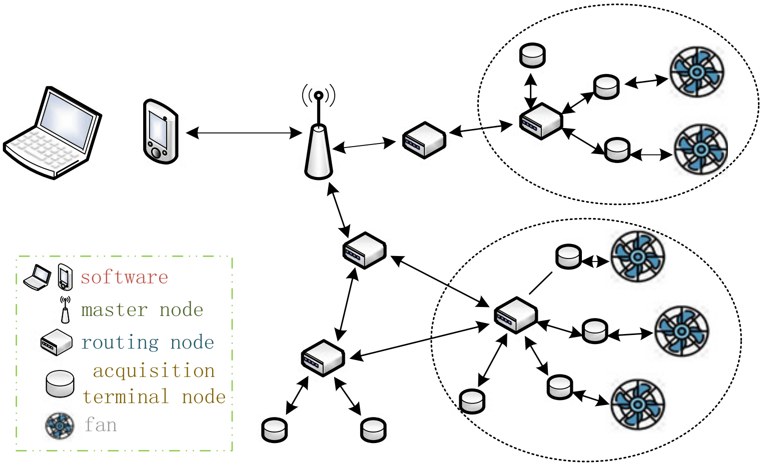


Fig. 1. System model

3 System Hardware Design

The hardware design of this system is mainly divided into three modules: exhaust fan control module, data acquisition module, and transmission module.

3.1 Exhaust Fan Control Module

The relay used in the exhaust fan control module is a solid-state relay [2] JGX-5. This type of relay used in the automatic control circuit is actually an “automatic switch” that uses a small current to control a large current. Figure 2 is the circuit diagram of the relay, where D3 is the light-emitting diode which indicates the open of the relay. U1 is an optical coupler; Q1 is an NPN transistor which plays the role of a switch in this circuit; D2 is a voltage stabilizing diode which is connected in parallel at a higher voltage to stabilize the voltage.

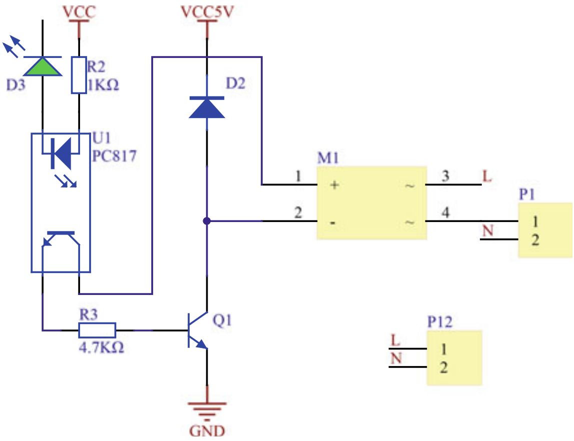


Fig. 2. Exhaust fan control module circuit

3.2 Data Acquisition Module

The sensor in the data acquisition module is the SHT20 [3], a new generation Sensirion temperature and humidity sensor equipped with a 4C CMOSENS® chip. The chip contains a capacitive relative temperature and humidity sensor, a gap temperature sensor, an amplifier, an A/D converter, OTP memory, and digital processing unit. SHT20 can also ensure stable performance in the case of high humidity. Therefore, it has high cost performance, and its maximum error is 3%, which fully meets the design requirements of the system. Figure 3 shows the temperature and humidity acquisition circuit. The power supply of SHT20 is 3.3 V, and a 100 nF decoupling capacitor is connected between the power supply (VDD) and grounding (GND) for filtering; The function of SCL pin is to synchronize the communication between MCU and SHT20. The role of SDA pin is the input and output of the data. When the sensor receives electrical signals, SDA is effective when SCL is in the rising edge state. When SCL is in the high power state, SDA must be guaranteed to be in a stable state. Only after SCL becomes the falling edge can the value of SDA be changed. Otherwise, it cannot be changed. In order to avoid signal conflicts, MCU can only drive SDA and SCL at low level. So, an external pull-up resistance is needed to upgrade the signal to high level, namely R5 and R6 in the figure.

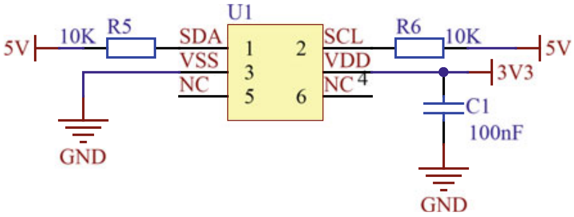


Fig. 3. Data acquisition module circuit

3.3 Transmission Module

STM32 microprocessor whose working frequency is 72 MHz is not only the core of the transmission module but also the core of the circuits in this system. The wireless communication module used in the transmission module is LORA [4]. The connection between this module and MCU is shown in Fig. 3. RXD and TXD are, respectively, connected to TXD and RXD of MCU. AUX is used to indicate the working state of the module. The user wakes up the external MCU and outputs the level during self-test initialization. MD0 and MD1 are connected to the IO port of the MCU, which determines the working mode of the module and cannot be suspended (Fig. 4).

4 System Software Design

The software design part of this system is divided into two parts: lower computer software and monitoring management software. Among them, monitoring management software is divided into PC side management software and mobile side management software.

4.1 Lower Computer Software Design

The lower computer software is written by Keil [5] software, and the whole software includes main function, sensor driver program, LORA wireless communication protocol, and exhaust fan control subroutine. Among them, the wireless communication protocol is the core of the lower computer software.

The wireless communication protocol is mainly divided into two parts: the transmission module obtains the data of the single-chip microcomputer and the single-chip microcomputer responses the data of transmission module. As shown in Table 1, the transmission module obtains the data of the single-chip microcomputer. As shown in Table 2, the single-chip microcomputer responses the data of transmission module.

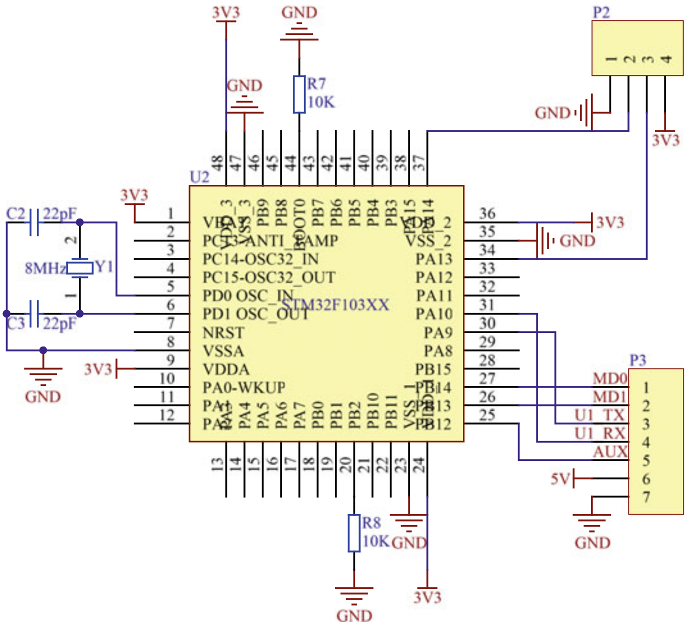


Fig. 4. Transmission module circuit

Among them, add represents the communication address of the receiver; cha means a communication channel; adds means the mailing address of the sending side; cmd is the check bit, which is used to tell the receiver the purpose of sending data; DATA is a sequence of messages that hold the messages to be transmitted during the transmission; end is the end bit, indicating the end of the transmission of the system.

(1) The purpose of including the sender address in the protocol is to facilitate the receiver to determine the source of the message;
(2) All instructions are sent in hexadecimal format;
(3) CMD is 0x01 when obtaining the data instruction of single-chip microcomputer. CMD is 0x02 when single-chip microcomputer responds;
(4) When CMD is 0x01, the DATA store the query instruction, and when CMD is 0x02, the DATA store the data of temperature and humidity;
(5) When the wireless module fails to receive the single-chip microcomputer data packet in 5 s, it sends the data again. If the above actions are repeated for three times and no response is received, the address of this receiver will be reported and the request data will not be sent to this address in the next cycle.

Table 1. Transmission protocol of obtaining the data of the single-chip microcomputer

add (2B)	cha (1B)	adds (2B)	cmd (1B)	DATA (nB)	end (1B)
Receiver address	Channel	Sender address	Check bit	Query command	End bit

Table 2. Transmission protocol of responding the data of the transmission module

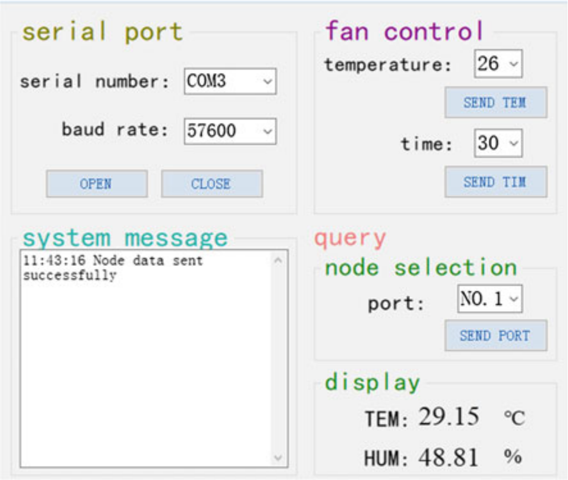
add (2B)	cha (1B)	adds (2B)	cmd (1B)	DATA (nB)		end (1B)
				Tem	Hum	
Receiver address	Channel	Sender address	Check bit	Temperature	Humidity	End bit

4.2 Monitoring Management Software Design

4.2.1 PC Side Management Software

PC side management software is written by Visual Studio software. Figure 5 is its main control interface. The interface includes four parts: serial port setting, fan control, system message, and temperature and humidity inquiry. The PC side management software mainly monitors the temperature and humidity of the nodes and manages the exhaust fans.

First of all, after the open of the PC side management software, it is necessary to set the corresponding port number and baud rate, and then open the serial port. If sending is successful after selecting the corresponding node and clicking send node button, the system message column will show the words "Node data sent successfully," and real-time display box will show the node's temperature and humidity, respectively. If sending is failure, the system message column will show the word "node data sent failure." It is necessary to send again or check whether the device is damaged at this time. Set the required temperature or time threshold in the fan control bar, and click the send button to send to control the switch of the exhaust fan.



serial number: COM3

baud rate: 57600

temperature: 26

time: 30

11:43:16 Node data sent successfully

port: NO. 1

TEM: 29.15 ℃

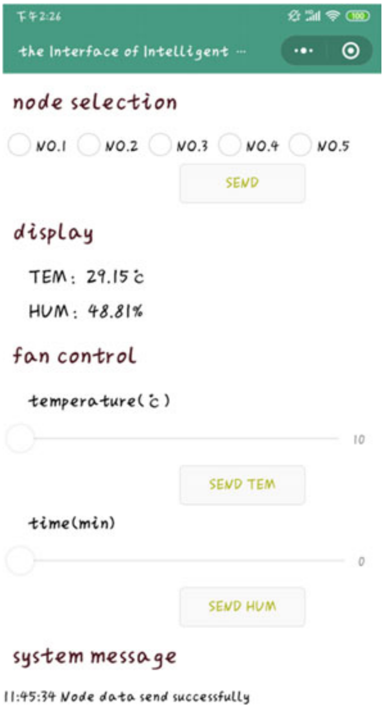
HUM: 48.81 %

Fig. 5. PC side management software interface

4.2.2 Mobile Side Management Software

This design uses WeChat [6] small program to write mobile side management software. WeChat small program is a new way of development. Developers can quickly develop a program, and this program can be easily obtained in WeChat. Do not need to install, scan code can be.

Figure 6 is the mobile side management software interface, which is mainly divided into four parts: node selection, temperature and humidity display, fan control and system message. Its function is consistent with PC side management software.



TEM: 29.15℃

HUM: 48.81%

11:45:34 Node data send successfully

Fig. 6. Mobile side management software interface

5 System Test

The principle, hardware design, and software design of the whole system have been introduced above. The following will prove the feasibility of the system according to the experimental data of the system.

Take node 1 as an example, it is necessary to start the system, energize the master node, routing nodes and acquisition terminal nodes 1, and open the PC side management software or mobile terminal management software. Then, set the temperature threshold of node to 28, and click the send button to send. In order to facilitate the explanation, different states of the exhaust fan are selected for explanation. The parameters and states are shown in Table 3.

Table 3. System data

Temperature (°C)	Humidity (%)	State
26.51	43.17	Close
27.43	41.38	Close
28.04	48.56	Open
28.61	50.49	Open

As can be seen from the table, when the temperature value of each node is greater than the threshold, the exhaust fan is opening; otherwise, the exhaust fan is closing. Test results show that the actual results are consistent with the preset.

The stability test is mainly aimed at the test of communication transmission reliability between the acquisition terminal nodes and the routing nodes. This design has five temperature and humidity sensors for data collection and each collection cycle is 1 s. The total data collection time is half an hour, and the results are shown in Table 4.

Table 4. Stability test

Acquisition node	Number of sending	Number of receiving	Success rate (%)
Node 1	1800	1782	99.0
Node 2	1800	1789	99.4
Node 3	1800	1779	98.8
Node 4	1800	1784	99.1
Node 5	1800	1790	99.4

As can be seen from Table 2, a total of 1800 times were collected within half an hour, and the success rate was about 99.14%. In the process of data transmission, packet loss occurred occasionally, and the system was very stable.

6 Conclusion

In this paper, an intelligent exhaust system based on STM32 is implemented. The system takes data collection as the underlying node, routing as the relay station for information collection, and PC side management software and mobile side management software as the control interface for data processing and storage. LORA communication is used to monitor and control the temperature and humidity of each node remotely. The communication protocol of this system greatly reduces the loss rate of data and ensures the stability of the system, which is of great significance for the factory to realize intelligent.

References

1. Yin ZP (2014) Method analysis and design on self-organizing network. In: International conference of information science and management engineering (ISME), vol. 46(4), pp 1221–1227
2. 周卫兵 (2010) 固态继电器的特点及应用. 山西电子技术 (01):90–92
3. Du JY, Guo JB, Xu, DD, Huang Q (2017) A remote monitoring system of temperature and humidity based on onenet cloud service platform. In: IEEE electrical design of advanced packaging and systems symposium (EDAPS)
4. Ke KH, Liang QW, Zeng GJ, Lin JH, Lee HC (2017) Demo abstract: a LORA wireless mesh networking module for campus-scale monitoring. In: 16th ACM/IEEE international conference on information processing in sensor networks (IPSN), pp 259–260
5. Zheng H (2016) Application of Proteus and Keil software in the teaching of MCU course. In: International conference on artificial intelligence and computer science (AICS), pp 411–414
6. Liu J, Wang Y, Xu H (2015) Research of Wechat network information transmission based on the complex network. In: International conference on intelligent systems research and mechatronics engineering (ISRME) (121), pp 1923–1926

Combined with DCT, SPIHT and ResNet to Identify Ancient Building Cracks from Aerial Remote Sensing Images

Tiantian Tang, Bingcai Chen(✉), and Ruiheng Hu

Xinjiang Normal University, Urumqi, China
528879201@qq.com, 33682049@qq.com

Abstract. In order to study the crack detection of ancient buildings in aerial remote sensing images, this paper proposes to preprocess remote sensing images by combining block DCT transform with SPIHT algorithm terminating at the threshold to retain the target information to the maximum extent and reduce the number of processed images. Then, ResNet model is used to detect the cracks in ancient buildings.

Keywords: SPIHT · DCT · ResNet · Crack detection · Aerial remote sensing images

1 Introduction

The traditional manual inspection of ancient buildings has obviously failed to meet the needs of comprehensive inspection and restoration of ancient buildings. The characteristic of remote sensing technology is that it does not need to detect the target on the spot, but can directly collect and identify relevant information. UAV remote sensing image has the characteristics of wide coverage, large change of gray value, complex and changeable texture information, and precious acquired information. Therefore, how to deal with massive remote sensing images, reduce training examples and maximize the retention of feature information, so as to facilitate classification is the research focus.

Discrete cosine transform (DCT), first proposed by Ahmed and Rao, is considered as the best method to transform language signal and image signal because its performance is close to that of k-l transform. DCT transform can concentrate image energy to a few low-frequency DCT coefficients, which has low complexity and strong generalization ability [1]. DCT uses Huffman coding, and the frequency determines the length of coding. The transformed image is not dynamically encoded and the image compression ratio cannot be customized according to the actual research object. Set Partitioning in Hierarchical Trees (SPIHT) algorithm determines the initial scanning threshold based on the image wavelet coefficient [2]. The initial threshold value is changed dynamically according to the last scan result, and the threshold value is halved on the basis of the last scan. After setting the compression ratio, if the scanning threshold is reduced to the expected value, the cyclic scanning will stop. Thus, the

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 313–318, 2020.
https://doi.org/10.1007/978-981-15-0187-6_36

problem of excessive memory storage occupied by multiple cyclic scans will not occur, and the encoding time and storage space occupation rate will be reduced.

AlexNet selects Re as CNN's excitation function to solve the problem of increasing gradient dispersion in network depth [3]. In the last several complete connection layers, some neurons were randomly neglected to avoid overfitting of the model. Max pool eliminates the blur effect of average pool. The step size is smaller than the size of the pool core, but the high-level features are not easy to be classified. Based on the fine-tuning of AlexNet, ZFNet changes the first layer of AlexNet: The size of the filter is changed from 11 × 11 to 7 × 7, and the step size is changed from 4 to 2, retaining more features. Through deconvolution, the feature map reflects the architecture of feature hierarchy. The first layer learns specific physical features, such as texture and edge, while the second layer learns and identifies abstract features related to types. Thus, the higher the features, the better the classification performance; however, the higher up the hierarchy, the longer the training is needed before convergence. VGGNet further studied the relationship between the depth and performance of the convolutional neural network [4], using a 3 × 3 small convolution kernel and a 2 × 2 maximum pooling layer, and repeated stacking. As the network deepens, the pool shrinks by half and the number of channels doubles. While simplifying the structure of convolutional neural network, a large number of training features are retained.

Generally, the idea of establishing an ideal optimal model is to increase the depth of the model or the number of neurons. But there are some drawbacks:

- If the training data set is limited and the parameters are too many, overfitting is likely to occur.
- The larger the network model structure is, the more complex the calculation will be, making it difficult to carry out practical application.
- The deeper the network model structure is, the more easily the gradient is lost in the deep layer, and the optimization model is difficult to be optimized.
- Degradation: Accuracy first increases with depth, and when saturation is reached, the increase of model depth leads to the decrease of accuracy.

In order to solve these problems, we believe that non-uniform sparse data computer software and hardware are inefficient. ResNet assumes that the learning target is converted into the difference value F(x) = H(x)−x between the expected output H(x) and the direct output x, and the initial result is defined as the input x [5]. The process is shown in Fig. 1.

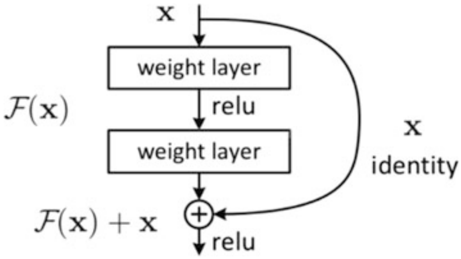

x
weight layer
relu
weight layer
$\mathcal{F}(\mathbf{x})$
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Fig. 1. Residual unit

The residual element decomposes a problem into multi-scale linear residual, which is very useful for optimization training. The input and output of the block are overlapped by the elements through short interception. Simple addition will not increase the additional parameters and calculation of the network, but can greatly improve the training speed of the model, improve the training effect; degradation is handled well as the network deepens.

The innovation of this paper lies in the combination of image compression algorithm and convolutional neural network. The improvement of block DCT is combined with SPIHT algorithm. The image information is retained to the greatest extent and the number of training samples is reduced to concentrate the image energy. Change the block size and step size of DCT image to ensure the image compression speed and feature retention integrity; the ResNet model is used to classify the preprocessed images accurately, which improves the accuracy of ancient building crack identification.

2 Block DCT

The image is kaleidoscopic in the spatial domain, and it is difficult to be processed in a unified range. Therefore, we consider transforming the image in the frequency domain and making the main frequency coefficients of the image relatively concentrated. Because the information in the low-frequency part of the image is much more than the information in the high-frequency part, DCT is a good way to do that. If the whole image is transformed by DCT, the complexity is very high. Therefore, DCT transformation and inverse transformation are implemented in parallel for each block to improve the efficiency of transformation.

In general, most of the images are divided into 8 × 8 blocks. The larger the block size is, the fewer blocks the image is divided into, the more obvious the boundary of the image block is, and the number of edges reflected by the block increases sharply. The more serious the quantization error is, the better the image will be visually. However, too small block size will increase the computational complexity and reduce the computational speed. In this paper, we divide each image into 10 × 10 blocks, which increases the speed compared with 8 × 8. The processing step size of each sub-block is segmented from 1 to 0.8. In the image of a single sub-block, the step size of each processing becomes smaller, which can retain image feature information to a greater extent. The formula of the basis function in the two-dimensional image is as follows:

$$\mathrm{b}[x,y] = \cos\left[\frac{(2x+1)u\pi}{16}\right]\cos\left[\frac{(2y+1)v\pi}{16}\right] \tag{1}$$

x and y represent the coordinates of the pixels in the spatial domain, and u and v represent the coordinates in the frequency domain of the basis function. Because the basis function is based on 10 by 10 blocks, x, y, v, u are all between 0 and 9.

3 Block DCT Combined with SPIHT

For the sake of reconstructing the fidelity of the image, the quantization method in compression does not use the quantization matrix, but uses the planar quantization method to retain the image details. When SPHIT algorithm is coding, the determined value of image wavelet coefficient is taken as the initial scanning threshold. In sequential scanning, the threshold value of each scan is halved on the basis of the previous one. As the threshold value decreases, the coefficient in the image increases continuously. However, in practical application, if the cyclic scanning reaches the threshold value of 1, let the scanning stop. A large amount of scan data will be generated, but these data are highly redundant, which is not very helpful for image compression and reconstruction. Therefore, we can flexibly set the termination threshold according to the actual requirements of image compression, and stop scanning and encoding when the expected value is reached. Fig. 2 is a flowchart combining block DCT with termination threshold SPIHT algorithm.

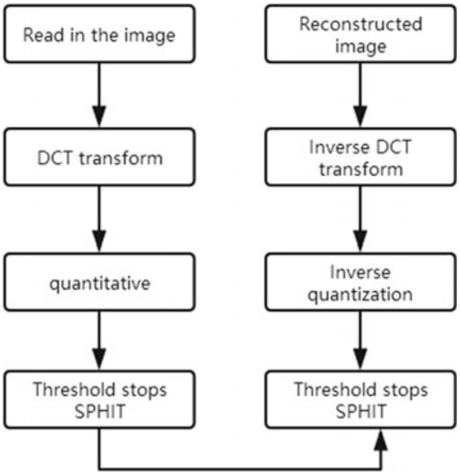


Fig. 2. Fusion of block DCT and SPIHT

4 ResNet

The challenge of traditional convolutional neural network is how to solve the problems of information loss and energy loss on the premise of obtaining the optimal model. With gradient disappearance and gradient explosion, the deep network cannot be trained, and the complexity increases but the accuracy decreases. ResNet directly detours the input information to the output, simplifying the model while protecting the integrity of the information. The entire network only needs to learn the difference between its input and output. And ResNet has many bypasses that connect the input directly to the back layer, the modules it really USES are not single, but take two ways, as shown in Fig. 3.

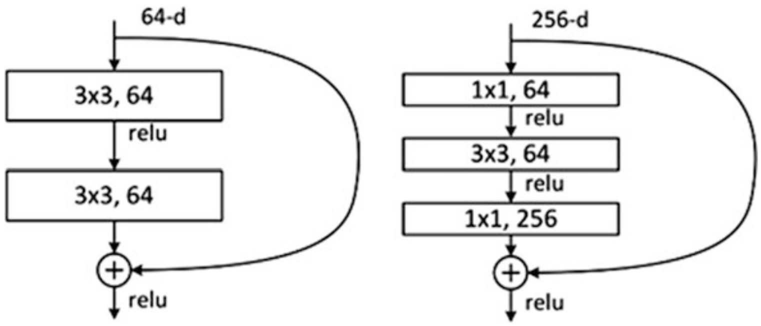


Fig. 3. Two different residual models

We can clearly see that the 256-dimensional channel has been reduced to 64 dimensions. Then, 1 × 1 convolution is adopted for recovery, which can directly reduce the number of dimensions and parameters through these two methods. When the number of channels F(x) and x are different, the calculation methods for different channels can be divided into the following two types (Fig. 4).

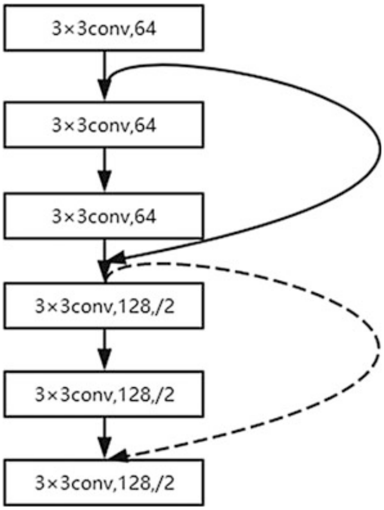


Fig. 4. Different stacking modes

The connection part of the solid line adopts the convolution of 3 × 3 × 64, and the number of channels is the same. The calculation method is as follows:

$$y = \mathrm{F}x + x \tag{2}$$

The dashed connection part adopts convolution of 3 × 3 × 64 and 3 × 3 × 128, respectively. The number of channels is different (64 and 128). The calculation method is as follows:

$$y = \mathrm{F}x + \mathrm{W}x \tag{3}$$

5 Conclusion

This paper combines ResNet convolutional neural network, block DCT and SPHIT. A large number of aerial images were compressed in advance, and feature information was retained to the greatest extent, so as to reduce training objects for volume and neural network. Block size and step size were changed to ensure compression rate and information retention. ResNet convolutional neural network is used to classify compressed images and identify ancient buildings with cracks.

References

1. Zhang J, Xu G (2015) Application of remote sensing image compression algorithm based on DCT. Avionics Technology
2. Qu H, Nie H (2018) Image compression algorithm based on SPIHT. Information and Computers Theory. 2018(07)
3. Progress in the study of convolutional neural networks (2018) J Hebei Univ (Natural Science Edition)
4. Huang W, Tang Y (2015) Actual combat of Tensor Flow. Beijing Electronic Industry Press. 05
5. He K (2015) Deep residual learning for image recognition

Design and Construction of Intelligent Voice Control System

Dapeng Zhu(✉), Shu Liu, and Chao Liu

Letter Box 21, Urumqi 83001, China
lingo_renlo@qq.com

Abstract. With the development of speech recognition technology, there are wide applications of intelligent voice around the world. In the paper, we propose an intelligent voice control system which can realize the functions of voice activity detection, speech recognition, speaker authentication, instruction analysis, and automatic answer. A new voice activity detection (VAD) algorithm and backtracking algorithm are proposed to improve system performance. Then, we test our system in our own silumation corpus, we create the system evaluation indexes and the text result is satisfactory.

Keywords: Speech recognition · VAD · Backtracking algorithm

1 Introduction

Speaker recognition (recognizing who was speaking) and speech recognition (recognizing what people said) are used widely in the field of artificial intelligence. For example, paper [1] designs a specific human voice recognition home control system by improving the corresponding algorithm. Paper [2] designs a Chinese text and audio editing system based on AI, improving the editing efficiency of audio content and the timeliness of audio news release.

In this paper, we propose an intelligent voice control system. The system uses speech recognition and speaker recognition technologies based on deep neural network. It can recognize specific speakers and process his or her continuous voice command in real time, realizing deep neural network interference and instruction analysis. We do not use the online services of speech-to-text transcription and voiceprint authentication supported by Internet providers. We build specialized instruction dataset and relevant Chinese corpus. In consideration of multi-speakers, high background noise, random speaking voice pauses, and no wake-up-word for speech recognition, we propose voice activity detection algorithm and backtracking algorithm to recognize right instruction. In addition, we apply Chinese initial and tunal final speech model label method. Finally, we define the system evaluation indexes to show us the system performance from multi-aspects.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 319–325, 2020.
https://doi.org/10.1007/978-981-15-0187-6_37

2 Project Description

Different from traditional manual input commands, the intelligent voice control system uses voice information to execute commands. Our system has following functions:

a. Acquire voice message.
b. Recognizing the command form speech data.
c. Speaker recognition and authentication.
d. Broadcast current status and parameters with synthesized speech when asked.

The platform is divided into three layers: physical layer, data and information layer, and business logic layer (see Fig. 1).

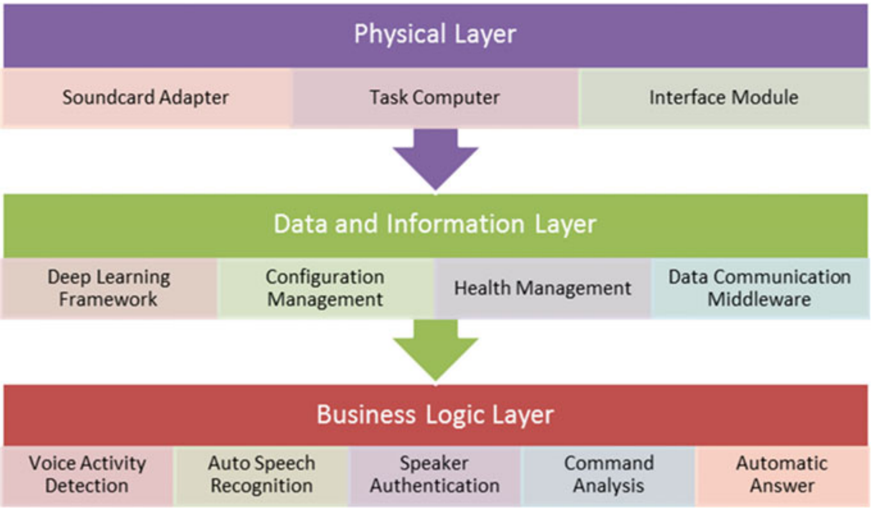


Fig. 1. System structure

Physical layer consists of sound card adapter, task computer, and interface module. Sound card adapter includes a chip microprocessor and a sound card, and it can collect speech signal and play synthesis speech. As for task computer, this is the main part of speech recognition, speaker authentication, instruction analysis, and speech synthesis; interface module realize interaction function to sensors.

Data and information layer includes deep learning framework, configuration management, health management, and data communication middleware.

Business logic layer consists of voice activity detection, auto speech recognition, speaker authentication, instruction analysis, and automatic answer.

Figure 2 is flow chart of system.

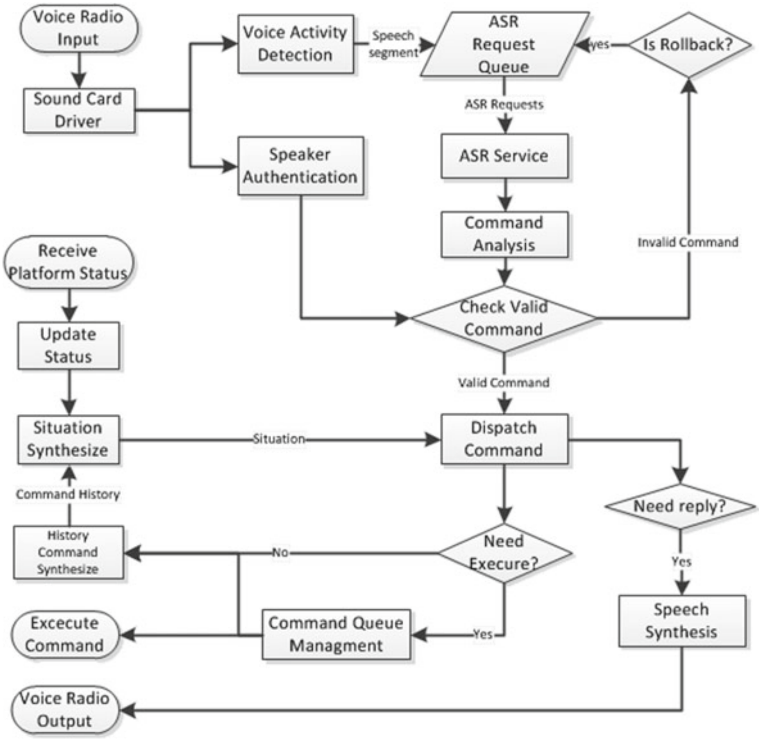


Fig. 2. Flowchart of system

3 System Algorithms

Our system involves speech recognition, speaker authentication, instruction analysis, mission planning, and automatic answer technologies. In the paper, we describe algorithms about speech recognition and instruction analysis in detail, and they are voice activity detection algorithm and backtracking algorithm.

3.1 Voice Activity Detection Algorithm

The voice activity detection algorithm is based on Google's webrtcvad technique [3] which is used to classify a piece of audio data as being voiced or unvoiced. By using smooth constant length windows detector, the continuous input speech stream is segmented into variable speech and non-speech portions, and we abstract all speech segments and sort them:

$$E = \{(s_i, e_i), i \in (0, k)\}$$

whereas, k is speech segments number, s_i and e_i represent starting time and ending time of Nth speech segment.

First, input audio data is segmented to frames at intervals of 10 ms: $x = (x_1, x_2, \ldots, x_t)$;

Second, webrtcvad algorithm and activity results: $v = (v_1, v_2, \ldots, v_t)$;

Third, smooth windows with the constant length ($n = 30$) and get average results: $v_t^* = \frac{\sum_{i=t-n}^{t} v_i}{n}$;

Fourth, abstract speech segments based on v^* value and product sequences.

3.2 End-to-End Speech Recognition Model

There are two widely used speech recognition models: "acoustic model and language model" and end-to-end recognition model, and our paper use the DeepSpeech2 model.

DeepSpeech2 model building: DeepSpeech2 model is based on PyTorch architecture, we segment the single-channel audio data at 16,000 Hz frequency into frames with 10 ms interval and compute mel-frequency cepstral coefficient (MFCC) features. Convolution kernel size at CNN layer 1 is $41 * 11$, at CNN layer 2 is $21 * 11$, as for the bidirectional recurrent layers, we apply four layers GRU with 512 width. Output layer is a fully connected layer.

As for model label, we design four plans to choose. They are:

1. Plan A: English alphabet + number + non-specific Chinese character, each label represents an English alphabet or a number or a Chinese character, the total labels number is 3583.
2. Plan B: English alphabet + number + specific Chinese character, each label represents an English alphabet or a number or a specific Chinese character included in the command set, the total labels number is 197.
3. Plan C: English alphabet + number + Chinese phonetic alphabet, each label represents an English alphabet or a number or a Chinese phonetic alphabet, the total labels number is 1449.
4. Plan D: English alphabet + number + Chinese syllable I/F (initials or finals), each label represents an English alphabet or a number or an i/F, the total labels number is 220.

Plan C and Plan D use Viterbi algorithm to map pinyin or syllable I/F to character. In the paper, we apply five-order language model to evaluate state transition probability.

We choose plan D and apply initial and tonal final combination two-variable label method which improves its correct recognition rate from 91 to 97%. The main phases are as follows:

A. Training, encode Chinese characters to initial and tonal final combination labels, single final character is decoded to one final label. For example, "安" is encoded to "ā n" tonal final label, "语" is encoded to both "y" initial label and "ǔ" tonal final label.
B. Interface, decode label sequences to Chinese phonetic alphabet sequences, use trie tree segment Chinese phonetic alphabet sequences to pinyin sequences, then use kenlm language model and sparse viterbi decoder [4] to map pinyin sequences to Chinese characters sequences.

In order to improve decoding speed, we use language model to compute state transition probability; meanwhile, we build the mapping table between pinyin and Chinese characters.

3.3 Instruction Analysis Algorithm

We define the instructions with specifical format criterion, and our paper uses part of speech label and pattern matching methods in instructions analysis; main steps are as follows:

1. Building part of speech pattern set including all instructions.
2. Part of speech label every input characters sequence.
3. Extract verbs, search for a corresponding pattern.
4. Match a pattern with the sequences and output matched instructions.

The paper uses Jieba module (Chinese Words Segmentation Utilities) [5], and revises user dictionary to make optimization of Chinese Words Segmentation and part of speech label.

3.4 Backtracking Algorithm

Only the complete instruction recognized by the system is effective. In order to avoid the system from segmenting a complete instruction speech to fragments, we usually set more aggressive mode of VAD, increase the width of smooth windows and reduce the VAD threshold. Nevertheless it will increase the system delay. Therefore, we propose backtracking algorithm to solve this problem.

We set less aggressive mode of VAD, decrease the width of smooth windows, and increase the VAD threshold. As for speech fragments, backtracking algorithm is used to combine these speech fragments to a speech segment which can be processed to a complete instruction. The backtracking algorithm is helpful for speakers who use long pauses.

Description: when the speech-to-text transcription does not match the instruction pattern, this speech segment is stored in cache. By recognizing different combinations of these speech segments, we can get the right instruction. Because the combinations' number increase with the square of the segment number as $\frac{k(k-1)}{2}$, we design recursion algorithm.

Assume at time T_k, the cache has k speech segments $E = \{(s_i, e_i), i = 1, \ldots, k\}$, then we combine segment as following formula:

$$E^* = \{(s_{k-1}, e_k), (s_{k-2}, e_k), \ldots, (s_1, e_k)\}$$

When E^* include effective instruction and clear the cache; Set chache size, when k is bigger than its size or than the value of $e_k - s_1 > \delta_T$, delete the $e_i < e_k - \delta_T$ segments ordered in caches. We can conclude that until k time, the computation number is $\frac{k(k-1)}{2}$.

4 Comprehensive Evaluation

We define the following test indexes:

1. Voice activity detection correct ratio (R_{vad}), the ratio of the number of correct speech samples detected by the system to all samples number:

$$R_{\text{vad}} = \frac{\sum_{i=1}^{N} C_{\text{vad}}}{N}$$

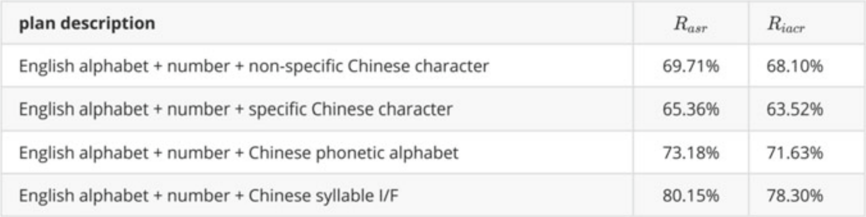
2. Speech recognition correct ratio (R_{asr}), word correct rate:

$$R_{\text{asr}} = \frac{\sum_{i=1}^{N} C_{\text{asr}}}{N}$$

3. Instruction analysis correct ratio (R_{iacr}), the ratio of the number of the correct instruction analyzed by the system to all samples number:

$$R_{\text{iacr}} = \frac{\sum_{i=1}^{N} C_{\text{iacr}}}{N}$$

In order to test the system comprehensive performance, we build the simulation test corpus. We firstly build speech samples set that includes samples collected from 150 people's speech samples, THCH30 (A Free Chinese Speech Corpus Released by Tsinghua University), and Baidu Voice Synthesis Set, and then we randomly extract samples from the set, add random intervals, and combine they with additive noise. Figure 3 is the table of speech recognition correct (Rasr) and instruction analysis correct ratio (Riacr) results in the test. Figure 4 shows the character error ratio (CER) and word error ratio (WER) during training. By comparing these results, it is clear that the 4th plan has the best comprehensive performance.



plan description	R_{asr}	R_{iacr}
English alphabet + number + non-specific Chinese character	69.71%	68.10%
English alphabet + number + specific Chinese character	65.36%	63.52%
English alphabet + number + Chinese phonetic alphabet	73.18%	71.63%
English alphabet + number + Chinese syllable I/F	80.15%	78.30%

Fig. 3. Result table

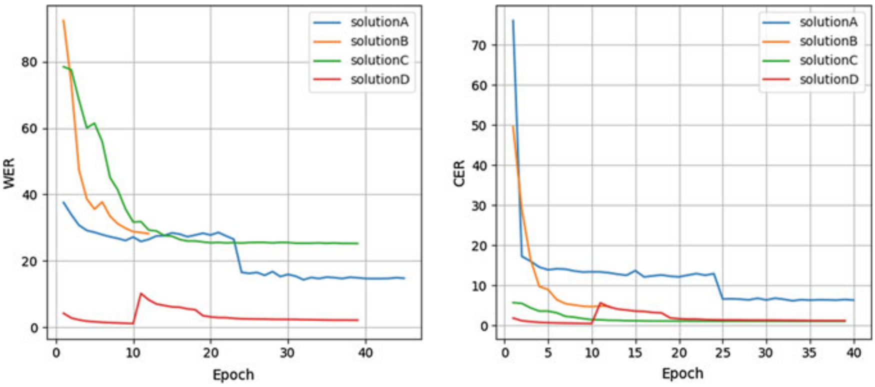


Fig. 4. Character correct ratio and word correct ratio

5 Conclusion

In the paper, we propose an intelligent voice control system which can realize the functions of voice activity detection, speech recognition, speaker authentication, instruction analysis, and automatic answer, performing us the good extension of artificial intelligent. In order to decrease system delay and increase correct recognition rate, we propose new VAD algorithm and backtracking algorithm; We contrast four model label plan and choose the fourth plan which uses initial and tonal final combination two-variable label method because of its good performance. Finally, we test our system in our own simulation corpus and the results are satisfactory. In the next work, we will test our system in an actual environment and update the system to improve its performance.

References

1. Peng Y, Zhang J, Chen R, Ren S (2017) A specific speech control home system. Comput Sci Appl 10:935–943
2. Niu S, Tang W (2018) Design of Chinese text and audio editing system based on AI. Content production and broadcasting, vol 45, pp 56–61
3. WebRTC. http://webrtc.org/
4. Jurafsky D, Martin JH (2018) An introduction to natural language processing, computational linguistic, and speech recognition
5. Chinese text segmentation. https://github.com/fxsjy/jieba

Estimation of Sea Clutter Distribution Parameters Using Deep Neural Network

Guoqing Wang[1], Hao Ding[1](✉), Chaopu Wang[2], and Ningbo Liu[3]

[1] Naval Aviation University, Shandong Yantai, People's Republic of China
gqwang80@126.com, hao3431@tom.com
[2] Beijing Institute of Technology, Beijing, People's Republic of China
[3] Naval Aviation University and No.23 Institute of the Second Academy, CASIC, Shandong Yantai and Beijing, People's Republic of China
lnb198300@163.com

Abstract. As a specific application of analytical methods on marine radar big data, this paper introduces deep learning theory into the field of sea clutter parameters estimation. A reasonable deep neural network model is built to estimate the parameters of amplitude distribution models so as to overcome the drawback of traditional methods based on statistical theory. In the proposed method, histogram method is used to preprocess the data, then deep neural network is trained with constructed dataset, and finally, parameter estimation results are obtained using test dataset. Validation results with simulation data and X-band radar-measured sea clutter data show that, compared with traditional estimation method, the deep neural network-based estimation method can improve parameter estimation accuracy significantly.

Keywords: Sea clutter · Parameter estimation · Deep neural network

1 Introduction

Sea clutter is one of the main factors that influence marine radar detection performance, and in order to minimize its effect, accurate statistical models are required in the design and development of advanced target detection algorithms [1–8]. Amplitude distribution model is an important type of statistical models, and current models mainly include Lognormal, Weibull, K, or Generalized K (GK). Traditionally, these model parameters are estimated via the statistical theory, such as maximum likelihood estimation method, moment-based estimation method, or numerical optimization method. Their main feature is that the requirement for sample support is big for a single estimation, which is not suitable for online real-time estimation. What's more, estimation results often fall into local optimal solutions under the condition of multiple parameters, resulting in a decline in modeling accuracy.

In order to overcome the drawback of traditional estimation methods, a reasonable deep neural network model is built to estimate the parameters of amplitude distribution models. In fact, in the past two years, intelligent processing technology, which is represented by deep learning theory, has been widely applied in radar. It is a more efficient intelligent signal processing method with the advantages of automatic

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 326–333, 2020.
https://doi.org/10.1007/978-981-15-0187-6_38

extraction of deep features with higher accuracy [9–12]. However, its application mainly concentrates on radar target detection or recognition, and in the field of parameter estimation, it is still a blank. In fact, for a specific theoretical distribution model, the parameter estimation problem can be abstractly treated as a complex nonlinear optimization problem, and deep learning theory has obvious technical advantages in solving this kind of problem. Based on this consideration, this paper constructs a deep neural network model with high precision parameter estimation ability. By selecting a reasonable structure of the deep neural network and training the deep learning model, the model can be equipped with multi-parameter estimation ability in the form of autonomous learning. Effectiveness of the estimation method is validated with simulation data and X-band radar-measured sea clutter data.

2 Sea Clutter Parameter Estimation Based on Deep Learning Theory

2.1 Deep Neural Network Model

Neural network is a mathematical model that simulates the behavior characteristics of animal neural network and performs distributed parallel information processing [12]. At present, the multilayer feed-forward neural network is the most widely used. The learning process of the network is divided into two parts, namely the forward conduction of the signal and the backpropagation of the error, that is, the error output is carried out in the direction from input to output, and the reverse correction is carried out in the direction from output to input when the weight is adjusted. In the case of signal forward transmission, the input information propagates to the hidden layer through the input layer acting on the weight, and then propagates to the output layer through the same effect on the weight, calculates the error between the expected output and the actual output, and then enters the process of error reverse transmission. The final error is conducted back to the input layer through the hidden layer, and the overall error is distributed to all nodes in each layer, and then the weight of each unit is adjusted to make the error drop in the direction of gradient. After repeated learning and adjustment, the optimal error value is obtained. A four-layer feed-forward neural network is shown in Fig. 1, which contains input layer, two hidden layers, and output layer, and the nodes between each layer are fully connected [13].

Let the input data X of the neural network be in the format of $[x_1, x_2, \ldots, x_n]$, and we use *Wki* to represent the weight of the *k*th node in the previous layer to the *i*th node in the later layer, then the weight *Wi* between each layer can be expressed as

$$w_i = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \cdots & w_{mn} \end{bmatrix} \tag{1}$$

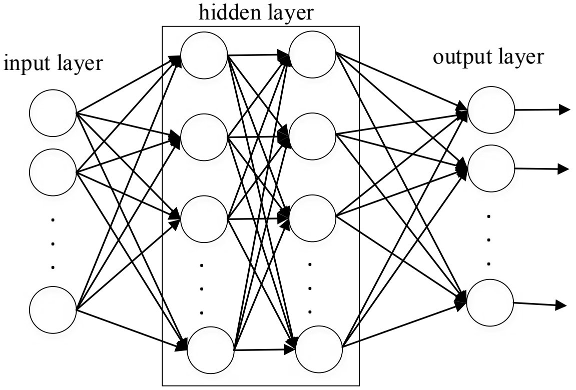


Fig. 1. Four-layer feed-forward neural network

In order to add nonlinear elements to the neural network, *ReLU* (modified linear element) is adopted as the activation function, and its expression is

$$ReLUx = \begin{cases} x & if \quad x > 0 \\ 0 & if \quad x \leq 0 \end{cases} \tag{2}$$

Thus, the output $O_i^{(l)}$ of the *i*th unit in the I layer is

$$O_i^{(l)} = ReLU\left(\sum_{j=1}^{n} w_{ij}^{(l-1)} x_j + b_i^{(l-1)}\right) \tag{3}$$

where *B* is the bias value, and $b_i^{(l-1)}$ is the *l*−1 bias value in the *i* layer.

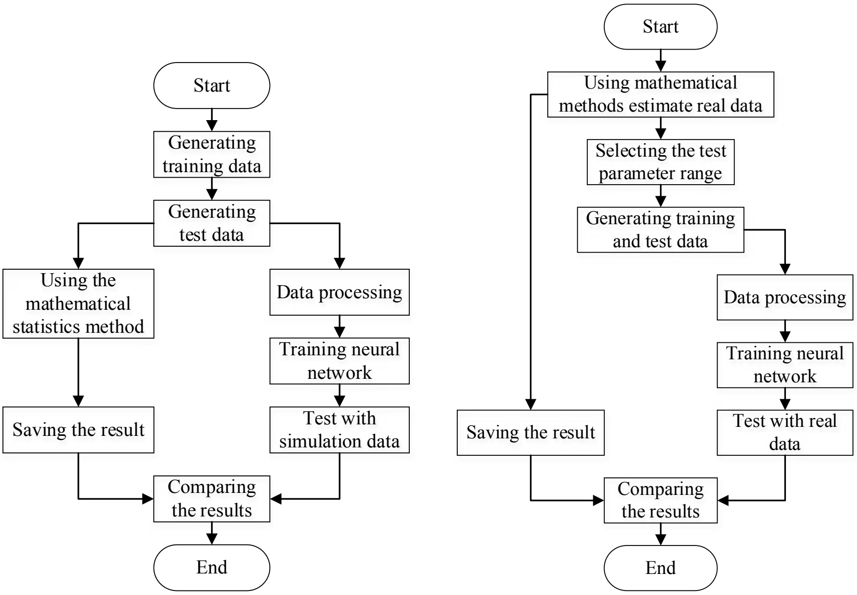
The loss of a single sample adopts mean square error, and its expression is

$$\text{loss} = \sum_{i=1}^{k} (p_i - t_i)^2 \tag{4}$$

where p_i is the sample predicted value, t_i is the sample real value, and *k* is the number of neurons in the output layer.

2.2 Estimation Method for Simulated Sea Clutter Data

The fully connected neural network is adopted, and the processed sea clutter data (simulation data) is taken as input data. After the multilayer fully connected neural network, the influence parameters of the sea clutter model were obtained, and then the parameters were substituted into the sea clutter model as the cost function feedback to carry out the reverse propagation, adjust the weight, and re-input the sea clutter data for training and calculation. The higher the fitting degree is, the more consistent the data is with the model. Until the parameters are more ideal, the parameters of the clutter model



(a) Flowchart for simulated data (b) Flowchart for measured data

Fig. 2. Parameter estimation flowchart

are estimated. Finally, the error between the theoretical model and the empirical fitting result is calculated. The flowchart is shown in Fig. 2a, which consists of the following steps.

(1) The dataset is constructed to generate pre-training data specified by parameters for training the neural network. The prediction test data with the same parameters and formats as the training data is generated to test the neural network, which is used to compare the difference between the estimated results of the neural network and the mathematical statistical methods.
(2) The generated pre-training data and predictive test data are processed. Histogram statistical method was adopted to select the appropriate range size and overall numerical range, and the processed data was used as training and test data.
(3) The parameters of all the predicted test data are estimated by mathematical statistics method, and the shape parameters and scale parameters of each data are estimated as one of the final comparison results, which is called the estimation result of mathematical statistics method.
(4) Input processed training data to train neural network. One network trains one parameter, and multiple networks correspond to multiple parameters. The trained network model corresponding to the parameters is obtained.

(5) The input test data is used to test the trained network model, and the estimated results of the neural network are obtained. The results are compared with the estimated results of mathematical statistics methods, and the average deviation of each sample label is used for evaluation. The deviation formula is

$$\partial_{\text{bias}} = \frac{\sum_{1}^{n} |L - L_{\text{label}}|}{n} \tag{5}$$

where n is the total number of data, L is the estimated result, and L_{label} is the actual label (the parameter when data is generated).

2.3 Estimation Method for Real Sea Clutter Data

Because there is no real tag for real sea clutter data and no supervised learning is possible, this paper proposes a method of training neural network model with simulation data and testing with real sea clutter data. The flowchart is shown in Fig. 2b, which consists of the following steps.

(1) An appropriate mathematical statistical estimation method is selected to estimate the real sea clutter data, and the estimation result of mathematical statistical method is obtained.

(2) The test parameters are selected to eliminate the results that are too large or too small in the parameters obtained by the mathematical statistics estimation method. Select a distribution parameter range with more real data as the test range. When selecting the parameter range, the training time and other issues should be considered, and the parameter range cannot be too large, otherwise there will be too many training data with different parameter combinations, or the span of training parameters will be too large after discretization, resulting in the decline of the neural network fitting effect.

(3) The parameter range is discretized and divided into several uniformly distributed values, and the corresponding simulation pre-training data is generated according to these discrete values. It should be noted in this step that if multiple parameters are to be estimated, a combined training network with multiple parameters and different values should be used, so that the network has the effect that the estimation of one parameter is not affected by other parameters.

(4) The pre-training data (simulation data) and real data are processed by histogram statistical method with appropriate numerical range and interval length.

(5) The neural network is trained with training data (simulation data), and the real data is predicted after the training is completed. The estimation results of real data by neural network are obtained, and the chi-squared test value is calculated. The chi-squared test value of the neural network results is compared with the chi-square test value of the result estimated by mathematical statistics method to verify the quality of the algorithm.

3 Validation Results

3.1 Results of Simulated Data

Three groups of different K-distribution parameters are, respectively, estimated by training the corresponding full-connected neural network. The simulation data was 4096 dimension, which was processed into 1000 dimension by histogram statistical method, with the numerical range of 0–10 and the interval length of 0.01. The neural network used is two 4-layer deep networks, with 1000 neurons in each input layer, 1000 neurons in each hidden layer and 1 neuron in each output layer. For each parameter, the corresponding neural network was trained.

Table 1. Estimation results of K-distribution parameter

Real parameter value		Mean bias of our method		Mean bias of traditional method	
Shape	Scale	Shape	Scale	Shape	Scale
1.0	0.5	$1.055 * 10^{-4}$	$5.287 * 10^{-5}$	$1.171 * 10^{-2}$	$4.997 * 10^{-3}$
1.0	1.0	$7 * 10^{-9}$	$2.597 * 10^{-5}$	$2.698 * 10^{-3}$	$1.566 * 10^{-2}$
1.0	1.5	$4.02 * 10^{-4}$	$3.124 * 10^{-3}$	$5.66 * 10^{-3}$	$7.736 * 10^{-2}$

The neural network uses the quadrature cost function, AdaOptimizer, learning rate 0.002, batch size 50, and 20,000 iterations. The K-distribution adopts the second-order and fourth-order moment estimation method to estimate parameters, and final parameter estimation results are shown in Table 1. From the mean bias of the estimated parameters, it can be seen that the proposed method has higher accuracy comparing with traditional moment-based method.

3.2 Results of Measured Data

The 1993 IPIX radar data are used for parameter estimation and method validation. After testing, the grouping length is selected as 2000, the range of the *K*-distribution shape parameter is 0.3–0.7, and the scale parameter is 0.8–1.3. According to this range, a total of 30 parameter combinations of training data are generated, including 31,200 pieces of data, each with 2000 dimensions. Four layers of fully connected neural networks are used for training.

Typical result is shown in Fig. 3. In this figure, the empirical probability distribution function (EPDF), K-distribution modeling results with both traditional estimation method and the proposed method, as well as lognormal fitting results are provided. It can be seen that K-distribution fit result with the deep neural network method as parameter estimation method shows a better agreement with the EPDF, which indicates the superiority of the proposed parameter estimation method further.

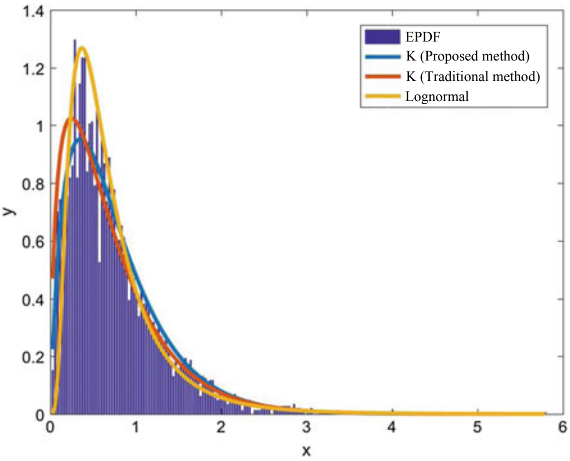


Fig. 3. Validation results with IPIX sea clutter data

4 Conclusions

In this paper, deep neural network model is built to estimate the parameters of amplitude distribution models. In order to solve the poor fitting problem when neural network is applied to the original data, the histogram statistical method is adopted to reduce the data dimension, thus enhancing the data correlation and increasing the value density of data. What's more, in order to solve the problem that real sea clutter data do not have accurate tags, this paper proposes the method of training neural network with simulation data instead of real data. Both simulation data and X-band radar-measured sea clutter data are used to validate the method, and the results indicate the superiority of the proposed parameter estimation.

Acknowledgements. This work was supported in part by the China Postdoctoral Science Foundation (2017M620862) and the National Natural Science Foundation of China (61871392, 61531020, 61871391).

References

1. Hao D, Yunlong D, Ningbo L et al (2016) Overview and prospects of research on sea clutter property cognition. J Radars 5(5):499–516
2. Wen H (2015) Parameter estimation of sea clutter amplitude distribution and target cell collection method. Xidian University, Xian
3. De Wind HJ, Cilliers JC, Herselman PL (2010) Sea clutter and small boat radar reflectivity databases. IEEE Signal Process Mag 32(2):145–148
4. Zhiguang S, Jianxiong Z, Fu Q (2007) Parameter estimation of K-distributed sea clutter. Signal Process 23(3):420–424
5. Jubo Z, Fu Y, Wendong G (2005) Analysis of sea clutter statistical characteristics. Modern Radar 27(11):4–6

6. Ward K, Watts S (2010) Use of sea clutter models in radar design and development. IET Radar Sonar Navig 4(2):146–157
7. Xie CQ, Yang JL (2008) A study on the time-frequency distribution of radar sea clutter spike. Modern Radar 30(5):10–13
8. Guan J, Zhang J, Liu NB (2009) Time-frequency entropy of Hilbert-Huang transformation for detecting weak target in the sea clutter. IEEE Radar Conference 2009:1–5
9. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 25(2)
10. Sun ZJ, Xue L, Xu YM et al (2012) Overview of deep learning. Appl Res Comput 29 (8):2806–2810
11. Yue H, Dongyang L Hua, Kui H et al (2019) Overview on deep learning. CAAI Trans Intell Syst 14(1):1–19
12. Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
13. Ling Z, Bo Z (1994) On the BP algorithm of feedforword neural networks. PR&AI 7 (3):191–195

Maritime Target Trajectory Prediction Model Based on the RNN Network

Jialong Jin, Wei Zhou(✉), and Baichen Jiang

Naval Aviation University, Shandong 264000, China
jjldwb@sina.cn, yeaweam@163.com, 704011136@qq.com

Abstract. The number of ships has increased in recent years and the requirements for maritime security protection have become more stringent. It is important to improve the prediction accuracy and efficiency of maritime target in order to sustain maritime security. According to the real-time, efficiency and accuracy requirements of maritime target trajectory prediction, a prediction model based on RNN network is proposed to realize maritime target trajectory prediction based on AIS data. This paper uses AIS data between Longkou and Dalian to conduct experiments, and compares the prediction effects of two network models RNN and LSTM on the maritime target trajectory. It proves that the RNN network model has higher prediction accuracy and stronger learning ability. Based on the experimental results, this paper analyzes the characteristics of deep learning in long-term prediction and historical data dependence at the same time.

Keywords: Trajectory prediction · Deep learning · Automatic identification system (AIS) · RNN · LSTM

1 Introduction

With the increasingly complex maritime situation, maritime security needs are increasing. As an important means to ensure maritime security, the trajectory prediction of maritime targets has been widely studied. At present, a lot of research has been done on target trajectory prediction, Gambs [1] proposed an extended Mobility Markov Chain in order to predict next place based on the n previous locations visited. Its accuracy can reach from 70 to 95%. But it cost a lot in terms of computation and space. Song [2] finds that a 93% potential predictability to predict the spread of human with measuring the entropy of each individual's trajectory. Based on the findings, Qiao [3] proposed to predict target's trajectory with Hidden Markov Model. But he does not consider the problem of efficiency under the background of Big Data. Then, he proposed a new dynamic trajectory prediction algorithm [4] based on Kalman filter [5], which can guarantee time performance. Besides, there are also many deep learning algorithms [6–8], but most of them focus on visual prediction. Quan [9] used LSTM network to predict trajectory based on AIS data and proved that it is better than BP network. This paper proposed a simplified RNN network, which is simpler than LSTM and behaves better.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 334–342, 2020.
https://doi.org/10.1007/978-981-15-0187-6_39

2 RNN Network Structure

Recurrent neural network is a kind of network model for processing sequential data, which is widely used in text processing, speech recognition and image tracking. The network design pattern adopted in this paper is the recurrent neural network expansion about time, which has a single output at the end of the sequence, as shown in Fig. 1.

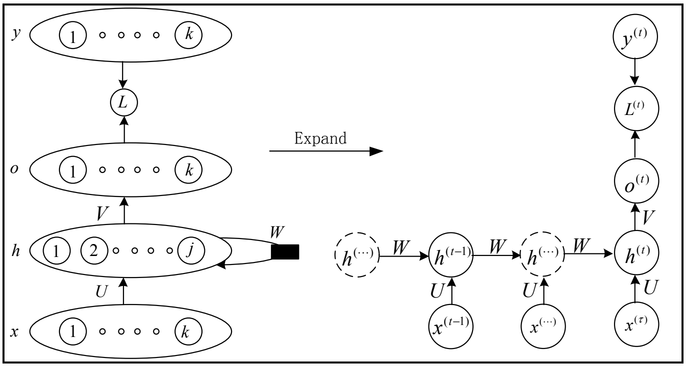


Fig. 1. RNN model

The basic RNN network structure includes input layer, hidden layer and output layer. The parameters sharing mechanism of the recurrent neural network makes the calculation graph of the network can be extended continuously and adapt to longer sequence samples of different lengths. Figure 1 shows the RNN network model adopted in this paper.

In Fig. 1, the right network figure shows the recurrent neural network expansion about time, τ is timestep. $L^{(t)}$ is the value of loss at time t, $x^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_k^{(t)})$ is the input vector at time t, which has a size of $1 \times k$. j is the number of cells in hidden layer, which has a full connect with input layer. The output vector is $h^{(t)} = (h_1^{(t)}, h_2^{(t)}, \ldots, h_j^{(t)})$, which has a size of $1 \times j$. The output layer has the same dimension as the input layer, outputs a vector $o^{(t)} = (o_1^{(t)}, o_2^{(t)}, \ldots, o_k^{(t)})$. The previous hidden layer connects with the hidden layer at next time point by matrix W and forms a recurrent network.

Network propagation model is the mathematical representation of network structure, including forward propagation model and back propagation model.

Forward propagation model as follow:

$$h^{(t)} = \tanh(W \cdot h^{(t-1)} + U \cdot x^{(t)} + b_i) \tag{2.1}$$

$$o^{(t)} = V \cdot h^{(t)} + b_o \tag{2.2}$$

$$L^{(t)} = f(o^{(t)}, y^{(t)}) \tag{2.3}$$

where f is the loss function.

The purpose of backpropagation is to update the network weights and biases to minimize the value of loss function. The size of weight update is determined by the learning rate of the network. Large learning rate will make the network model unstable or fall into local optimal solution. Small learning rate will lead to the decrease of network learning rate and slow convergence.

At present, many algorithms are used to optimize the backpropagation process; for example, SGD, Momentum, Nesterov, Adgrad, Adam, etc. The optimization algorithm adopted in this paper is Adam [10].

3 Maritime Target Trajectory Prediction Model Based on the RNN Network

3.1 Data Selection

Automatic identification system (AIS) is an integrated digital navigation aid system composed of shore-based facilities and ship-borne equipment. AIS data can provide static, dynamic data of the ship, which is widely used in the fields of position report, beacon telemetry, VTS [11] and ship collision avoidance [12]. In the literature [9], the velocity, course, longitude, latitude and time interval of the maritime target are adopted as input of the network. According to the characteristics of AIS data and the requirements of network training, this paper selects the longitude, latitude and time interval of the maritime target as input, mainly based on the following five points:

i. Fault tolerance to velocity and course is large.
ii. Speed and course can be estimated from the position information.
iii. Speed and course will reduce the accuracy considering the global loss.
iv. Increase dimension of input data leads network complex and difficult to train.
v. Speed, course from AIS are not accurate enough to be used as training input.

3.2 Data Normalization

In order to reduce the error caused by the difference of magnitude and dimension, the input data is normalized. In this paper, the deviation standardization method [9] is adopted, and the calculation formula is:

$$X^* = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{3.1}$$

Data normalization will not change the characteristics and distribution of data, and the prediction results of the RNN network model need to be de-normalized according to Formula 3.1, so that the data can be consistent with the practical significance.

3.3 Model Construction

According to the RNN network structure in Sect. 1.2 and the actual application characteristics of maritime target trajectory prediction, in order to improve the prediction accuracy of the network, the appropriate timestep, the number of units in hidden layer and the loss function need to be selected.

I. Timestep Selection

Timestep is an important parameter of time sequence prediction, which determines the amount of accumulated historical information. The selection of timestep affects the prediction accuracy of the network, small timestep will lead to insufficient learning of trajectory characteristics, and large timestep will lead to gradient disappearance (explosion). In this paper, the target trajectory data at the 3–7 moment is selected as input, the timestep is 3–7.

II. Selection of the Number of Units in Hidden Layer

The number of units is generally determined by the dimension of input data, which is generally double dimension. Too few units leads to insufficient feature extraction, which requires more training times and lower prediction accuracy. Too many units will lead to overfitting. In this paper, the longitude, dimension and time interval of the maritime target are selected as input, so the number of units in hidden layer is selected as 3–9.

III. Evaluation Index Selection

The evaluation index of the experiment determines the criterion for selecting the optimal parameters and evaluating the prediction effect of the network. The evaluation indexes for the test set are the sum of square error (SSE) and mean absolute error (MAE), they reflect the dispersion and prediction accuracy.

The formulas of evaluation index are as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left(y^{(i)} - o^{(i)}\right) \tag{3.2}$$

$$\mathrm{SSE} = \sum_{i=1}^{N}\left(y^{(i)} - o^{(i)}\right)^2 \tag{3.3}$$

IV. Loss Function Selection

As an important part of network training, loss function is the learning criterion of the whole network. Loss function reflects the inconsistency between predicted value and real value. The smaller the value of the loss function is, the better performance of the model will be. In this paper, Absolute loss (Formula 3.4) and Square loss (Formula 3.5) functions with good results are selected for comparison. Take RNN network under different timestep as an example.

$$L^{(t)} = \frac{1}{k}\sum_{i=1}^{k} \left| y_i^{(t)} - o_i^{(t)} \right| \tag{3.4}$$

$$L^{(t)} = \frac{1}{k}\sum_{i=1}^{k} (y_i^{(t)} - o_i^{(t)})^2 \tag{3.5}$$

Contrast test adopted RNN network, the number of units in hidden layer is 5, and the timestep is 3–7. It can be seen from Table 1 that the test results with Absolute loss as the loss function are within the acceptable range, and the test results with Square loss as the loss function are seriously deviated from the true value, and the Absolute error of longitude and latitude reaches above 0.1°. Therefore, this paper adopts Absolute loss as the loss function to conduct the simulation experiment.

Table 1. Loss function contrast

Loss function	Absolute loss		Square loss	
Timestep	SSE	MAE	SSE	MAE
3	0.116	0.048	0.278	0.143
4	0.115	0.037	0.369	0.265
5	0.114	0.035	0.350	0.212
6	0.117	0.041	0.943	0.575
7	0.118	0.042	0.405	0.260

4 Model Simulation and Analysis

4.1 Model Simulation

The AIS data used in this paper is the passenger ship trajectory data between Longkou and Dalian, whose MMSI is 413324830, and the time is from 10:00 to 16:54 in April 1, 2019. The first 130 trajectory points are chosen as training set, and the remaining 42 trajectory points are chosen as test set.

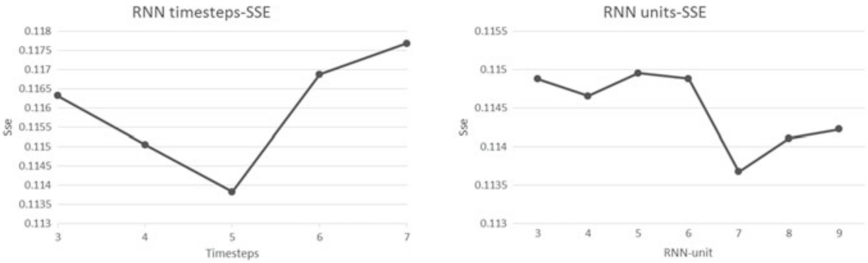
The hardware used in the experiment: the processor was Intel(R) Core(TM) i7-4710MQ CPU @2.50 GHz, and the size of memory was 8 GB. The size of experiment was based on tensorflow version 1.12.0, the programming version was Python 3.6.8, and the integrated environment was pycharm 4.0.4.

4.2 Analysis of Simulation Results

I. Parameter Analysis

Selecting appropriate network parameters can greatly improve the learning performance of the network. In this paper, timestep of the RNN network is selected between 3 and 7, the number of units in hidden is selected between 3 and 9. When timestep analysis is performed, the number of units in hidden layer is the optimal value in the same timestep. When the number of units in hidden layer is performed, the timestep is the optimal value in the same hidden layer.

Figure 2a shows the change curve of SSE in different timesteps, and Fig. 2b shows the change curve of SSE in different number of units in hidden layer. As can be seen from Fig. 2a, SSE decreases first and then increases, and reaches the minimum when timestep is equal to 5. As can be seen from Fig. 2b, two minimum points appear in SSE curve, but SSE with the units number of 7 is obviously less than that with the units number of 4. Therefore, when the timestep is equal to 5 and the number of units is equal to 7, the prediction trajectory of network is optimal.



(a) Timestep contrast

(b) RNN units contrast

Fig. 2. Parameter analysis

II. Model effect analysis

The LSTM network is an improvement on RNN network to solve gradient disappearance (explosion). RNN is a simpler sequence prediction model, and the two networks are compared in terms of whether the trajectory prediction with RNN is better than LSTM for short and low dimensions trajectory.

In this paper, the parameter analysis of the RNN and LSTM is performed, and determine the optimal timestep and the number of units as shown in Sect. 1.4.2(I). Compared with the optimal prediction results of the two networks, the comparison of the evaluation index is shown in Table 2.

Table 2. Evaluation index of models contrast

	SSE	MAE
RNN	0.113	0.036
LSTM	0.115	0.044

As seen from Fig. 3, as time increases, the prediction trajectory of the LSTM is gradually deviating from the real trajectory. On the one hand, the distance between prediction trajectory of LSTM and real trajectory is increased. On the other hand, the

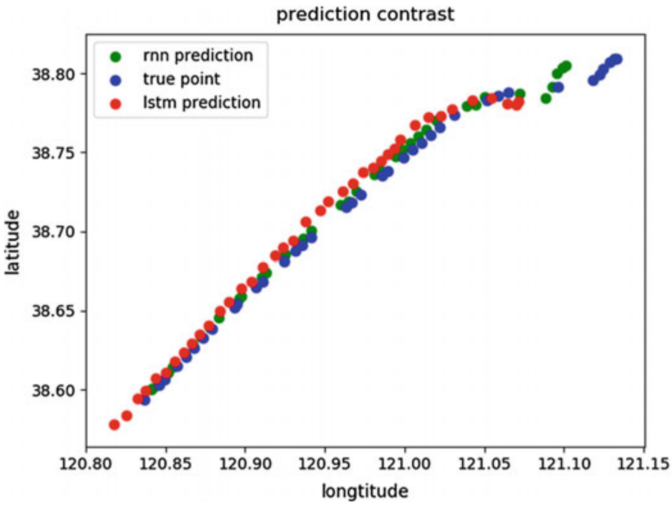


Fig. 3. Models contrast

interval between the points is not consistent with the real trajectory. The prediction trajectory of RNN has a good adhesion degree to the real trajectory. In the longer time prediction, the results of the RNN are closer to the real trajectory. The prediction trajectory of the LSTM is distorted, indicating that the LSTM prediction has lost the details of the historical trajectory.

III. Analysis of the effect of deep learning in trajectory prediction

Figure 3 also shows the common problems of the two models. The network prediction model is getting worse when predicting a long time trajectory.

Take the RNN network prediction model with timestep 5 and number of units 5 as an example. As Fig. 4 shows, although the amplitude of MAE fluctuates, it is on the rise as a whole. The reasons are as follows:

i. Deep learning is based on learning trajectory characteristics of historical trajectory. The increase of time is the increase of trajectory uncertainty. One way to improve the accuracy of long-term prediction is to use short-term target trajectory as training data to iterate.
ii. The prediction of target trajectory by deep learning is greatly affected by historical information. Only by using a large volume of data for training can accurately predict all kinds of the ship trajectory.

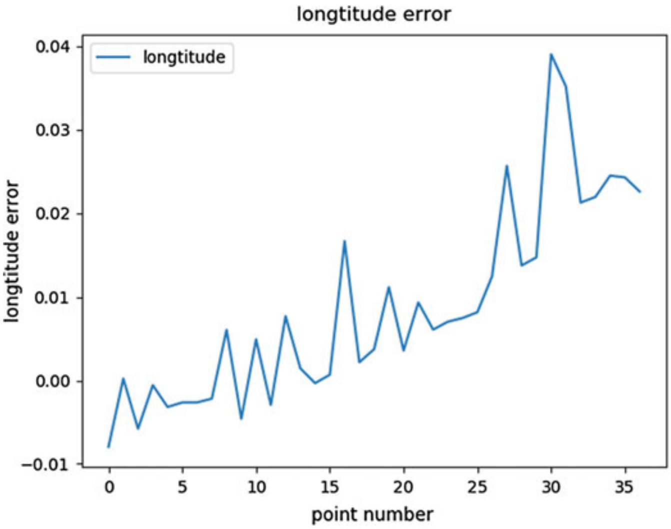


Fig. 4. MAE of RNN for different units

5 Conclusion

Based on the characteristics of AIS information and network training requirements, this paper adopts the longitude, latitude and time interval of target as input to establish a simplified RNN target trajectory prediction model. In this paper, the loss function of the network is optimized through the experimental simulation, and Absolute loss is finally selected as loss function. In order to select appropriate parameters to optimize network, this paper carried out control variable experiments with timestep and the number of units in hidden layer as variables. Finally, RNN network was optimal when the timestep is equal to 5 and the number of units is equal to 7. Under the premise that all network models are optimal, this paper compares the effect of RNN model and LSTM model. Experimental results show that RNN network model proposed in this paper has better prediction effect. Finally, according to the prediction results of RNN and LSTM, the characteristics of deep learning in the field of ship trajectory prediction are analyzed.

References

1. Gambs S, Killijian M, Cortez DP, Mguel N (2012) Next place prediction using mobility markov chains. In: Proceedings of the 1st workshop on measurement, privacy, and mobility, vol 3. ACM, New York, pp 1–6
2. Song C, Qu Z, Blumm N, Barabsi A-L (2010) Limits of predictability in human mobility. Science 327(5968):1018–1021
3. Qiao S, Nan H, Xinwen Z et al (2018) A dynamic trajectory prediction algorithm based on Kalman filter. Acta Electronica Sinica 46(2):418–423

4. Qiao S, Shen D, Wang X, Han N, Zhu W (2015) A self-adaptive parameter selection trajectory prediction approach via hidden Markov models. IEEE Trans Intell Transport Syst 16(1):284–296
5. Kalman RE (1960) A new approach to linear filtering and prediction problems. J Basic Eng 82(1):35
6. Li H, Li Y, Porikli F (2015) DeepTrack: learning discriminative feature representations online for robust visual tracking. IEEE Trans Image Process 25(4):1834–1848
7. Wang N, Yeung DY (2013) Learning a deep compact image representation for visual tracking. In: International conference on neural information processing systems. Curran Associates Inc
8. Wang L, Zhang L, Yi Z (2017) Trajectory predictor by using recurrent neural networks in visual tracking. IEEE Trans Cyber 99: 1–12
9. Quan B, Yang B, Hu K et al (2018) Prediction model of ship trajectory based on LSTM. Comput Sci 45(S2): 136–141
10. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. Comput Sci
11. Kim KI, Lee KM (2017) Pre-processing ship trajectory data for applying artificial neural network in harbour area. In: European conference on electrical engineering and computer science (EECS), Bern, Switzerland, pp 147–149
12. Piotr B (2017) The ship movement trajectory prediction algorithm using navigational data fusion. Sensors 17(6):1432–1444

Industrial Ventilator Monitoring System Based on Android

Junjiao Zhang(✉), Yongjie Yang(✉), and Zhongxing Huo

School of Information Science and Technology, Nantong University,
Nantong 226019, China
1047891636@qq.com, yang.yj@ntu.edu.cn

Abstract. The traditional industrial production site ventilator adopts manual management, which is cumbersome and cannot be dealt with in the first time when abnormal conditions occur. In view of the above situation and demands, an industrial ventilator monitoring system based on Android is designed and implemented. The system takes ESP8266 module as the core, collects smoke value of industrial environment through smoke sensor, establishes server with Raspberry Pi, and develops an app with Android studio as development tool. Through the visual app interface, the on-duty personnel can not only view the status and usage situation of ventilators in different areas in real time, but also remotely control the opening and closing of multiple ventilators and set the timing task of ventilators. In addition, the system also achieves abnormal automatic processing. Once the abnormal smoke value is detected, the ventilator will be automatically opened, which has strong flexibility and high application value.

Keywords: Monitoring system · ESP8266 module · Raspberry Pi · Android

1 Introduction

In the industrial production site, most industrial ventilators are in constant power working condition for a long time, and the waste of electric energy is serious. In addition, the use of manual management of ventilators, abnormal conditions cannot be dealt with at the first time, will cause a certain degree of safety risk.

The Android-based monitoring system for industrial ventilator is designed in this paper. Monitoring data collected by sensors and other hardware circuits are transmitted to Raspberry Pi Server by ESP8266 Wi-Fi module. It also provides an app interface for managers to view current monitoring data and control ventilators, which solves the problem of remote monitoring of ventilators in industrial field and saves time and manpower.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 343–352, 2020.
https://doi.org/10.1007/978-981-15-0187-6_40

2 Overall Scheme Design of the System

The system uses a three-tier Internet of things architecture, from top to bottom is the management layer, transmission layer, and acquisition layer. The overall structure of the system is shown in Fig. 1.

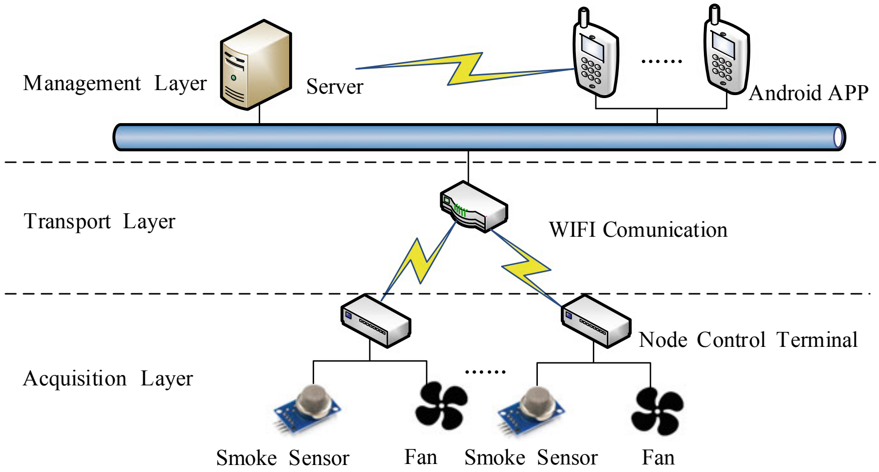


Fig. 1. System structure diagram

Data acquisition layer mainly collects data, using ESP8266 module embedded Wi-Fi and MCU as the main control chip, carrying smoke sensor to collect monitoring data.

As an intermediate bridge, the transmission layer mainly completes the two-way communication between the acquisition layer and the management layer. This system uses Wi-Fi to communicate, avoiding the wiring problem of wired transmission, making data transmission more convenient and stable.

The management layer is composed of server and app. When the server receives the monitoring data transmitted by the data acquisition terminal, it will update and save it in the database. App requests monitoring data from the server constantly, which is used to update the data displayed in the app interface in real time.

3 Hardware Design

Hardware design is the basis of system function realization, which mainly includes: main control chip, power supply circuit, smoke monitoring circuit, relay control circuit, etc. The structure is shown in Fig. 2.

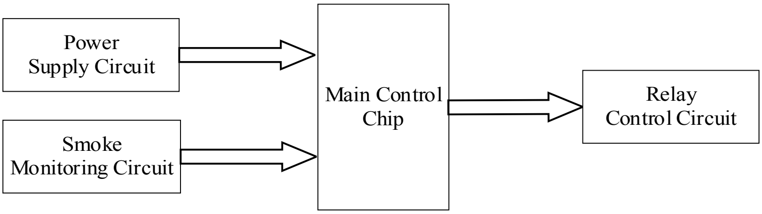


Fig. 2. Hardware design diagram

3.1 Main Control Chip

The ESP8266 Wi-Fi module shown in Fig. 3 is used in the main control chip. It can fully meet the design requirements of the system in terms of performance, power consumption, and cost. ESP8266 module contains front-end Wi-Fi and high-performance 32-bit MCU and has the advantages of rich interface, small size, low power consumption and low price [1]. This module has three working modes: STA, AP, and STA + AP. The system uses STA mode, connecting the Internet through routers, providing access for remote data transmission, and realizing remote control of nodes.

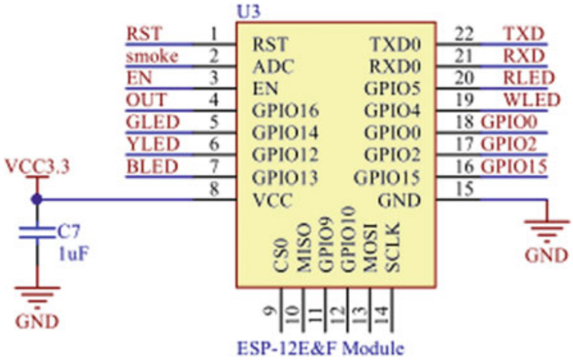


Fig. 3. ESP8266 Wi-Fi module

3.2 Power Supply Circuit

The power supply circuit is the basis of hardware circuit realization. The system uses external 220 V power input. In view of the power supply demand of 5 and 3.3 V in the module of hardware design, the buck circuit is designed as shown in Fig. 4.

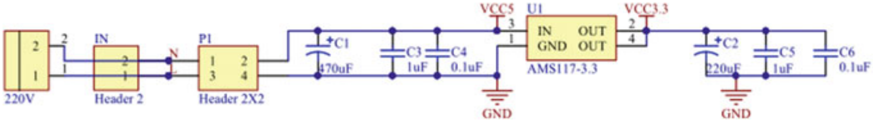


Fig. 4. Power supply circuit diagram

The whole circuit adopts a multistage decompression method to avoid the problem of chip heating caused by excessive voltage drop. 220 V power supply connects with the power input port (IN). The voltage is reduced to 5 V by isolated AC–DC step-down module (P1) and then to 3.3 V by AMS117 (U1). After each step-down, the power supply needs to be filtered to make the generated voltage more stable, so as to supply power to other modules.

3.3 Smoke Monitoring Circuit

Sensor technology, as the main technology in the Internet of things, has become an important means to obtain information in the physical world. In this system, MQ-2 smoke sensor is used to detect the smoke value which is processed and transmitted to the main control chip [2]. The smoke monitoring circuit is shown in Fig. 5. MQ-2 with 5 V power supply input connects voltage follower and resistance to reduce the circuit voltage which does not exceed the 3.3 V voltage required by the main control chip ESP8266.

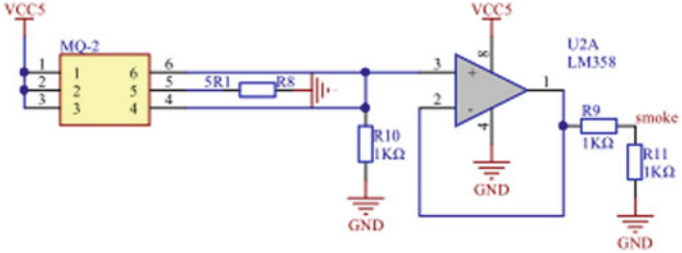


Fig. 5. Smoke monitoring circuit diagram

3.4 Relay Control Circuit

The relay control circuit is shown in Fig. 6. When the circuit input is 0 V, the triode is turned on, and the current passes through the coil of the relay, so that the relay is connected with the switch 3, and the green indicator light is on; similarly, when the input is +VCC, the triode is turned off, and the relay releases the switch 3. The control circuit is externally connected to the ventilator, and the switching of the ventilator can be realized by controlling the input voltage.

4 Software Design

The software design of the system is mainly divided into three parts: the software design of acquisition terminal, server, and App.

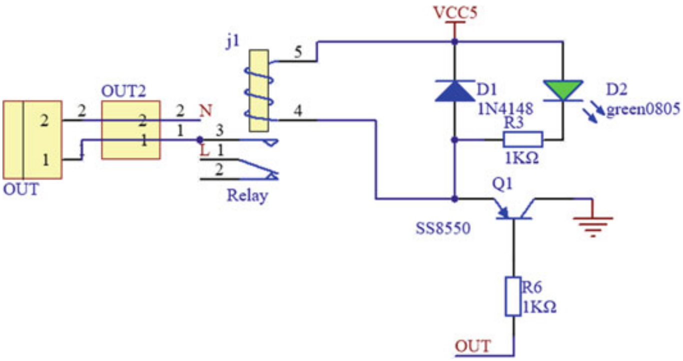


Fig. 6. Relay control circuit diagram

4.1 Software Design of Acquisition Terminal

The software design of acquisition terminal is written in Lua script language, and the design flow is shown in Fig. 7. First, initialization operations are performed, including initialization of IO, timer, Wi-Fi, server connection, etc.

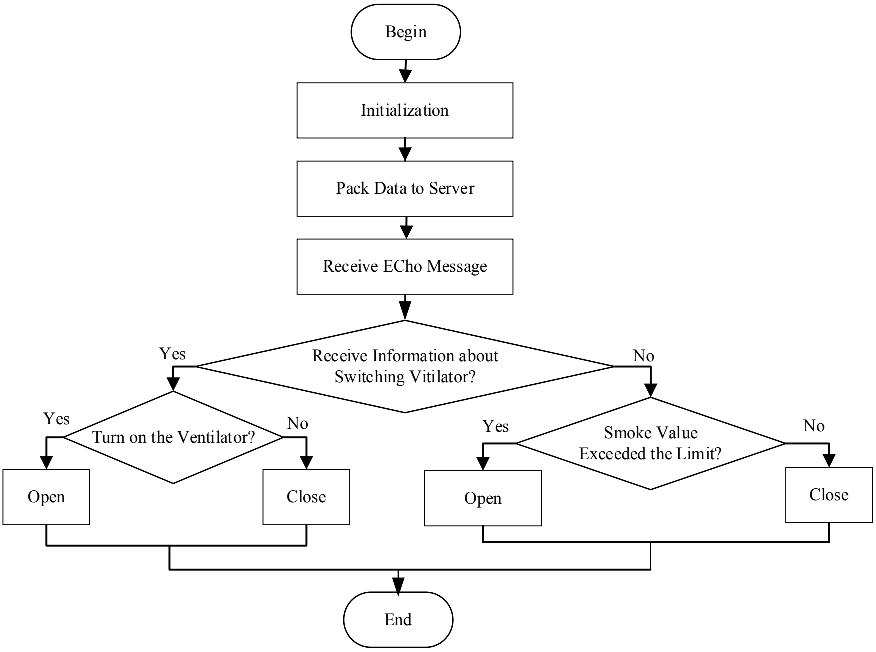


Fig. 7. Flow chart of software design for acquisition terminal

MQTT protocol is used to communicate between acquisition terminal and server [3]. It is a publish/subscribe protocol based on TCP/IP for mobile terminal nodes.

After the initialization is completed, the acquisition terminal packs and sends the data to server. Meanwhile, the acquisition terminal continuously receives information from server and then performs corresponding control operations after parsing. Once the acquisition terminal finds out that the smoke value is abnormal and has not received any operation instructions from server, it will automatically turn on the ventilator.

4.2 Software Design of Sever

The server is served by Raspberry Pi, an embedded system. The software program of server is written in Java language and uses eclipse as the development platform. The functions of server include the interaction with the acquisition terminal, app and database.

When server communicates with app, it first declares a ServerSocket object and specifies the port number, then calls accept() method to listen for the connection request, and returns a Socket object after receiving the request [4]. After the connection is successful, the output and input streams are obtained through the getInputStream() and getOutputStream() methods of the Socket object.

The server-side database uses MYSQL lightweight relational database. The “region,” “node,” and “timing_task” database table is established according to the requirements of the system. After the connection between Java program and database is established by JDBC, the data in the database can be added, deleted, and modified by using SQL statement.

4.3 Software Design of Android Terminal

The software design of Android uses Android studio as the development environment and Java object-oriented language as the development language. The main program mainly includes: XML page layout, server connection, data transfer, etc. The software design flow chart is shown in Fig. 8.

XML layout combines multiple controls with multiple layout managers to generates the login interface, the region interface, node interface, node details interface, timing task interface, and so on [5]. By monitoring the login button of the login interface, the submitted login information is sent to server. If the validation success message from server is received, it will enter the next interface through the startActivity() method. The interface of node and node details request the server every 10 s through the postDelayed (task, 10,000) method of Handler to refresh the monitoring information regularly. When activity calls the onStop () method, the timer is closed through the removeCallbacks (task) method of Handler.

Android as a client connects to server through Socket based on TCP/IP protocol. Every time Android establishes a connection with the server, it is necessary to specify the IP address and port number of the server, then call getOutputStream() and getInputStream() methods to obtain the output stream and input stream.

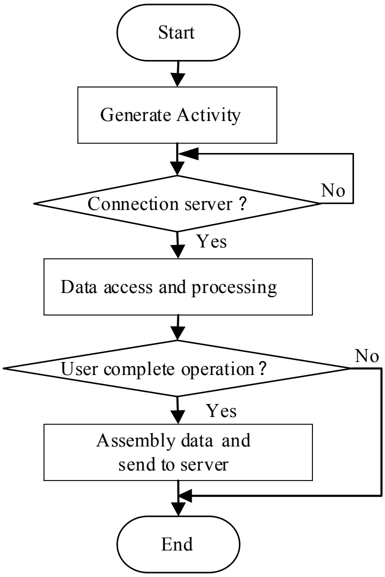

Start
Generate Activity
Connection server ?
No
Yes
Data access and processing
User complete operation ?
No
Yes
Assembly data and send to server
End

Fig. 8. Android software design flow chart

5 Test

The hardware physical diagram is shown in Fig. 9.

The login interface is shown in Fig. 10. Enter the correct login information and click login button, enter the regional interface, as shown in Fig. 11. Click the "Test" column of the regional interface to enter the node interface, as shown in Fig. 12. In the node interface of "Test" (Fig. 12), click the timing task on the menu bar to enter the

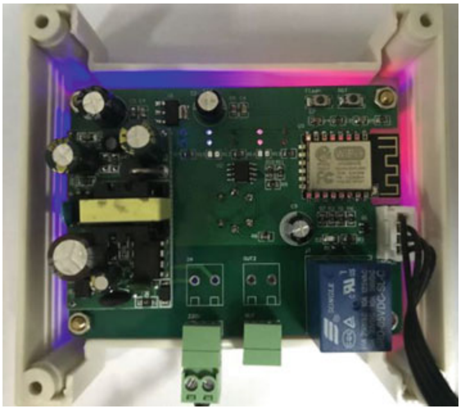
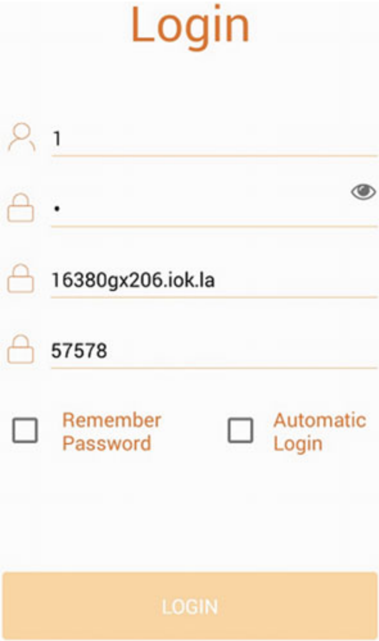


Fig. 9. Hardware physical diagram



Login

1

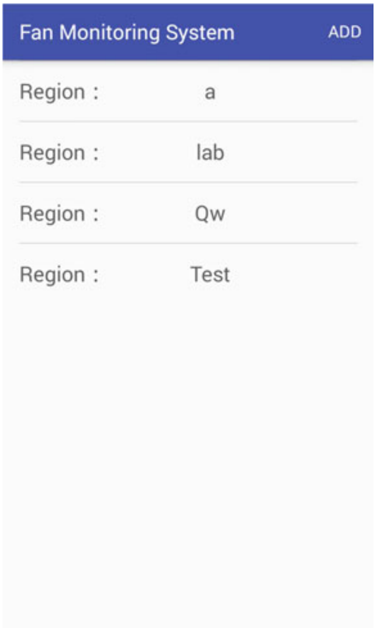
•

16380gx206.iok.la

57578

□ Remember Password □ Automatic Login

Fig. 10. Login interface



Region : a

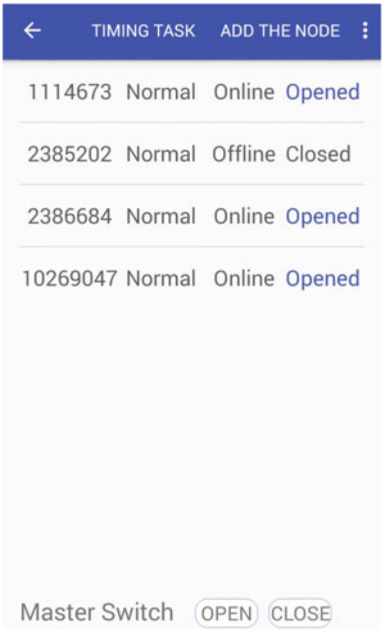
Region : lab

Region : Qw

Region : Test

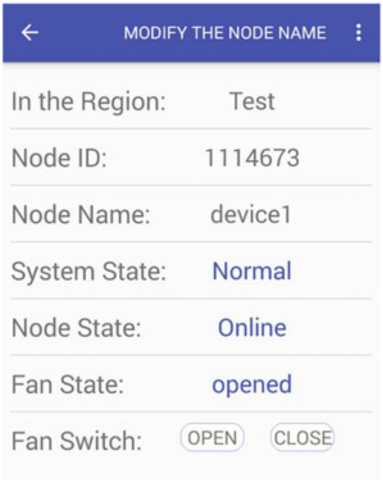
Fig. 11. Region interface

timing task interface as shown in Fig. 13. This interface shows all the timing tasks of "Test." In the node interface (Fig. 12), click the node whose id is 1114673 and enter the node details interface as shown in Fig. 14.



1114673	Normal	Online	Opened
2385202	Normal	Offline	Closed
2386684	Normal	Online	Opened
10269047	Normal	Online	Opened

Fig. 12. Node interface



In the Region: Test

Node ID: 1114673

Node Name: device1

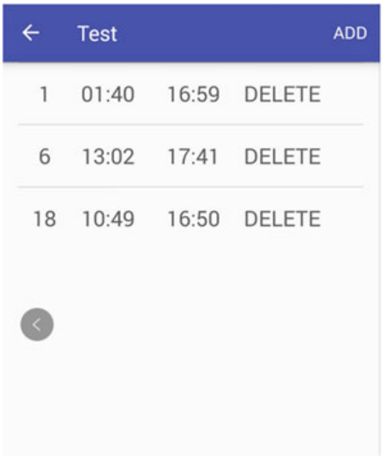
System State: Normal

Node State: Online

Fan State: opened

Fan Switch:

Fig. 13. Timing task interface



← Test ADD

1	01:40	16:59	DELETE
6	13:02	17:41	DELETE
18	10:49	16:50	DELETE

Fig. 14. Node details interface

Test results show that, the data displayed in the app interface and the state of the object are identical. In addition, switching a single ventilator or all ventilators in a certain region and setting the regional timing task, the ventilator can be linked.

6 Conclusions

Aiming at the energy saving of industrial ventilator and the safety problem of industrial production environment, this paper develops an industrial ventilator monitoring system based on Android, which combines MQ-2 smoke sensor and ESP8266 Wi-Fi module to realize smoke numerical acquisition and data communication, and designs Android app to realize the visualization display and convenient control of ventilator status. When abnormal situation occurs, it can be dealt with in the first time. This system solves the problem of energy saving and safety of industrial fan well and has strong practicability.

References

1. Bhatt A, Saxena A, Chauhan S et al (2018) Android-based home automation using Bluetooth and ESP8266
2. Zhang QQ, Zhao QN (2015) Design of fire alarm system based on MQ-2 sensor. Value Eng
3. Guan QY, Hong-Bin LI, Bo YU (2014) Research and application of the MQTT protocol on the android platform. Comput Syst Appl
4. Muddebihalkar V, Gaudar RM (2014) Fast remote data access for control of TCP/IP network using android mobile device. Int J Eng Res Appl 4(4):59–64
5. Li-Jing BI, Jiang BC, Wang XL et al (2017) Real-time vehicle monitoring system based on android. Comput Knowl Technol

An Overview of Blockchain-Based Swarm Robotics System

Yanchang Du(✉), Jiang Cao, Junsong Yin, and Shuang Song

PLA Academy of Military Science, Beijing 100091, China
duyanchang198@163.com, 48796447@qq.com

Abstract. As a disruptive technology, swarm robotics is developing everyday, and attracts great attention. However, swarm robotics face some challenges which hinder swarm robots from broader application. Blockchain technology can provide a basic credible information environment of swarm robots, expand its application. This paper discusses how blockchain technology can provide benefits to swarm robotics, give a basic structure of swarm robots-oriented blockchain conceptual model. Finally, limitations and possible future problems that arise from the combination of these two technologies are described.

Keywords: Blockchain · Swarm robots · Information security · Distributed decision-making · Behaviors collaboration

1 Introduction

Swarms robots is an autonomous system of multi-robots that acts in a self-organized way. It is supposed that a desired collective behavior emerges from the interaction between the robots and the interaction of robots with the environment [1]. Swarm robotics roughly divides into two kinds of swarm: heterogeneous swarm and homogeneous swarm [2].

Usually swarm robotics doesn't rely on any centralized node for communication, collective decision-making, and collaboration. Autonomy, self-organization, high redundancy are mainly characteristics of swarm robotics to make them have a strong adaptability to severe and hostile environments, and have the potential to many applications in target searching, geology survey, and complex military operation. Swarm robots technology has been seen as a game-changing military power by US military and will change the war form in the future.

But, as other distributed autonomous systems, swarm robotics face some challenges in communication, information security, distributed decision-making, global acknowledge collaborative behavior, which hinders swarm robots from broader application. There is a critical need to develop a secured, trusted, and decentralized architecture for swarm robotics.

As an emerging technology, blockchain has usually been seen as a tamper-proof decentralized system used as database and computing platform, also can be seen as an overlay network that can connect nodes to form a distributed network. It can ensure the integrity, consistency, credibility of transaction data in untrustworthy and complex

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 353–360, 2020.
https://doi.org/10.1007/978-981-15-0187-6_41

environment, give the robots global information which makes it be a basic credible information environment of swarm robots, expands the application of swarm robots.

2 Blockchain: Concept and Properties

Blockchain idea originates from the foundational article entitled "Bitcoin: a peer-to-peer electronic cash system" in 2008 written by "Satoshi Nakamoto" [3]. Blockchain can be narrowly defined as a kind of decentralized shared ledger that uses chronological, encrypted and chained blocks to store verifiable and synchronized data across distributed networks. The basic structure of block illustrates in Fig. 1.

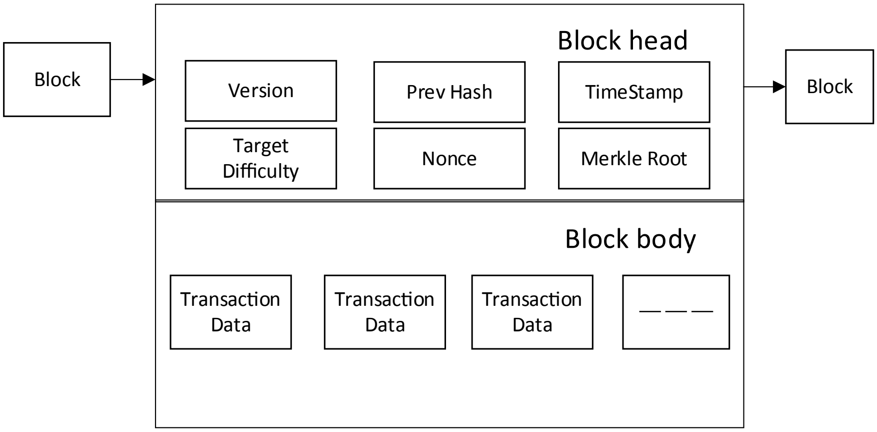


Fig. 1. The basic structure of block

Blockchain utilizes distributed storage, distributed network, consensus mechanism, encryption algorithm, and other technologies. It has the characteristics of decentralization, openness, autonomy, and tamper-proofing. These features make blockchain attract more and more attention, and been regarded as one of the key technologies for the future Internet. The first application of blockchain is "Bitcoin", which was a most popular cryptocurrency. Later, in 2014, the Ethereum was released [4], which can run arbitrary Turing-complete applications via smart contracts, which make Ethereum can run other specific, non-financial applications (e.g., voting, identity management, and so on). Another major application of the blockchain is Ethereum. By now, Blockchain is evolving into the underlying IT infrastructure, supporting multiple blockchains to realize enterprise-level application. In addition to being applied to financial markets, blockchain can also be applied to intellectual property, intelligent transport systems, internet of things and cloud security.

3 Application Prospect

As distributed systems, blockchain-based robot swarm system emphasizes the autonomous nodes that have low-frequency interaction between nodes. At present, blockchain-based swarm robot system is in the infancy. Researchers have done various research on introduction of blockchain technologies to robotic systems. Work conducted by Eduardo Castelló Ferrer presents the benefits of combining the blockchain [5]. The authors of RoboChain presented a framework to tackle privacy issues regarding using personal data by robots during a human–robot interaction [6]. Jason A. Tran, et al. propose a novel protocol, SwarmDAG, that enables the maintenance of a distributed ledger-based on the concept of extended virtual synchrony while managing and tolerating network partitions [7]. Overall, the benefit from blockchain-based swarm robots system will mainly focus on information safety, distributed decision-making, and collaborative behaviors.

3.1 Information Safety

As a network-enabled system, swarm robotics hardly ensures the safety and reliability of information transmission in a complex environment. Higgins et al. present comprehensive survey of security challenges in swarm robotics [8]: tampered swarm members or failing sensors; attacked or noisy communication channels; loss of availability.

Blockchain can provide reliable peer-to-peer communication with security measures over a trustless network, introduce a way to trust the data, trust other participants, provision of core services such as data confidentiality, data integrity, entity authentication, and data origin authentication.

Blockchain can provide a reliable asynchronous communication mechanism in distributed network, assure all the nodes in the network can receive information that been verified. In the blockchain encryption scheme, techniques such as public key and digital signature cryptography are accepted means of not only making transactions using unsafe and shared channels but also of proving the identity of specific agents in a network. Blockchain uses the public key and private key generated by the hash algorithm to provide the communication address and identity and uses asymmetric encryption and digital signature to implement encrypted transaction. The unique distributed architecture, consensus mechanism chain-based data structure ensures that the reliable and trusted transmission of operational information, in the environment of failure or destruction of some communication nodes.

Through consensus protocol, all the transaction information will be verified by all the nodes in the network to assure correction, once the information has been blocked in the blockchain, with hash algorithm and time-stamp technology, the data have the feature of veracity, tamper-proof, traceability, Protect against malicious attack.

3.2 Distributed Decision-Making

Distributed decision-making tasks are divided into consensus achievement problems and task allocation problems [9]. Distributed decision-making algorithms have played a crucial role in the development of swarm systems. Limited by the environment and interaction mechanisms, the swarm robots system is often difficult to obtain global information through a centralized control node to form collaborative decision-making.

Blockchain ensures all participants in a decentralized network share identical views of global information, any node can give collective decision opinion. With smart contract-based voting system, the swarm robot system will achieve decision-making consistency, and can also use MultiSig technologies to achieve decision-making. For the decision made by Swarm robotics, a new joined node automatically synchronizes the blockchain data, know the previous decision, and operation accordingly.

3.3 Behaviors Collaboration

As a programmable distributed infrastructure, blockchain provides a basic information environment for the autonomous task coordination of robots swarms systems. Robot swarms may only operate the same behavior, it is also possible that different nodes perform different activities to accomplish a certain goal.

When different nodes in a robot swarm have different behaviors, we can use Smart Contracts to represent different behaviors. In addition, it is also possible to characterize an activity by using blockchain and hierarchically link different blockchains using sidechain techniques [10], which would allow robotic swarm nodes to act differently according to the particular blockchain being used, where different parameters can be customized for different swarm behaviors.

4 Swarm Robotics-Oriented Blockchain Model

As distributed systems, the combination of robotic swarm systems with blockchain can provide the necessary capabilities to make robotic swarm operations more secure, autonomous, and flexible.

This section presents a Swarm robotics-oriented, seven-layer conceptual model for the typical architecture and major components of blockchain systems. This model will form a blockchain-based information environment for swarm robotics (Fig. 2).

Physical layer. This layer is platform and environment that the blockchain runs, which encapsulates various kinds of physical entities (e.g., UUV, UTV, UAV), store resource, computing resource, network infrastructure, and operation system, etc.

Data Layer. Ledge module provides the chained data blocks, together with the related techniques including asymmetric encryption, time-stamping, hash algorithms, and Merkle trees. The cryptographic module provides basic cryptographic algorithms for other components, as well as key maintenance and storage; the data storage module uses specific databases, such as LevelDB, MySQL, etc. to store ledger data, user data, and so on. The user management module is used to implement access control and node identification authentication.

Network layer. This layer completes the data transmission with other nodes by discovering neighbor nodes, establishing communication links, and provide node

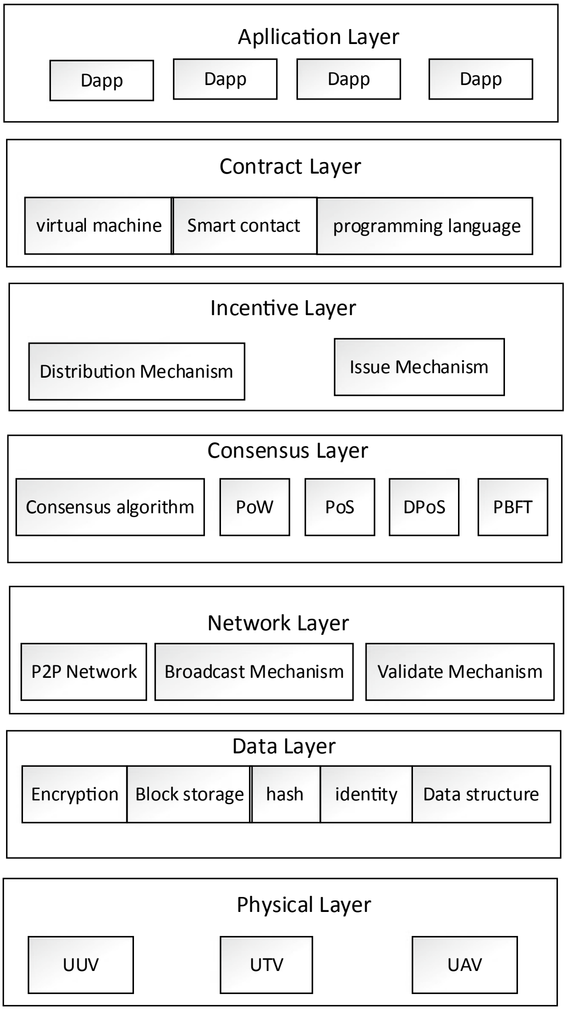


Fig. 2. The basic model of swarm robotics-oriented blockchain

authentication. In blockchain-based swarm robot system, there is usually no fixed communication hub. A direct communication link is established between nodes, and each node has routing function. The robots are not always close enough to communicate with each other.

Consensus layer. This layer is mainly used to maintain node data consistency, have the function of node voting, data consensus, data synchronism. So far, the main consensus mechanism includes POW, POS, DPOS, PBFT, BFT. The consensus mechanism should be modular in design and adopt different consensus algorithms based on different application scenarios. The consensus mechanism will also affect the logical network topology presented at the network layer. In the application scenarios of robots swarm system, limited by communication environment, it is difficult to achieve global consensus, often reach local consistency. Collaborative decision-making should achieve by local consistency.

Incentive layer. This layer is mainly used for value rewards, which gives the blockchain-based Swarm robots Economic attribute, make the swarm robots network a value network.

Contract layer. The application of smart contracts is still in its infancy, and it is also the hardest-hit area for blockchain security. Swarm robots mainly use blockchain-based smart contracts to realize data transmission and complete task coordination. Different smart contracts can accomplish different missions. A node can deploy multiple smart contracts depending on the task.

Application layer. Distributed Application (DApps) is used to customize different types of activities. The DApps provide different services to users by calling smart contracts and adapting various application scenarios of the blockchain.

It should be pointed out that in a small and highly trusted scenario, a private chain can be constructed, adopting a minimal blockchain architecture, for example, removing the incentive layer and the verification mechanism, and simplifying the consensus mechanism and block structure as shown in Fig. 3.

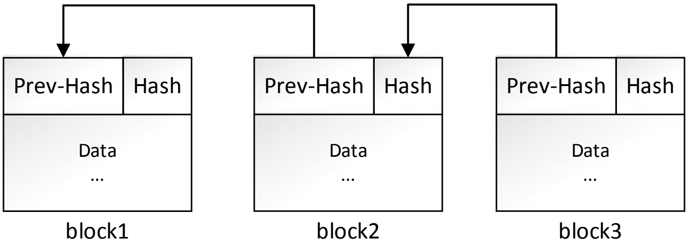


Fig. 3. Simplified block structure

5 Limitation and Challenge

Blockchain has the potential to help establish a secured, trusted and decentralized Swarm robotics ecosystem. However, from a research perspective, several key research issues still need to be addressed for Blockchain-based swarm robotics systems to reach its full potential. For example, the difficulty of achieving consistency quickly; computational complexity affects the application of blockchain on small platforms; the inefficiency of block generation.

1. latency

The latency issue becomes highly relevant when robots are used in formation control or cooperative tasks. The delay will have serious consequences in a variety of mission scenarios, such as formation control, cooperative command.

At present, the block generation time of mainstream blockchain application scenarios is generally too long. Increasing the frequency of block generation, it will reduce security, and may also cause double-flowering problems, lead to an increase in block size. Sergio Demian Lerner put forward the concept of DagCoin, which merges the concepts of transactions and blocks and makes each user a miner [11]. The resulting authenticated data structure is a Direct Acyclic Graph (DAG) of transactions where each transaction "confirms" one or more previous transactions. In DagCoin network, the transaction data will not be been packed in block, which saves time for packaged transactions, effectively reduces the delay of data transmission.

2. block size

The computing and storage performance of different nodes may be different in blockchain-based autonomous robots network. As the blockchain increases, the amount of data stored in the blockchain will become larger, which makes some nodes unable to store the full ledger of transactions anymore. This problem, which the Bitcoin community calls "bloat". The rapidly expanding data capacity of the blockchain will greatly limit the application of blockchains in Swarm robots.

Blockchain-based robots network can reduce the amount of transaction data between nodes by improving the autonomy of nodes, or reduce the data types stored in the blockchain, large amounts of data store in local nodes, been accessed on demand.

3. throughput

Bitcoin and Ethereum adopt block-based data structures. All the blocks been chained in blockchain adopting single-chain structure and cannot handle high concurrent requests. If we reduce the size of block, the speed of block generation will increase, but the security will be reduced.

An idea is separate different types of data in different blockchain with sidechain, subchain. For example, the main chain is responsible for token paying, and the DApp installed on the side chain runs operation data (for example, instruction release), business data is stored locally and is not written into the blockchain. When relevant data is needed, the node can access data by querying the relevant node address stored in blockchain. The hash value of the business data is stored in the subchain for verification. This reduces the amount of interaction data and communication overhead on bandwidth. In addition, the DAG-based blockchain has a meshed data structure, which packages transactions and can processes data concurrently, which greatly improves the throughput of the blockchain, but needs further improvement in terms of security and consistency.

References

1. Tan Y, Zheng Z (2013) Research advance in swarm robotics defence technology **9**:19-9. http://dx.doi.org/10.1016/j.dt.2013.03.001
2. Garattoni L, Birattari M (2016 15 Aug) Swarm robotics. https://doi.org/10.1002/047134608X.W8312
3. Nakamoto S (2008) Bitcoin: a peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf

4. Buterin V (2014) A next-generation smart contract and decentralized application platform. Ethereum project white paper. https://github.com/ethereum/wiki/wiki/White-Paper
5. Castelló Ferrer E (2016) The blockchain: a new framework for robotic swarm systems. arXiv (accessed 9 Mar 2019) http://arxiv.org/abs/1608.00695
6. Castelló Ferrer E, Rudovic O, Hardjono T, Pentland (2018) A robochain: a secure data-sharing framework for human-robot interaction. arXiv (accessed 9 Mar 2019) http://arxiv.org/abs/1802.04480
7. Tran JA, Ramachandran GS, Shah PM et al (2019) SwarmDAG: a partition tolerant distributed ledger protocol for swarm robotics. https://ledgerjournal.org/ojs/index.php/ledger/article/view/174/123
8. Higgins F, Tomlinson A, Martin KM (2009) Survey on security challenges for swarm robotics. In: Proceedings of fifth international conference autonomic and autonomous systems. IEEE Press, 307–312. https://doi.org/10.1109/ICAS.2009.62
9. Brambilla M, Ferrante E, Birattari M, Dorigo M (2013) Swarm robotics: a review from the swarm engineering perspective. Swarm Intell 7(1):1–41. https://doi.org/10.1007/s11721-012-0075-2
10. Back A, Corallo M, Dashjr L et al (2014) Enabling blockchain innovations with pegged sidechains. Accessed 9 Jun 2019. https://www.blockstream.com/sidechains.pdf
11. Lerner SD (2015) DagCoin Draft.9 (accessed 1 May 2019). https://bitslog.files.wordpress.com/2015/09/dagcoin-v41.pdf

A Transmission Method for Intra-flight Ad Hoc Networks Based on Business Classification

Songhua Huang(✉)

Science and Technology on Information Systems Engineering Key Laboratory,
Nanjing Research Institute of Electronics Engineering, Nanjing 210007, China
shhuang@aliyun.com

Abstract. With dynamic topology, insufficient wireless bandwidth and variable path quality, new IP based applications across intra-flight ad hoc networks are always challenged by the fact that transmission services cannot always meet the requirements of the businesses. After analyzing the current status of transmission method of mobile ad hoc networks, this paper proposes a multi-mode transmission method based on business classification for intra-flight ad hoc networks. And the simulation results indicate that, compared with traditional transmission protocol, this method increases the throughput of intra-flight ad hoc networks greatly, while meeting the requirements of various businesses, such as the small capacity and low latency, the large capacity and low jitter, and the high reliability.

Keywords: Intra-flight ad hoc networks · Multi-mode transmission · Self-adaptive · Cross-layer sharing

1 Introduction

Intra-flight ad hoc networks is the application of mobile network technology in the field of aviation communication, which mainly by routing and forwarding within multiple hops helps to support automatic connections and communications between aircraft platforms, and then to realize distributions and exchanges of instructions, intelligence, environment perception information, and flight status that military and civilian aviation platforms need urgently [1]. Since the network layer of TCP/IP stack provides a connectionless datagram service, which means the IP datagram transmission will be lost, repetitive or out-of-order, so the transport layer of TCP/IP stack becomes extremely important. From the perspective of reliable transmission, situation information, instruction, and target need to be obtained more timely and accurately under military environment, to improve the target strike chain with transmission efficiency. Also at the civil level, the real-time communication, streaming service and data sharing experience of air users during flight should be ensured. At present, the transmission bandwidth and reliability provided in the wireless environment are far from satisfying many concurrent requirements of new applications and services based on IP. And furthermore, time-varying of delay and bit error rate may lead to poor application experience [2]. Based

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 361–368, 2020.
https://doi.org/10.1007/978-981-15-0187-6_42

on the analysis of the existing transmission modes and challenges of mobile ad hoc networks, a multi-mode transmission method with self-adaptation is proposed for intra-flight ad hoc networks to guarantee business transmission reliability under high dynamic environment, and to improve the overall transmission capacity meanwhile.

2 Related Works

Since topology changes dynamically, link quality is not stable, and routing paths are not always there in mobile ad hoc networks, the standard TCP protocol is improved by many scholars, to avoid the dropping of network throughput caused by a slow start and fast retransmission, which is triggered by TCP data packet loss or out-of-order. The TCP improvement schemes are as follows.

Kim KB and Chandran K et al. introduced TCP-feedback mechanisms, improving the performance of transmission control protocol, and dealing with the problem of routing faults based on network state feedback in ad hoc networks [3, 4]. Compared with the original TCP, these methods improved the throughput of ad hoc networks, since the TCP sender can continue to carry out packet transmission in large windows after the routing reconstruction. Prajapati HB proposed feedback mechanisms based on cross-layer ELFN (Explicit Link Failure Notification), to decide if TCP enters the standby mode or the normal operating condition, by sensing the information interaction between the intermediate nodes and the sender [5]. Based on ELFN notices, Sengottaiyan N proposed EPLN-BEAD (Early Packet Loss Notification and Best-Effort Acknowledge Delivery) mechanism, in order to distinguish whether data packets or confirmation packets are lost, and then to reduce the TCP timeout events and improve throughput of mobile ad hoc networks [6].

The proposed improvements to TCP for mobile ad hoc networks include ATCP (Ad hoc TCP) and TCP-BuS (TCP Buffering capability and Sequence information) [7, 8]. The former inserts the middle layer between the transport layer and the network layer on the sending end. And the middle tier monitors information from the network layer, and gives feedback to TCP. Meanwhile, through monitoring the number of received acknowledge packets to determine the merits of the channel, in order to solve the poor performance of the transport layer owing to the routing problem or high bit error rate conditions; The latter also uses the feedback information of the network layer to detect the routing failure and then adopts the corresponding strategy, and in the route reconstruction process, the related packets from the source node to the intermediate node will be cached.

In addition, in response to the transmission spectrum switching of cognitive wireless network, some researchers proposed TCP improved protocol TCP-CR (Cognitive Radio) based on cross-layer mechanism between TCP and MAC, to solve the problem of TCP timeout retransmission and frequent slow start caused by frequency spectrum switching, and to improve the channel utilization, transmission efficiency and the end-to-end throughput [9, 10].

Schemes mentioned above mainly use the feedback information from network layer and link layer, to distinguish reasons for packet loss, such as network congestion, routing failure or link failures, to improve the transmission efficiency by improving the process of slow start. However, these solutions need building the end-to-end connection, and the delay produced cannot be endured for the low latency collaborative information. Also for high dynamic air environment, it is difficult to guarantee the end-to-end connection.

This paper integrates cross-layer perception (requirements of various businesses and reasons of packet loss), cross-layer and lateral transmission, segmented handshake for confirmation, error correction, put forwards a variety of transmission mode, like segment-reliable mode, half reliable mode and efficient mode, and also the method of adaptive handoff, to satisfy the various requirements of intra-flight ad hoc networks, such as low latency, high reliability, large capacities and so on.

3 Multi Transmission Modes

According to the link characteristics of intra-flight ad hoc networks and to meet the transmission requirements of instructions, intelligence, and control information, the segment-reliable mode, half reliable mode, and efficient mode are proposed by adopting different handshake and confirmation mechanisms based on segmented connections at the transport layer.

3.1 Frame Structure of the Transmission Protocol

In order to communicate with TCP/IP based nodes, transport protocol for intra-flight ad hoc networks should be able to interoperate effectively with TCP and UDP. At the same time, in the airborne environment, the link bandwidth is limited, thus the efficient data frame structure is very important. The data frame structure of the intra-flight ad hoc network transmission protocol is shown in Table 1.

Table 1. Frame structure of the transmission protocol

Source port			Destination port
Sequence number			
Timestamp			
Mode	Resv.	ECN + flags	HEC
Payload			
CRC-32			

In this frame, Source port is the port number of the application that sends the data; Destination port is the port number of the application that receives the data; Sequence number is the serial number of the data frame, which can solve the chaotic sequence of data frames to the destination node. Timestamp is when the data are sent; Mode indicates which transport service mode is used in the current data transmission; The HEC is the check code of the header to ensure the integrity of the message header in the wireless channel; Payload is data to be transferred; Crc-32 is the check code for the data to be transferred to ensure the correctness of the data to be transmitted, which perform different functions in different transmission modes.

3.2 Segment-Reliable Transmission Mode

Segment-reliable transmission mode introduces a broken-point continuingly-transferring mechanism, and where data package can be incepted, acknowledged and cached in an en-route node and then opportunistic routing [11, 12] will be triggered when the next node temporarily gives no response. In extreme cases, end-to-end message validation between source node and destination node will be replaced by hop confirmation and on-demand routing, to ensure the transport services, namely an en-route node, when receiving data packages, sends a confirmation message immediately, and then transmission reliability will be ensured by the nodes coming after in order.

In this mode, TCP was improved for each segment, as follows: (1) the connection request ASYN and data are sent at the same time, and the three-way handshake between the source and destination is unnecessary. (2) Piecewise, connection is dismantled when data is sent and AFIN signaling comes after, so as to adapt to topology dynamics of intra-flight ad hoc networks and fluctuation of link quality. (3) The sender senses clearly the true state of the networks, through the cross-layer bus where network layer, link layer, and physical layer are properly involved in the process of finding and solving problems like network congestion, routing failure (temporary failure caused mobility), link failure and so on. For example, in the physical layer the node movement trend will be predicted by measuring the strength of the wireless signal to calculate the distance and its variation as well according to the function relation between signal strength and node distance; when the distance exceeds a certain threshold, and the transport layer sending failure is considered to be packet loss caused by routing interruption; otherwise, when the node is within a certain range, the failure is considered to be a random error. In the network layer, if the acknowledge cannot be received, it can be considered that the path is interrupted; and if failure is caused by package error, it can be considered that the link quality drops and the error code occurs. (4) When the routing interrupt is found, the sending end stops sending the data and freezes the current environment variables; when the sender is informed of the routing recovery, the data transfer is restored using the value of the variables frozen before. (5) When the failure of the transmission frequently appears in the link-layer due to the error code, the sending end decelerates the data sent to alleviate the pressure of the channel. (6) To improve the efficiency of network transmission, data cache and breakpoint propagation are adopted in intermediate nodes, adapting to the dynamic characteristics of intra-flight ad hoc networks.

3.3 Half-Reliable Transmission Mode

Similar to the segment-reliable transmission mode, half-reliable transmission mode will establish some sections between source node and destination node, where the connection request and error correction mechanisms for every segment are adopted. Thus, an error correction mechanism is mainly used to ensure transport services.

In this mode, data and its error-correcting code are sent with the connection request ASYN at the same time. The ASYN only plays the role of informing the next node that should be ready for data receiving, inspecting and error-correcting, forwarding, but need not confirming. After the data is sent, a one-way connection is removed when time is out, without sending an AFIN signaling.

3.4 Efficient Transmission Mode

Efficient transmission mode establishes a direct link between the source node and destination node. At the same time, link service is called directly across the underlying network layer after packets of transport layer are encapsulated by the link-layer packets to ensure near real-time transport services for small-capacity data.

In order to improve the adaptability of the network, and to provide awareness of requirements from the application layer, different periods of life are set up, and three strategies such as “low delay, high reliability, and large throughput” are provided.

(1) Order and text information: because of the small amount, but high-reliability requirements, adopt “high reliability” strategy, using segment-reliable transmission mode, where, the path is divided into multiple segments which are relatively stable. And in each segment, the improved transmission mode is used, to ensure the reliability of information transmission under the high dynamic environment.
(2) Image and the streaming media information: because of the large amount and strict time delay requirement, adopt “big throughput” strategy, using half-reliable transmission mode, to improve the throughput of overall transmission for inter-flight ad hoc networks, and to guarantee the reliability of the large-capacity information transmission.
(3) Control information: because the little amount and nearly real-time requirements, adopt “low latency” strategy, using efficient transmission mode, namely the link-layer header encapsulation is used for transport layer packet directly, and then the packet is sent through direct link with higher power, to reduce the time costs of forwarding, ensuring the timeliness of control information transmission.

4 Comparison Analysis

Based on the OPNET 14.5 simulation platform, the M^2T simulation software for inter-flight ad hoc networks is developed. Compared with the traditional TCP, the validity of M^2T is verified by simulation.

4.1 Validity of M^2T

The packet loss rate and bandwidth occupancy of M^2T are tested with different link quality. Simulation scenario is set as follows: link error rate is in a range from 0.1 to 5.0%; the maximum packet delay allowed is 300 ms (a packet, whose delay is over 300 ms, is invalid and will be discarded directly); One-way network delay is 50–75 ms, it means that a packet can be retransmitted once or twice.

The test case is a real-time bi-directional audio stream between node 1 and node 2. And then node 3, which is configured as the relay node, is inserted between node 1 and node 2. The audio stream, employing the G.711a code, is sent at 4 ms intervals with the length of 84 bytes for each packet. Table 2 shows the performance of one-way voice data transmission with different link error rates and delay conditions.

Table 2. Performance of one-way voice data transmission

Link error rates (%)	Number of effective packets	Number of lost flames in link layer	Number of lost packets (50 ms link delay)	Packet loss ratio (50 ms link delay) (%)	Number of lost packets (75 ms link delay)	Packet loss ratio (75 ms link delay) (%)
0.1	25,198	70	0	0	0	0
0.2	25,198	101	0	0	0	0
0.5	25,198	256	0	0	0	0.01587
1.0	25,198	556	0	0	2	0.03969
2.0	25,198	1077	0	0	21	0.1032
5.0	25,198	2912	2	0.0119	68	0.5318

As can be seen from Table 2, M^2T method can effectively reduce the packet loss rate caused by link quality fluctuation and improve network throughput during the dynamic segment transmission of inter-flight ad hoc networks.

4.2 Comparisons Between M^2T and TCP

In the same simulation environment, the throughput of M^2T and standard TCP in different packet loss rates are tested. Under the simulation scenario, the range of link error rate is from 0.1 to 5.0%, and the communication channel of 4 Mbps is allocated by TDMA, i.e., 500 Kbps. The test cases are files, which are grouped at 50 MB, transferring between two nodes, and the test results are shown in Table 3.

Table 3. Comparisons between M^2T and TCP

Link error rate (%)	Throughput of TCP (KBps)	Throughput of M^2T (KBps)	Comparisons (%)
0	460.10	460.10	100
0.1	128.204	396.823	309.525
0.2	96.177	333.322	346.571
0.5	58.383	312.499	535.257
1.0	34.381	238.095	692.52
2.0	18.933	138.888	733.58
5.0	6.257	23.377	373.61

It can be seen from Table 3 that the throughput of M^2T is much higher than that of traditional TCP transmission mode when the link quality is poor or fluctuating.

5 Conclusions

There are various business requirements of inter-flight ad hoc networks, and the service quality is confronted with challenges such as the time-varying topology, inaccurate network status and the fluctuation of link quality. This paper analyzes the research status of transmission methods for inter-flight ad hoc networks, and then integrates cross-layer perception (transmission requirements and reasons of packet loss), cross-layer and lateral transmission, segmented handshake confirmation, error correction, proposes a multi-mode transmission and its adaptation method based on the classification of the business requirements of inter-flight ad hoc networks. Compared with the traditional TCP improvement scheme, this method, while meeting the various transmission requirements like low latency with small capacity, low jitter with large capacity, high reliability, improves the overall transmission capacity, and enhances the reliability and stability of inter-flight ad hoc networks.

References

1. Zheng B, Zhang HY, Huang GC (2011) Status and development of aeronautical ad hoc networks. Telecommun Sci 30(5):38–47
2. Grace K (2008) Airborne network quality-of-service management. The MITRE Corporation
3. Kim KB (2006) Design of feedback controls supporting TCP based on the state-space approach. IEEE Trans Autom Control 51(7):1086–1099
4. Chandran K, Raghunathan S, Venkatesan S et al (2001) A feedback-based scheme for improving TCP performance in ad hoc wireless networks. IEEE Pers Commun 8(1):34–39
5. Prajapati HB, Bhatt BS, Dabhi VK (2008) An FLFN-based adaptive strategy to improve TCP performance in ad hoc netwroks. In: IEEE proceedings of the international conference on computer and communication engineering, Kuala Lumpur, Malaysia, 570–573
6. Sengottaiyan N, Somasundaram R, Arumugam S (2010) An modified approach for measuring TCP performance in wireless ad hoc network. In: IEEE proceedings of the international conference on advances in recent technologies in communication and computing, Kottayam, Kerala, India, 267–270
7. Huang A, Bao J (2013) Transmission performance analysis for VANET based on 802.11p. In: IEEE proceedings of the international conference on communicational and information sciences, Shiyan, Hubei, China, 1178–1181
8. Kim D, Toh CK, Choi Y (2001) TCP-BuS: improving TCP performance in wireless ad hoc networks. J Commun Netw 3(2):1–12
9. Sun W, Chen M, Jiang L et al (2015) The improvement and performance analysis of TCP protocol based on cognitive radio network. J Hangzhou Dianzi Univ (Natural Sciences) 35 (5), 18–22

10. Xue Y, Zhou L (2016) Research on TCP enhancement in cognitive network based on cross-layer designing. J CAEIT 11(2):208–213
11. Jadhav P, Satao R (2016) A Survey on opportunistic routing protocols for wireless sensor networks. Procedia Comput Sci 79:603–609
12. Luo J, Hu J, Wu D et al (2015) Opportunistic routing algorithm for relay node selection in wireless sensor networks. IEEE Trans Indu Inf 11(1):112–121

3D Statistical Resolution Limit of Two Close Spaced Targets for Active Array in the GLRT Context

Pei Peng(✉), Yunlei Zhang, and Jianbin Lu

Electronic Engineering Institute, Navy University of Engineering, Wuhan, China
46190145@qq.com

Abstract. The current research of the statistical resolution limit (SRL), which is based on the hypothesis test, usually takes Taylor expansion and approximation to get a linear model of general likelihood rate test (GLRT) and achieves an analytical expression of the SRL. In the way, one dimension (range, angle) and two dimensions (range-Doppler, angle-Doppler) SRLs have been explored. In this paper we dwell on the three-dimension (3D) SRL in range-angle-Doppler domain and discuss the factors which make effects on 3D SRL. Our theoretical and simulation results both demonstrate that the 3D SRL is the weighting square sum of three respective SRLs, which will throw some insights into the systemic design to improve the resolution ability.

Keywords: Hypothesis test · Active array · 3D SRL · GLRT · Taylor expansion and approximation

1 Introduction

Statistical resolution limit (SRL) indicates the minimum separation from the statistical standpoint, which describes the ability of the system distinguishing two very close parameters under some given resolution rate and false-alarm rate, and the ability is related to the signal-noise-rate (SNR) and signal waveforms, etc. Nowadays, there are a lot of researches about this, and on the one-dimension aspect, the authors in [1] studied the range SRL in the optics domain, while authors in [2] and [3] made researches on the passive array, which adopted the approximation general likelihood rate test (GLRT) and the information entropy criteria, respectively. The DOA and DOD united SRL was explored in [4]. On the multi-dimension aspect, the range-Doppler SRL was studied in [5] for the active array, while the angle-Doppler SRL with space-time snaps was explored in [6]. The authors in [7] had expended the united angle SRL to the Doppler dimension. However, there is no research on the range-angle-Doppler domain yet.

In the reality for sensors like radar observing two or more very closely targets, their parameters along with the range, angle and Doppler may be very close, so the research of 3D SRL is necessary. From the results of the 2D SRL we can see that, the unit SRL is a function of single SRLs, while for the transmitted the linear frequency modulation (LFM) signal, the range-Doppler SRL in [5] is the weighting sum of range SRL and Doppler SRL. In [6] the angle-Doppler SRL is also the function of the angle SRL and

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 369–377, 2020.
https://doi.org/10.1007/978-981-15-0187-6_43

Doppler SRL. Therefore, the range-angle-Doppler SRL should also be a function of the three respective SRLs and the expression of it is our concerned problem. Besides, the range-Doppler SRL in [5] is for the inner pulse, which is very poor as the pulse length is usually very short, so we will investigate Doppler SRL among multiple pulses.

Similar to the existing researches, we will first establish the binary hypothesis test model, then get a linear model of the GLRT with the Taylor expansion and approximation, further we will explore the factors which will make effects on the united SRL and make some simulations.

1.1 3D SRL Model

In this section, we will extend the results in [5] to the angle domain for the active array, by bringing in array steering vector, and also extent inner pulse-Doppler to inter-pulse Doppler.

This work was supported by the National Natural Science Foundation of China under Grant 61501486.

Assume the complex signal vector is $\mathbf{s}$, while its maximum bandwidth is $B_{\max}$ and pulse length is τ. Two targets' echoes amplitudes are α_1 and α_2 ,the arrival time are t_1 and t_2, and their interval is $\delta_t = t_1 - t_2$, which is smaller than the resolution unit, i.e., $\delta_t < 1/B_{\max}$.

Being similar to the existing research, we assume the center of the two echoes, $n_0 T_{\mathrm{s}} = (n_1 + n_2)T_{\mathrm{s}}/2$ is known, where $t_1 = n_1 T_{\mathrm{s}}, t_2 = n_2 T_{\mathrm{s}}$, and T_{s} is the sampling interval period. As the inner pulse-Doppler is so tiny that we omit it. After Taylor expansion and keeping the first order item as approximation, we get the single echo pulse in vector form as:

$$\begin{aligned} \mathbf{s}_1 &\triangleq s\left(t_0 - \frac{\delta_t}{2}\right)_{t=nT_{\mathrm{s}}} \approx \mathbf{s}_0 - \frac{\delta_t}{2}\dot{\mathbf{s}}_0 \\ \mathbf{s}_2 &\triangleq s\left(t_0 + \frac{\delta_t}{2}\right)_{t=nT_{\mathrm{s}}} \approx \mathbf{s}_0 + \frac{\delta_t}{2}\dot{\mathbf{s}}_0 \end{aligned} \tag{1}$$

where $\mathbf{s}_0 \triangleq \mathbf{s}(n_0)$, $\dot{\mathbf{s}}_0 \triangleq \dot{\mathbf{s}}(n_0) = \mathrm{d}s(t)/\mathrm{d}t|_{t=n_0 T_{\mathrm{s}}}$.

Assume a uniform linear array with half-wavelength spacing M sensors, and make the center position as reference, then the position vector can be expressed as:

$$\mathbf{d} = [\, d_{-(M-1)/2} \quad \cdots d_m \quad \cdots \quad d_{(M-1)/2} \,]^{\mathrm{T}} \tag{2}$$

where the superscript T is the transpose operation. Then the steering vector is:

$$\begin{aligned} &\left[\mathbf{v}_{sp}\right]_m \triangleq \left[\mathbf{v}_s(w_{sp})\right]_m = \exp(jd_m w_{sp}), \\ &m = -(M-1)/2 : (M-1)/2, p = 1,2 \end{aligned} \tag{3}$$

where $w_{sp} = \pi \sin\theta_p$ is the angular frequency of the pth target, and θ_p is its DOA. For simplicity and without loss of the generality, here we assume that the gravity of the position vector is located at its center, i.e., $\sum_m d_m = 0$.

Similarly, assume the center angle of the two targets, $w_{s0} = (w_{s1} + w_{s2})/2$ is known and $\delta_a = w_{s2} - w_{s1}$ is the separation in the angle dimension. After a similar transformation with (1), we get the approximation steering vector as:

$$\begin{aligned} \mathbf{v}_{s1} &\triangleq \mathbf{v}_s\left(w_{s0} - \frac{\delta_a}{2}\right) \approx \mathbf{v}_{s0} - \frac{\delta_a}{2}\dot{\mathbf{v}}_{s0} \\ \mathbf{v}_{s2} &\triangleq \mathbf{v}_s\left(w_{s0} + \frac{\delta_a}{2}\right) \approx \mathbf{v}_{s0} + \frac{\delta_a}{2}\dot{\mathbf{v}}_{s0}, \end{aligned} \tag{4}$$

where $\mathbf{v}_{s0} \triangleq \mathbf{v}(w_{s0})$, $\dot{\mathbf{v}}_0 \triangleq \dot{\mathbf{v}}(w_{s0}) = \mathrm{d}\mathbf{v}(w_s)/\mathrm{d}w_s|_{w_s = w_{s0}}$.

Assume there are L pulses and their repeat frequency is f_r, and make the center as a reference, then the steering vector in Doppler domain is:

$$\mathbf{v}_{\mathrm{dp}} = \mathbf{v}_d(w_{\mathrm{dp}}) = \left[\, \mathrm{e}^{-j\frac{L-1}{2f_\mathrm{r}}w_{\mathrm{dp}}} \quad \cdots \quad \mathrm{e}^{j\frac{L-1}{2f_\mathrm{r}}w_{\mathrm{dp}}} \,\right]^\mathrm{T}, \quad p = 1, 2 \tag{5}$$

where w_{dp} is the Doppler frequency of the pth target. Without loss of generality, the pulse ID also meets that $\sum_{-(L-1)/2}^{(L-1)/2} l = 0$.

Again, assume the center $w_{d0} = (w_{d1} + w_{d2})/2$ is known, and the interval is $\delta_d = w_{d2} - w_{d1}$, then the approximation vectors are:

$$\begin{aligned} \mathbf{v}_{d1} &\triangleq \mathbf{v}_d\left(w_{d0} - \frac{\delta_d}{2}\right) \approx \mathbf{v}_{d0} - \frac{\delta_d}{2}\dot{\mathbf{v}}_{d0} \\ \mathbf{v}_{d2} &\triangleq \mathbf{v}_d\left(w_{d0} + \frac{\delta_d}{2}\right) \approx \mathbf{v}_{d0} + \frac{\delta_d}{2}\dot{\mathbf{v}}_{d0}, \end{aligned} \tag{6}$$

where $\mathbf{v}_{d0} \triangleq \mathbf{v}(w_{d0})$, $\dot{\mathbf{v}}_0 \triangleq \dot{\mathbf{v}}(w_{d0}) = d\mathbf{v}(w_d)/dw_d|_{w_d = w_{d0}}$.

Furthermore, $\mathbf{w}$ is assumed to be complex white Gauss noise (CWGN) with zero mean value and covariance $\sigma^2\mathbf{I}$. With (1–5) and some approximation, the L pulses received by the active array can be expressed as:

$$\begin{aligned} \mathbf{x} &= \alpha_1(\mathbf{v}_{s1} \otimes \mathbf{v}_{d1} \otimes \mathbf{s}_1) + \alpha_2(\mathbf{v}_{s2} \otimes \mathbf{v}_{d2} \otimes \mathbf{s}_2) + \mathbf{w} \\ &\approx \alpha_+\mathbf{p} + \alpha_-\delta_t\dot{\mathbf{p}}_t/2 + \alpha_-\delta_a\dot{\mathbf{p}}_s/2 + \alpha_-\delta_d\dot{\mathbf{p}}_d/2 + \mathbf{w} \\ &= \underbrace{[\mathbf{p}\,\dot{\mathbf{p}}_t\,\dot{\mathbf{p}}_s\,\dot{\mathbf{p}}_d]}_{\mathbf{H}}\underbrace{[\alpha_+ \; \alpha_-\delta_t/2\,\alpha_-\delta_s/2\,\alpha_-\delta_d/2]^\mathrm{T}}_{\boldsymbol{\theta}} + \mathbf{w} \end{aligned} \tag{7}$$

where $\alpha_+ = \alpha_1 + \alpha_2$ and $\alpha_- = \alpha_2 - \alpha_1$, $\mathbf{p} \triangleq \mathbf{v}_{s0} \otimes \mathbf{v}_{d0} \otimes \mathbf{s}_0$, $\dot{\mathbf{p}}_t \triangleq \mathbf{v}_{s0} \otimes \mathbf{v}_{d0} \otimes \dot{\mathbf{s}}_0$, $\dot{\mathbf{p}}_s \triangleq \dot{\mathbf{v}}_{s0} \otimes \mathbf{v}_{d0} \otimes \mathbf{s}_0$ and $\dot{\mathbf{p}}_d \triangleq \mathbf{v}_{s0} \otimes \dot{\mathbf{v}}_{d0} \otimes \mathbf{s}_0$.

Then our task is to decide whether there is one or two targets present in the echo, which can be modeled as a binary hypothesis test problem, where H_0 represents only one target, while H_1 represents two targets, as:

$$\begin{cases} \mathrm{H}_0\colon \delta_t = \delta_a = \delta_d = 0 \\ \mathrm{H}_1\colon \delta_t \neq 0 \text{ or } \delta_a \neq 0 \text{ or } \delta_d \neq 0 \end{cases}. \tag{8}$$

1.2 3D Resolution Statistical and Its Performance

For the binary hypothesis test model in (7), we bring in a selecting matrix to remodel it as

$$\begin{cases} \mathrm{H}_0\colon \mathbf{A}\boldsymbol{\theta} = \mathbf{0}, \alpha_1, \alpha_2 \\ \mathrm{H}_1\colon \mathbf{A}\boldsymbol{\theta} \neq \mathbf{0}, \alpha_1, \alpha_2 \end{cases} \tag{9}$$

where the selecting matrix $\mathbf{A}$ is

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{10}$$

and $\boldsymbol{\theta} = [\alpha_+ \; \alpha_-\delta_t/2 \, \alpha_-\delta_s/2 \, \alpha_-\delta_d/2]^{\mathrm{T}}$ is the unknown parameter vector, which can be got from (6), including the nuisance parameter $\alpha_+ = \alpha_1 + \alpha_2$. Therefore, this is complex test model and we should apply GLRT. We first get the most likelihood estimation of the unknown parameters, as $\hat{\boldsymbol{\theta}}_1 = (\mathbf{H}^H\mathbf{H})^{-1}\Re(\mathbf{H}^H\mathbf{z})$, where $\Re(\bullet)$ denotes the real part of a complex number. Then we can get the hypothesis test statistic as:

$$2\ln L_G(\mathbf{z}) = \frac{(\mathbf{A}\hat{\boldsymbol{\theta}}_1)^H[\mathbf{A}(\mathbf{H}^H\mathbf{H})^{-1}\mathbf{A}^{\mathrm{T}}]^{-1}(\mathbf{A}\hat{\boldsymbol{\theta}}_1)}{\sigma^2/2} \gtrless \gamma \tag{11}$$

where γ is the resolution threshold. The distribution of the statistic is [8]:

$$2\ln L_G(\mathbf{z}) \overset{a}{\sim} \begin{cases} \chi_3^2, & \mathrm{H}_0 \\ \chi_3'^2(\lambda), & \mathrm{H}_1 \end{cases} \tag{12}$$

where the non-centrality parameter is:

$$\lambda = \frac{(\mathbf{A}\boldsymbol{\theta}_1)^H\left(\mathbf{A}\left(\mathbf{H}^H\mathbf{H}\right)^{-1}\mathbf{A}^{\mathrm{T}}\right)^{-1}(\mathbf{A}\boldsymbol{\theta}_1)}{\sigma^2/2} \tag{13}$$

If K times sampling frequency of the bandwidth B of LFM signal is transmitted, i.e., $f_{\mathrm{s}} = KB$, this parameter can be further calculated as:

$$\lambda = \frac{2\pi^2|\alpha_-|^2}{\sigma^2}\left(\begin{array}{l} \frac{MLK_1(N)}{N^2}\left(\frac{\delta_t}{\delta_t^R}\right)^2 \\ + \frac{NLK_2(M)}{M^2}\left(\frac{\delta_a}{\delta_a^R}\right)^2 + \frac{MNK_3(L)}{L^2}\left(\frac{\delta_d}{\delta_d^R}\right)^2 \end{array}\right) \tag{14}$$

where $K_1(N) = \sum_{n=-(N-1)/2}^{(N-1)/2} n^2$, $K_2(M) = \sum_{m=-(M-1)/2}^{(M-1)/2} m^2$, $K_3(L) = \sum_{l=-(L-1)/2}^{(L-1)/2} l^2$, which are square sum of the sampling numbers in range, angle and Doppler dimension.$\delta_t^R = 1/B = K\tau/N$, $\delta_a^R = 2\pi/M$, $\delta_d^R = 2\pi/L$ are, respectively, the Rayleigh limits in three dimensions. The derivation of (13) is similar with the appendix in [5], which is omitted here for simplify.

One can get that this is different with the conclusion in [5], where it is a linear weighting sum of different SRLs normalized by Rayleigh limit, while here it is a weighting square sum of them.

2 The Influence Factors on 3D SRL

According to the knowledge of the Chi-Square distribution, with a given non-centrality parameter, the bigger non-centrality parameter is, the better the resolution rate is. In this section, we will utilize this conclusion to analyze the influence factors of the 3D SRL.

With (13), we can define the 3D SRL as:

$$\delta = \sqrt{\left(\frac{\delta_t^0}{a}\right)^2 + \left(\frac{\delta_a^0}{b}\right)^2 + \left(\frac{\delta_d^0}{c}\right)^2} \tag{15}$$

where $\delta_t^0 = \delta_t/\delta_t^R$, $\delta_a^0 = \delta_a/\delta_a^R$ and $\delta_d^0 = \delta_d/\delta_d^R$ are the normalized limits along the range, angle, and Doppler, respectively, which are divided by Rayleigh limit.$a = \sqrt{N^2/K_1(N)/M/L}$, $b = \sqrt{M^2/K_2(M)/N/L}$ and $c = \sqrt{L^2/K_3(L)/N/M}$ are the corresponding radius of the ellipsoid defined by (14). With the above conclusion, if the value of (14) becomes bigger, then the resolution rate will improve, or if we fix the resolution rate, a small value of (14) can be needed.

On the other hand, we define the different SNR as $\mathrm{SNR_D} = |\alpha_-|^2/\sigma^2$, so the 3D SRL can also be expressed as:

$$\delta = \sqrt{\frac{\lambda(p_\mathrm{d}, p_\mathrm{f})}{2\pi^2 \mathrm{SNR_D}}} \tag{16}$$

Based on the above results, some conclusions can be drawn:

(1) 3D SRL is related to the sampling point numbers, including *M*, *N,* and *K*. They affect not only the normalized limit, but also the radius of the ellipsoid. As their relations are complex, we will dwell on it by the simulation in the next section.

(2) 3D SRL can be traded off, i.e., if we want to relax some limit, we have to improve its sampling point number. However, the result will become poorer as it goes higher.

(3) Similar to the conclusion of 2D SRL in [5], 3D SRL is also reverse ratio to different SNR, which means that huge difference power of two signals will resort to better performance.

3 Simulation

All simulations are based on the half-wavelength separation uniform linear array. Considering the carry frequency will not affect the results, we take zero-carried frequency LFM signal as $s(t) = e^{j\pi K_r t^2}$, whose frequency modulation rate is $K_r = B/\tau$, and the time is $-\tau/2 \leq t \leq \tau/2$. The false-alarm rate is set as $p_{\mathrm{f}} = 0.01$.

Example 1 To verify the correctness of the theoretical results, and plot the resolution rate versus the separation and SNR. The number of the array sensors is M = 5, signal bandwidth $B = 1\,\mathrm{MHz}$ and time length $\tau = 10\,\mu\mathrm{s}$. The sampling frequency is $f_{\mathrm{s}} = 10\,\mathrm{B} = 10\,\mathrm{MHz}$, the angle center is $w_{\mathrm{s}0} = 0\,\mathrm{rad}$ the Doppler center is $w_{\mathrm{d}0} = 2\pi \times 10^6\,\mathrm{rad}$. The separation in Fig is normalized. The SNR is defined as $\mathrm{SNR}_i = |\alpha_i|^2/\sigma^2$. The results are given in Fig. 1, where the lines are the theoretical results and the makes are the Metro Carlo results.

We can see the two results coincide well with each other. With the SNR or the separation increasing, the resolution rate also increases.

Example 2 Keep the absolute separations along 3D constant and simulate the resolution w.r.t. the number of sampling points. Set $\delta_t = 10^{-6}\,\mathrm{s}$, $\delta_a = 2\pi * 0.1\,\mathrm{rad}$, $\delta_d = 2\pi * 0.1\,\mathrm{rad}$ and $\mathrm{SNR}_D = 0\,\mathrm{dB}$. The results are shown in Fig. 2.

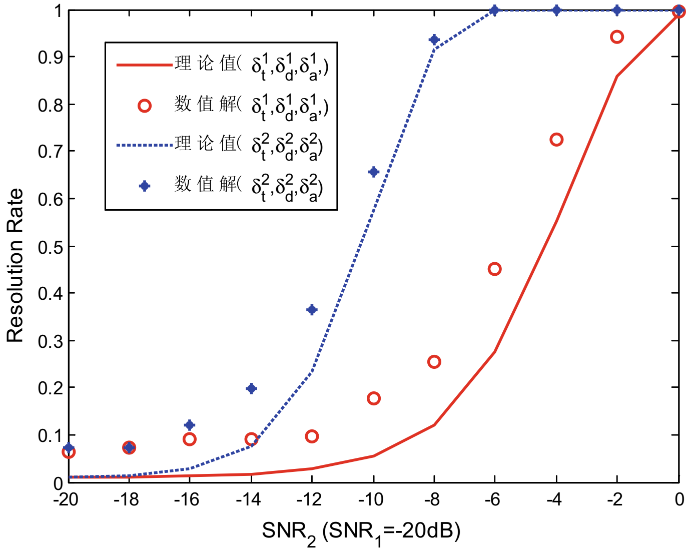


Fig. 1. Comparison of the theoretical and Metro Carlo results

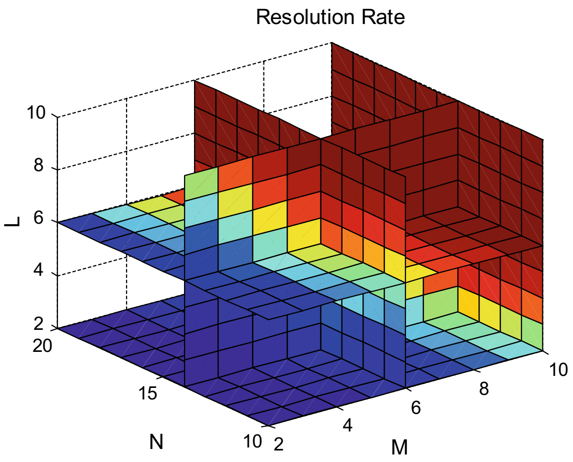


Fig. 2. Resolution rate versus number of the sampling points (absolute separation constant)

One can get that with either one of sampling points increasing, the resolution rate will increase.

Example 3 Keep the normalized separation constant, and simulate the resolution rate w.r.t. the number of sampling points. Set $\delta_t^0 = \delta_a^0 = \delta_d^0 = 0.1$ and $\mathrm{SNR}_D = 0\,\mathrm{dB}$, the results are shown in Fig. 3.

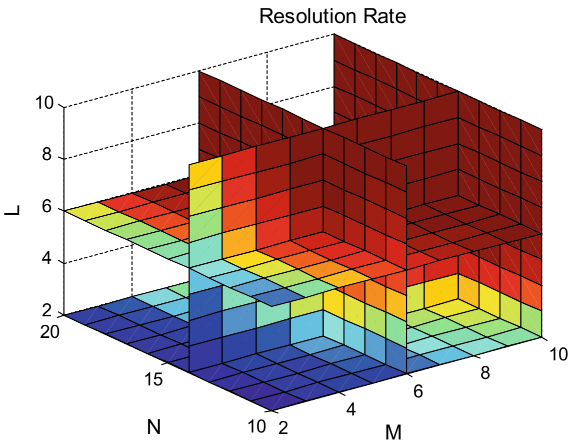


Fig. 3. Resolution rate versus number of the sampling points (relative separation constant)

One can see the results are similar to Fig. 2, where the resolution rate increases as either sampling point number goes high.

Example 4 Keep the 3D SRL and other two normalized SRLs constants, and simulate the trade-off of Doppler sampling number and its normalized resolution separation. Set $p_{\mathrm{d}} = 0.9$ and $p_{\mathrm{f}} = 0.01$, so $\lambda = 19.2$ is got. Set $\delta_t^0 = \delta_a^0 = 0.1$, and the results are shown in Fig. 4.

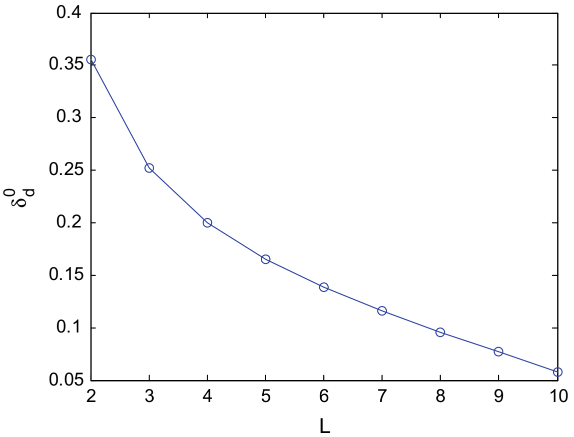


Fig. 4. Doppler SRL versus its sampling number $\left(\delta_t^0 = \delta_a^0 = 0.1\right)$

One can get that with the increase of sampling point number *L*, the Doppler SRL increases, while the effect will become poor as the number goes higher.

4 Conclusions

In this paper, we establish a binary hypothesis model and apply the Taylor expansion and approximation to get an analysis expression of 3D SRL, based on which we make a discussion of the influent factors to 3D SRL. For the LFM signal, increasing the sampling number will improve the resolution performance, with either absolute or relative separation constant. This will greatly help us to design a system for improving its resolvability.

References

1. Shahram M, Milanfae P (2004) Imaging below the diffraction limit: a statistical analysis. IEEE Trans Image Process 13(5):677–689
2. El Korso MN, Boyer R, Renaux A et al (2012) On the asymptotic resolvability of two point sources in known subspace interference using a GLRT-based framework. Sig Process 92 (10):2471–2483

3. Zhu W, Tang J, Wan S (2014) Angular resolution limit of two closely-spaced point sources based on information theoretic criteria. In ICASSP, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp 86–90
4. El Korso MN, Boyer R, Renaux A et al (2012) Statistical resolution limit for source localization with clutter interference in a MIMO radar context. IEEE Trans Signal Process 60 (2):987–992
5. Zhang YL, Wang L, Tang J et al (2018) The Range-Doppler SRL of two close targets for active sensor in the GLRT context. In: International conference on radar (RADAR), 1–6
6. Lu DH, Li Y, Liang C (2013) Resolving targets using space-time snapshot based on GLRT algorithm. In: IET international radar conference, 0348–0348
7. Zhang N, Tang J, Wan S, Zhu W (2014) Three-dimensional extensions of resolvability of two point targets in MIMO radar using a hypothesis test formulation. Int J Antennas Propag 2014:1–12
8. Kay SM (1998) Fundamentals of statistical signal processing: detection theory, vol 2. Prentice Hall, NJ

A Scene Semantic Recognition Method of Remote Sensing Images Based on CSIFT and PLSA

Yan-li Sun, Wei Zhou(✉), Ya-zhou Zhang, and Jialong Jin

Naval Aviation University, Yantai 264001, China
yeaweam@gmail.com

Abstract. Aiming at the fast recognition of local image scenes with clear semantics in high-resolution remote sensing images, such as ports, airports and oil depots, a visual feature representation method for remote sensing images based on CSIFT features and a scene semantic recognition method based on PLSA is proposed. Experiments on typical remote sensing image scenes fully verify the effectiveness of proposed method.

Keywords: Semantic recognition of scenes · Visual feature · PLSA

1 Introduction

Information interpretation of high-resolution remote sensing images usually starts with the detection and recognition of specific types of objects [1]. Cognitive understanding of the scene can only be achieved after detecting and recognizing all kinds of targets in the scene. However, in the military reconnaissance, we often need to quickly identify local image scenes with definitive semantic meaning such as ports, airports, oil depot from high-resolution remote sensing images. These local scenes often correspond to a complex target group, contain many different types of rigid sub-targets according to certain spatial topological relations [2]. If by conventional method, detect subgoals of complex target group firstly, then to identify the complex target group of semantic properties by reasoning the spatial topological relations, but the processing efficiency is very low. And the analysis of sub-target is mainly on pixel level by using its implicit heuristic features. There is a significant gap between this complete pixel-level processing and image semantic understanding. It is very important to establish a semantic recognition method to classify complex local scene in remote sensing image quickly. And the scene semantic recognition result will be useful for the detection and recognition of specific targets in large remote sensing images. In this paper, related concepts and schemes of image scene recognition are first summarized. Then visual feature representation method for remote sensing image is provided and also a semantic recognition algorithm based on PLSA is given. Finally it is verified by experiments in 10 typical remote sensing image scenes.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 378–384, 2020.
https://doi.org/10.1007/978-981-15-0187-6_44

2 Schemes of Image Scene Recognition

At present, most of the scene recognition methods mainly focus on the basic level of visual cognition. Firstly to extract the image features from the image, then use the obtained image feature to design the semantic classifier, to implement mapping from low-level features of image to high-level semantic. The main research methods include two categories: low-level feature modeling method and middle semantic modeling method.

There are mainly three types of methods in middle-level semantic modeling: [3], (1) Construct the semantic object, to describe the scene by detecting or identifying semantic objects in images. (2) Torralba and Oliva proposed Gist model, the model avoids the segmentation of single object or region, to describe the scene structure by a low dimensional space envelope. Among them, five kinds of sensory attributes, such as nature, degree of openness, roughness, expansion, and roughness, separately corresponding to one dimension in space envelope. And each dimension as the basis of semantic division of scene corresponds to a meaningful spatial attribute in the scene. (3) Establish the local semantic concepts of image [4]. First, detect interest points automatically in image, and describe these points by the local descriptor, then establish the mapping of local descriptors to some local semantic concepts, realize the image scene recognition by distribution of the local image semantic concept. This paper mainly adopts this method for scene recognition of remote sensing images.

3 Visual Bag of Features of Remote Sensing Images

In order to realize the accurate recognition of remote sensing image scenes, no matter adopt low-level feature modeling method or middle semantic modeling method, must both extract discriminating features such as regional characteristics, sub-block features, local invariant features from remote sensing images. Because different characteristics reflect different types of information, it has its own advantages for a specific category. And in many cases, the analysis of the image content also needs to combine different characteristics. So the integration of multiple features is useful for improving the performance of image scene recognition. Bag of Words [5] is the text simplified description model used most commonly in text processing field. The model does not consider the grammar and word order, only expresses the text into a disordered word group. In text classified applications, often combined BOW model and the SVM classifier, simple Bayesian classifier, obtained very good classification results [5]. After the model is applied to the field of computer vision, it is popularized to Bag of Features (BOF). Which basic principle is to generate visual words to describe the image or image set by vector quantifying of various kinds local visual features.

The realization of BOF expression in remote sensing image is that detect the key point of image firstly, then calculate the local descriptor around key points, quantize the descriptor into visual words, and format visual words. Measure the frequency of the specific visual words in the image, and finally establish the visual bag of features of the image.

In order to capture the low-level visual features of different scenes accurately, we can find some discriminating key points in the image. Such as corner point, edge, center point of specific area, endpoint, center of gravity, extreme point, intersection, etc.) [6]. The most common key point is the corner point, which corresponds to the dramatic changes in image brightness or curvature maximum point in edge curve of image. The corner of a large number of features, such as road intersection in remote sensing images is presented as corner points. The neighborhood of the key points is also a region which has stable and rich information in the image. Because the difference of geometric relationship between remote sensing image and natural image is large, different imaging viewpoint often leads same scene in remote sensing images to appear larger difference. So it is necessary to search key point feature extraction algorithm which maintains a certain invariance to image changes such as rotation, scaling, affine transformation, angle of view, illumination changes and so on and maintains good matching with the object motion, cover, noise, and other factors.

The feature extraction algorithm SIFT (Scale Invariant Feature Transform) proposed by David G. Lowe not only has invariance to the scale and rotation of the image but also has certain stability to the brightness and the angle of view. So key points of the image can be detected effectively by applying SIFT algorithm. Because the scale range of remote sensing image has larger variation, and the scene types are complex and diverse, so it is obviously low efficiency if segment region first and then extract feature. And local invariant feature points are usually sparse. Although they have good effect on the natural images, it is not robust enough for the complex and changeable remote sensing image.

The classical SIFT operator is effective for grayscale images, Considering the spectral feature is an important sign to distinguish the type of remote sensing image scene, this paper introduced the spectral information, which can help to improve the performance of scene recognition based on visual feature representation. Firstly, the normalized multi-spectral remote sensing image is transformed into HSV space, and the image is divided into a dense grid of $n * n$, SIFT feature description of the grid center points are calculated on the HSV channels. And Splice them together into a 384-dimensional vector as CSIFT characteristics of the grid. Do K-means clustering of CSIFT features extracted from all the training images. And M visual words corresponding to M clustering center is obtained.

For an under-recognized remote sensing image (or region), the CSIFT feature is extracted by the same method as the training image. According to the nearest neighbor rule, determine the visual words classification of each CSIFT. Count the frequency of the visual words in the remote sensing image (or region) to be classified, and obtain visual bag of features of the under-recognized remote sensing image. The visual bag of feature of remote sensing image avoids by the process of target segmentation and detection in the scene, and transforms the identification of the scene into the learning problem of visual word distribution. So build a bridge for “the semantic gap” between low-level features and high-level semantics.

4 Semantic Recognition of Scenes Based on PLSA

Although the frequency of visual words can be used as an important basis for distinguishing different scenes, in a complex remote sensing image scene, the same target entity may appear in different scene categories. So it may lead to ambiguity and similarity problems between visual words and scene semantics. In the case of insufficient training samples, by using an identification method that associates the scene categories with the extracted feature vector directly, it cannot approximate the actual scene semantics and will reduce accuracy of scene recognition. Our thinking is to apply the probabilistic latent semantic analysis model to typical training image, to extract the latent semantic of image. And according to the probability distribution of latent semantic, determine the scene type of identified images.

(1) Extract the characteristics of all images. Selected part of image randomly from each class of training image, extract CSIFT feature vector of these images, and generate M visual vocabularies by k-means clustering algorithm. To measure the similarity of each visual word with the feature vector of each training image, and obtain “image–Words” co-occurrence frequency matrix $n_0(d_i, w_j)$ which is $N * M$ dimension. Among them $i \in (1, N), j \in (1, M)$. It represents frequency of visual words w_j in image d_i.

(2) Obtain the approximate maximum likelihood estimation result of PLSA model by using EM algorithm, And then obtain distribution $p(w_j|z_{\mathrm{Train}})$ of visual words when latent semantic appear in the image.

(3) Extract the feature vector of the test image. A similar measure respectively with M visual vocabularies obtained in (1). Obtain “image–Words” co-occurrence frequency matrix $n_T(d_T, w_j)$ of test image. The $p(w_j|z_{\mathrm{Train}})$ and co-occurrence frequency matrix $n_T(d_T, w_j)$ of test image is used as the input of the PLSA model, remain $p(w_j|z_{Train})$ unchanged. The latent semantic distribution $p(z \mid d_T)$ of the test image is obtained. And K dimensional semantic vector of the test image is built.

(4) The KNN classifier is applied to the latent semantic vector of the test image, and the scene recognition of image is accomplished.

5 Experimental Results and Analysis

Total 1794 image slices from 10 different type scenes are acquired from the Earth Google. The resolution and size of the image are not limited. The scene type is determined by the main object of the slice. 50 images are selected randomly from the 10 types of images as training images and the rest as testing images.

In order to verify the performance of the algorithm in this paper, firstly, we analyze the effect of different feature extraction methods with the recognition results, further compare the recognition effect by nearest neighbor classification between BOF expression based on CSIFT features directly and introducing PLSA model. Finally, compare the recognition results with different visual words number and different latent semantic topics.

5.1 A Comparison by Feature Extraction Method of Different Low Level

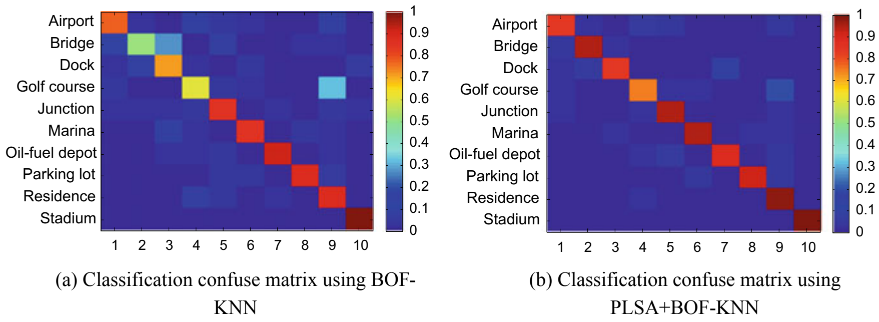
In this paper, the method of visual word generation is CSIFT feature, and the SIFT feature is mainly aimed at gray images. So convert the color image into grayscale image first when extracting the feature. When the feature is extracted, the sampling interval of dense grid is 8 * 8, and the number of visual words is 600, and the number of latent semantic topics is 20. The experiment results obtained from description methods of different low-level feature is shown in Table 1. Using CSIFT features as a low-level feature is better than the conventional SIFT feature based on gray level. Only for the "depot oil-fuel" scene, the performance of SIFT feature based on gray level is a little better. This is mainly because the target of the main position in this kind of scene is some cylindrical oil storage tank, Its shape feature is the most effective distinguishing feature. And the tone difference of oil depot in different regions is large, therefore, for this type of scene, the advantages of CSIFT are not obvious. As far as average recognition rate of 10 targets, CSIFI is 90.2%, SIFT is 79.67%.

Table 1. The results of scene recognition using different low-level feature extraction methods

Scene type	1	2	3	4	5	6	7	8	9	10
SIFT	0.783	0.492	0.719	0.560	0.846	0.857	0.923	0.891	0.896	1.0
C SIFT	0.838	0.953	0.855	0.741	0.944	0.941	0.875	0.910	0.963	1.0

5.2 The Improvement of Recognition Result by PLSA

The algorithm in this paper is realized by introducing the KNN model which is trained by PLSA model and based on the BOF expression of the remote sensing image, mark PLSA+BOF-KNN. In fact, the BOF expression of remote sensing image is obtained. We can also train the KNN classifier for scene recognition directly, denoted as BOF-KNN. Still set sampling interval of dense grid is 8 * 8, and the number of visual words is 600. The identification result is given in the form of a classification confusion matrix, as shown in Fig. 1. The result of recognition using BOF+KNN show some obvious ambiguity, such as bridge & dock and golf course & residence, because large number of visual words are shared among different type of scenes. We may eliminate such phenomenon and improve the performance of scene recognition by introduce the PLSA.



(a) Classification confuse matrix using BOF-KNN

(b) Classification confuse matrix using PLSA+BOF-KNN

Fig. 1. Scene recognition result using different methods

5.3 Effect of Different Visual Words on the Recognition Results

In the previous experiment, the number of visual words is set to 600. Next just adjust the number of visual words, sampling interval of dense grid and latent semantic number is unchanged. Compare the final average recognition rate. The results are shown in Fig. 2. The overall performance is relatively good when the number of visual words is between 500 and 900. Less number of visual words will lead to a lack of discriminating latent semantic topics. A large number of redundant information in the visual words increases the ambiguity of the latent semantic topics, so it will lead to a decline in overall performance.

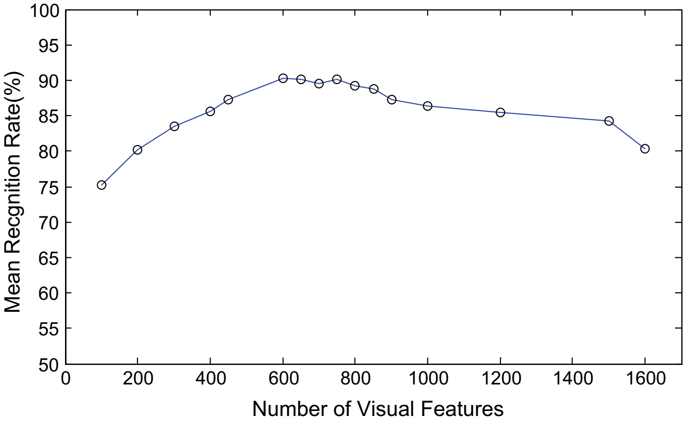


Fig. 2. The influence of different visual words numbers on recognition result

5.4 Effect of Recognition Results Caused by Different Number of Latent Semantic Topics

In the experiment, the number of the scene is 10, and the method of visual word extraction is unchanged. The number of fixed visual words is 600, and the number of latent semantic topics varies from 8 to 50. Compare the average recognition rate and the result is shown in Fig. 3. The average recognition rate is the highest when the latent semantic topic number is 20. Too many or too few potential semantic topics will lead to decrease in the rate of recognition. Although the optimal number of visual words and latent semantic topics exist in theory. It is very difficult to solve exactly in practical application. It is mainly through a lot of experiments to determine an empirical value.

In order to further verify the feasibility of the method of scene semantic recognition, the recognition model is trained in the aforementioned experiments, and a remote sensing image in North Korea Pyongyang area is used to make an interactive annotation. Taking into account the recognition model carry out specific targeted training only for 10 kinds of typical scenes, so in the interactive selection, we try to select the aforesaid 10 categories of scenes which can be manually identified. Identification result shows that there are better recognition results in different types of scenes.

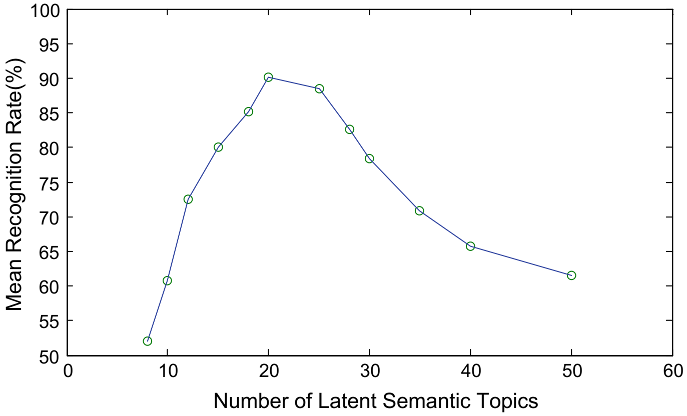


Fig. 3. The influence of different semantic topic numbers on recognition result

6 Conclusion

The semantic recognition of image scene is an important content of high-resolution remote sensing image interpretation. In this paper, related concepts and schemes of image scene recognition are firstly analyzed, then visual feature representation method for remote sensing image is provided and also a semantic recognition algorithm based on PLSA is given. Experiment on recognition of ten typical remote sensing image scenes verified the effectiveness of proposed algorithm. The recognition model obtained by training can be further applied to the large remote sensing images for interactive annotation.

References

1. Zhao R, Grosky WI, Negotiating the semantic gap: from feature maps to semantic landscapes. In: Pacholski L, Ruička P (eds) SOFSEM 2001, LNCS 2234, 33–52
2. Kumars H, Sivaprakash P (2013) New approach for action recognition using motion based features. In: Proceedings of 2013 IEEE conference on information & communication technologies. IEEE Computer Society, Washington DC, USA, 1247–1252
3. Liang Z, Wang X, Huangr et al (2014) An expressive deep model for human action parsing from a single image. In: Proceedings of 2014 IEEE international conference on multi-media and expo. IEEE Computer Society, Washington DC, USA, 1–6
4. Zhang X, Zhang H, Zhang Y et al (2016) Deep fusion of multiple semantic cues for complex event recognition. IEEE Trans. Image Process. 25(3):1033–1046
5. Pang L, Zhu S, Ngo CW (2015) Deep multimodal learning for affective analysis and retrieval. IEEE Trans Multimedia 17(11):2008–2020
6. Kim H, Kim J, Oh T et al (2017) Blind sharpness prediction for ultra-high-definition video based on human visual resolution. IEEE Trans Circuits Syst Video Technol 27(5):951–964

Applications of Brain-Inspired Intelligence in Intelligentization of Command and Control System

Shuangling Wang(✉), Chao Liu, Songhua Huang, and Kan Yi

Science and Technology on Information Systems Engineering Key Laboratory,
Nanjing, Jiangsu 210007, China
wsltongxin@163.com

Abstract. In this paper, we analyze the development tendency of the intelligentization of command and control system and explore the application advantages of brain-inspired intelligence over traditional artificial intelligence methods in the intelligentization of command and control system. We also present a new approach through which general intelligence and system intelligence can be realized in command and control systems, and provide the evolution path of the invocation pattern of intelligent algorithms in command and control system.

Keywords: Command and control system · Brain-inspired intelligence · Military intelligence

1 Introduction

Recently, artificial intelligence (AI) and related techniques are penetrating into military applications step by step. AI is regarded as a representative technique that drives the organization of warfare forms to transform from data center and information center into cognition/action center. It will promote the evolution of command and control system from information-oriented to intelligence-oriented [1].

The discipline of AI origins from the vision that intelligent systems can think as humans do in the future [2]. After its first proposition in 1956, there have been two distinct comprehensions toward the discipline of AI, namely, the view of engineering and the view of science. The view of engineering considers AI as an important branch of computer science, aimed at enabling machines fulfill tasks which only human beings can do, by means of computer programming. The category of view of engineering ranges from the initial expert system to the recent fast-growing deep neural networks. In contrast, the view of science considers AI as part of cognition science. It emphasizes that AI must base on the cognition principles of human beings, or it cannot be termed as AI [3].

Inspired by the information processing mechanism of brains [4, 5], brain-inspired intelligence is a branch of AI which falls under the category of the view of science. Borrowing the experiences from cranial nerve mechanism and cognitive behavior mechanism, it achieves the effect of machine learning by investigating the biological

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 385–390, 2020.
https://doi.org/10.1007/978-981-15-0187-6_45

structure of brains, analyzing the mechanisms of biological processes of brains such as growing, learning and memorizing, and building models to simulate the structure and mechanism of brains [6–10].

Brain-inspired intelligence is featured by self-learning, self-adaption, collaborative cognition of multiple brain areas, etc. It provides a technical approach for command and control systems to borrow the experience from cognition processes of human brains, to achieve the intelligent operation of functional domains such as situation awareness, command, and decision-making, action control, support and guarantee, as well as collaborative operation between them.

2 Development Tendency of Intelligentization of Command and Control System

Military command and control system has been developing from the first generation, which is featured by single-arm and single-point, to the fourth generation under comprehensive construction nowadays, which is featured by inter-arm flat networking [11]. It has entered the stage of system of systems (SoS) construction. The command and control system will realize intelligent sharing across war zones in the future. This sharing will be knowledge sharing, rather than information sharing merely, thus facilitating advantages in decision-making and head start over the enemy.

In order to realize the knowledge sharing and ability evolution, the command and control system should achieve the transformation from specialized intelligence to general intelligence, and transformation from intelligence of point to intelligence of SoS [12].

(1) From specialized intelligence to general intelligence

At present several specialized AI modules have already been introduced into the command and control system, changing some kinds of information into knowledge, such as prediction of plane types and prediction of flight paths. These modules can only deal with specific issues and are not universal. Thus we call them specialized AI systems. With the complication of the environment of battlefield, command staffs may raise new intelligentized requirements at any time based on the actual situation, rendering the problem of exploding demands. This requires that AI systems be universal, in order to quickly understand knowledge in different forms, and quickly make decisions.

(2) From intelligence of point to intelligence of SoS

Now AI modules are especially aiming at single domains and providing local intelligent services such as activity pattern analysis, target identification, intelligent issues, study and judgment of situations, etc. The intelligence ability of SoS have still not formed. In the future, faced with combat between SoS, we will develop intelligence abilities such as operational planning based on swarm intelligence, feedback learning based on combat effects, manned and unmanned autonomous collaborated engagement, etc.

3 Application Advantages of Brain-Inspired Intelligence

Brain-inspired intelligence shows quasi-man features in the matter of adaptation. It provides a new technique path for the formation of general intelligence and intelligence of SoS, as well as for the development of military intelligence.

Intelligentization of command and control system is faced with such special conditions and requirements: (1) Limited data resources. Compared with applications in Civil Internet, the sample capacity available in military activities is quite limited, since military activities are accidental and less frequent. Thus it is often very hard to collect thousands of or tens of thousands of samples for one kind of weapon. (2) Lots of low-quality samples. Because of the confidentiality of military activities, there are lots of samples with unclear features. (3) Exploding battlefield requirements. With the dynamic change of the situation of battlefield, command staffs and warfighters may raise new intelligentized requirements at any time according to the current situation.

However, at present, the AI techniques are faced with such bottlenecks: (1) Huge demand for data. The learning process relies on huge amount of sample data. (2) Strict requirement for data quality. There is strict requirement for the sample articulation, sample integrity, and label accuracy. (3) Strong dependency on models. The accuracy of models strongly depends on the source and quality of sample data, and these models are only suited for specific issues.

Brain-inspired intelligence is precisely a kind of adaptable AI, whose characteristics of learning with small samples, anti-noise ability, and general intelligence can serve the command and control system with a new way to solve the above problems. (1) Learning with small samples. While human beings can draw inferences about other cases from one instance, machines are still seeking hard for the similarity between new instances and what they have seen before even after training with tens of thousands of examples. Machines often can hardly identify the ships/planes if there are a small amount of training examples. In contrast, brain-inspired intelligence provides machines with the ability of drawing inferences like human brains, by multi-modal awareness and learning processes with synergy of multiple brain areas [6]. (2) Anti-noise ability. Human beings can identify the planes and ships in a photo taken in the rain or in the mist, as well as capture the essence from fragmentary pictures. However, machines are usually lost in excessive concern with local features. Therefore the rate of identification seriously drops when the pictures are noisy or fragmentary. By comparison, brain-inspired intelligence reproduces the magical operating mechanism of human brains by introducing neuron models with the ability of excitation and inhibition, making machines capable of self-adaptive denoising [7]. (3) General intelligence. Under control of brains, human beings have the skills of reading, thinking, learning, etc. However, AI at present is often only capable of doing a single job, for example prediction of plane types or flight paths. Although in some recent studies, deployment of algorithms as required is realized through construction of algorithm libraries, improving the problem of fixed algorithm function to some degree, it is still quite less flexible compared with humans. In contrast, based on the synergy mechanism between brain areas, brain-inspired intelligence performs self-learning and knowledge reasoning by constructing pulse neural network, making machines capable of carrying out multiple tasks [8].

4 Evolution Path of the Invocation Pattern of Intelligent Algorithms in Command and Control System

According to the trend of intelligentization, the invocation pattern of intelligent algorithms in command and control systems can evolve through three stages: specialized invocation of specialized algorithms, invocation of general algorithms as required, and dynamic adaption based on brain-inspired software architectures. Then the intelligent service under general intelligence and intelligence of SoS can be gradually realized.

The first stage is specialized algorithms directed at specific tasks for specialized invocations. In this stage, the intelligent supporting environment which provides support for development, operation, train, and services can be constructed based on existing cloud platforms, as shown in Fig. 1a. Thereinto, in the intelligent data management layer, data is accumulated by construction of basic model libraries, specialized model libraries, mapping knowledge domains and so on; in the intelligent computation framework layer, computation engines as well as specialized intelligent algorithms, such as object identification and graphic searching, are deployed to provide service framework for intelligent computation; in the intelligent service supporting layer, services like planning and reasoning, intelligent detection, etc. are formed, providing services and operating environment for typical applications such as situation awareness, command and decision, so as to realize specialized invocation of specialized algorithms.

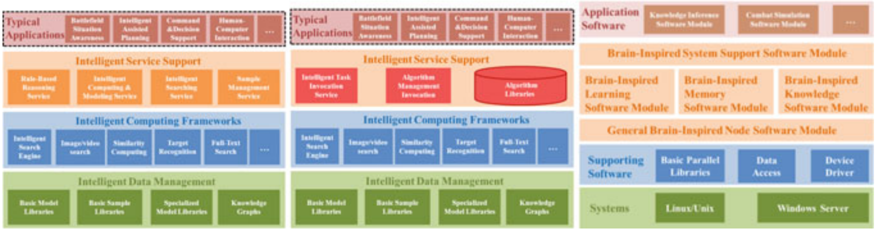

Typical Applications
Battlefield Situation Awareness
Intelligent Assisted Planning
Command &Decision Support
Human-Computer Interaction
...
Intelligent Service Support
Rule-Based Reasoning Service
Intelligent Computing & Modeling Service
Intelligent Searching Service
Sample Management Service
Intelligent Computing Frameworks
Intelligent Search Engine
Image/video search
Similarity Computing
Target Recognition
Full-Text Search
...
Intelligent Data Management
Basic Model Libraries
Basic Sample Libraries
Specialized Model Libraries
Knowledge Graphs
Typical Applications
Battlefield Situation Awareness
Intelligent Assisted Planning
Command &Decision Support
Human-Computer Interaction
...
Intelligent Service Support
Intelligent Task Invocation Service
Algorithm Management Invocation
Algorithm Libraries
Intelligent Computing Frameworks
Intelligent Search Engine
Image/video search
Similarity Computing
Target Recognition
Full-Text Search
...
Intelligent Data Management
Basic Model Libraries
Basic Sample Libraries
Specialized Model Libraries
Knowledge Graphs
Application Software
Knowledge Inference Software Module
Combat Simulation Software Module
...
Brain-Inspired System Support Software Module
Brain-Inspired Learning Software Module
Brain-Inspired Memory Software Module
Brain-Inspired Knowledge Software Module
General Brain-Inspired Node Software Module
Supporting Software
Basic Parallel Libraries
Data Access
Device Driver
Systems
Linux/Unix
Windows Server

Fig. 1. Evolution diagram of the invocation pattern of intelligent algorithms in command and control system: **a** specialized invocation of specialized algorithms (left), **b** invocation of general algorithms as required (middle), and **c** dynamic adaption based on brain-inspired software architectures (right).

The second stage is management and invocation of algorithm resources according to tasks, directed at the problem that different task environments have different requirements for intelligent algorithms. Based on the first stage, in this stage algorithm libraries are constructed in the intelligent service supporting layer of the intelligent supporting environment. As shown in Fig. 1b, algorithms are integrated and encapsulated into services which can be directly oriented to special applications, thus realizing invocation of general algorithms as required.

In the third stage, the general intelligence ability of command and control system is generated gradually, through the adoption of brain-inspired intelligence techniques.

The brain-inspired software architecture will be built as shown in Fig. 1c. Supported by common operating systems such as Linux/Unix or Window Server, software modules corresponding to various nodes are realized, such as general brain-inspired nodes, brain-inspired learning nodes, brain-inspired memory nodes, and brain-inspired knowledge nodes. By interconnection of brain-inspired nodes, the mechanisms and functions of processes like learning and memory of super-brain network system are achieved, supporting the command and control system for the brain-inspired cognitive pattern, providing dynamic adaptive intelligent services for the fight.

From the above evolution path, we can see that the dynamic adaption based on brain-inspired software architecture is the advanced stage of the invocation pattern of intelligence algorithms in command and control systems. Based on brain-inspired software architecture, combined with brain-inspired algorithms such as autonomous reinforcement learning, the construction of super-brain network system can achieve general intelligence ability for command and control system to some degree, in ways of speech recognition, image analysis, knowledge reasoning, augmented reality, natural language processing, etc. Moreover, it provides intelligent support for domains such as situation awareness, command and decision, action control, support and guarantee, etc., thus achieving fast awareness of battlefield situation, scientific command and decision-making, agile control of actions as the occasion requires, and real-time guarantee of material support, gaining the ability of SoS intelligence.

5 Conclusion

In this paper we analyze the development tendency of intelligentization in military command and control system, which will be from specialized intelligence to general intelligence, and from intelligence of point to intelligence of SoS. The advantages of brain-inspired intelligence in learning with small samples, anti-noising, and general intelligence, contribute to promoting general intelligence and SoS intelligence in command and control system. Besides, the evolution path of invocation pattern of intelligent algorithms in command and control system can be divided into three stages: specialized invocation of specialized algorithms, invocation of general algorithms as required and dynamic adaption based on brain-inspired software architectures.

References

1. U.S. DoD Defense Science Board, (2016) Summer study on autonomy. http://autonomousweapons.org/department-of-defense-science-board-summer-study-on-autonmy/
2. Turing AM (1950) Computing machinery and intelligence. Mind 49:433–460
3. Wilson R, Keil F (1999) The MIT encyclopedia of the cognitive sciences. MIT, Cambridge, MA
4. Maass W (1997) Networks of spiking neurons: the third generation of neural network models. Neural Netw 10(9):1659–1671
5. Dayan P, Abbott LF (2003) Computational and mathematical modeling of neural systems: Model neurons I: Neuroelectronic. MIT Press, Cambridge

6. Zhao F, Zeng Y, Xu B (2018) A brain-inspired decision-making spiking neural network and its application in unmanned aerial vehicle. Front Neurorobotics 12(56)
7. Zeng Y, Liu C-L, TAN T-N (2016) Retrospect and outlook of brain-inspired intelligence research. Chin J Comput 39(1):212–222
8. Zeng Y, Wang G, Xu B (2016) A basal ganglia network centric reinforcement learning model and its application in unmanned aerial vehicle. IEEE Trans Cognit Dev Syst, 1–15
9. Eliasmith C (2013) How to build a brain. Reprint edition. Oxford Press, New York
10. Eliasmith C, Stewart TC, Choo X, Bekolay T, DeWolf T, Tang Y et al (2012) A large-scale model of the functioning brain. Science 338(6111):1202–1205
11. Wang H, Zhang J, Guo C, Duanmu Z, Yi K (2017) Network operation method for C ∼ 4ISR system based on collaboration mechanism of system resources. Command Inf Syst Technol 8(1):15–21
12. Wang S, Huang S, Yi K, Mao X, Zhang Z (2018) Brain-inspired intelligence and applications in next generation command and control systems. Command Inf Syst Technol 9 (5):25–30

The Integrative Technology of Testability Design and Fault Diagnosis for Complex Electronic Information System

Leqing Ou and Fang Bai(✉)

The 28th Research Institute of China Electronics Technology Group Corporation,
Nanjing 210007, China
baifang@nuaa.edu.cn

Abstract. Complex electronic system has complicated principal and excess amount of single equipment and malfunctions, which makes it difficult to run quick failure diagnosis and thus security was low. On the basis of directed graph model and correlation model, this paper explores the integrative technology of testability design and failure diagnosis for complex electronic information systems. Using vehicle electronic information system as an example establishes functional directed graph fault model and analyzes its testability, which provides reference for quick fault diagnosis.

Keywords: Complex electronic information system · Testability design · Fault diagnosis

1 Introduction

Testability model is mainly used in describing the logical relationship between system failure and testability and the occupancy of test resources. It provides effective design and evaluation for testability design. The testability model of the system is different from the system model. The system model is mainly used to describe the structure, behavior and function information of the system. Testability model is often seen equivalent to fault model. Fault model is a standard abstract expression of fault information, and it is a series of definitions for describing system diagnosis knowledge and functional characteristics [1–4].

There are two main types of testability models: one is structured model. Similar to the system schematic diagram, the structural model represents the structure/function connection relationship and the direction of transmission in the form of directional diagram, while the corresponding relationship between the node of the structural model and the actual system module is represented by the directed graph. Testability analysis based on structural model is simple, fast and convenient for the testing and analysis of large-scaled systems. However, structure is not equivalent to function, especially for the complex function dependency embedded into simple block diagram. Thus, to obtain the complete function dependency, structured model cannot be fully satisfied [5, 6].

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 391–397, 2020.
https://doi.org/10.1007/978-981-15-0187-6_46

Another type of model is correlation model, such as reasoning model, diagnostic reasoning model or functional correlation model. The causal relationship between model failure source and test or between test and test is needed to be expressed by directed graph. Due to the inherent simplification characteristics of directed graph, correlation model is the main modeling technology in testability analysis tools at the moment.

The integrative technology of testability design and diagnosis for complex electronic information systems should focus on solving the problem of information modeling. Model is the quantitative description of system information and the foundation of system analysis. In the aspect of testability modeling, there are three representative models, which are testability block diagram description model, information flow model, and multi-signal flow graph model. The testability block diagram description model is a structured model, while the information flow model and the multi-signal flow graph model are testability design model based on correlation. The multi-signal flow graph model is also a directed graph model. The most representatives of directed graph model are multi-signal flow graph model, Petri network model, and Bayes network model. It can be said that in the field of testability modeling, correlation and directed graph are the most representative theories. The correlation relationship can be expressed as Fig. 1.

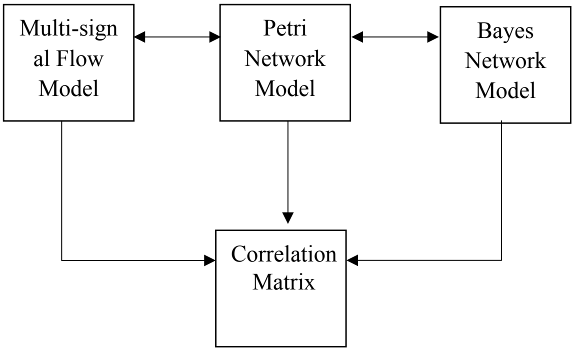


Fig. 1. The relationship between directed graph model and correlation model

2 Digraph Model (Digraph = Directed Graph)

2.1 Definition of Digraph

The structure of directed "graph" is composed of the set of vertices and the set of relations between vertices, which is:

$$G = (V, E)$$

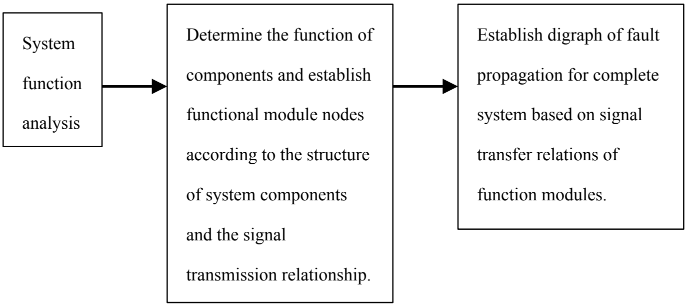
G is the total set of the graphs; V is the finite nonempty set of the vertices; E is the finite set of relations between vertices.

The vertex pair <*u*, *v*> of the digraph is in order in the graph, which represents a directed edge from vertex *u* to vertex *v*.

If the function module is represented by node *v* and the relationship *R* existed between the function modules is represented by the edge e connecting two points, the line graph $G = (V, E)$ is the digraph model corresponding to the system, where $v \epsilon V$, $e \epsilon E$ are all function module nodes and various connection relationships, respectively.

2.2 Diagraph Model

Diagraph model refers to the step-by-step analysis of system function structure and its signal transmission relationship in fault diagnosis based on parts or all of the functions of each component of the system until the specific location of the fault is determined, which is, the representation of the function model in the digraph is established first. Then analyze the cause of the fault and the propagation path of the fault. The functional modeling process is shown in Fig. 2.



System function analysis

Determine the function of components and establish functional module nodes according to the structure of system components and the signal transmission relationship.

Establish digraph of fault propagation for complete system based on signal transfer relations of function modules.

Fig. 2. Digraph functional modeling flow chart

Based on the analysis of the system composition and the function modeling method, establish the digraph function model of the system. Nodes are divided into functional modules. Each node represents a function, and constitutes a set of functional nodes $V = \{v_1, v_2, \ldots, v_n\}$ and directed edge $E = \{e_{ij}\}$ (where $i \neq j$) which represents functional relations constitute digraph *G*. The fault in digraph refers to the deviation of the function output when the system receives the wrong signal or its own fault. The directed edge refers to the fault propagation path between modules under the function module. The fault propagation path from node *i* to node *j* is represented by e_{ij}. If and only if the failure of node *i* directly results in the failure of node *j*, the node *i* and node *j* is the digraph can be connected by the directed line segment.

2.3 Fault Diagnosis Model Based on Digraph

Fault diagnosis model based on digraph consists of the following elements:

All possible sets of fault sources $F = \{f_1, f_2, \ldots, f_m\}$;

Sets of fault observation sensor $T = \{t_1, t_2, \ldots, t_n\}$;

Fault dependency matrix D for describing the relationship between the fault source and sensor output.

Consider graph $G = (V,\ E)$, where set V contains two sets of nodes: $F = \{f_1, f_2, \ldots, f_m\}$ and $T = \{t_1, t_2, \ldots, t_n\}$, and requires that F can be completely covered by T. $E = \{e_{ij}\}$ is the edge connecting node v_i and v_j, representing the information flow between nodes. Graph G captures the first-order causal dependency between nodes, which is v_i affect v_j or v_j affect v_k. On this basis, we can obtain system fault dependency matrix D. This is a dimension binary matrix $m \times n$ with fault source as row and fault observation sensor as column. It reflects the observation ability of sensor $t_j \in T$. If f_i fails and can be observed by t_j, then $d_{ij} = 1$; otherwise, $d_{ij} = 0$, which means that the fault source f_i cannot be observed by t_j. In order to analyze the observing ability of sensor $t_j \in T$ to the given fault source $f_j \in F$, we usually use fault propagation algorithm. This algorithm starts with any given fault source $f_j \in F$ and propagates the fault along $e_{ij} \in E$ to determine which subsequent nodes are affected by it. When the fault arrives at the node v_i, if v_i is the test node, the appropriate output is generated at the node and its copy is transmitted to the next node along the edge connected with v_i. In order to prevent the algorithm from entering a dead cycle, when another fault arrived at one node, other faults stop replication and propagation at that node. The termination condition of the algorithm is that the fault copy can no longer be propagated to any other new nodes.

3 Correlation Model

3.1 Definition of Correlation

Correlation refers to the causal relationship between an entity (physical or conceptual) and another entity. This correlation is directional. If z can be deduced from x, the z is related to x or z depends on x; if x and z can be deduced from each other, then they are interrelated.

Faults and testing have this correlation in the system. If the fault f_i occurs can derive that t_j fails and t_j pass can derive that f_i did not happen, then t_j and f_i are interrelated, or t_j is a symmetric test. If only the fault f_i occurs can derive that t_j fails, but not t_j pass can derive that f_i did not happen, then t_j is an asymmetric test.

First-order correlation: refers to the direct causality, such as z can be directly derived from x, which is called the first-order correlation of x and z. If a fault node is directly connected with a measurement point in the system, then they are called first-order correlation.

N-order correlation: also known as higher-order correlation, which refers to the indirect causality. For example, x can be derived from w, y can be derived from x, and z can be derived from y, it is said that z is related to w in order 3, y is related to w and z is related to x in order 2. In the system, if a fault node is connected to a measurement point through some intermediate nodes, they are called N-order correlation.

3.2 Correlation Graphic Model

After the reasonable division of structure and function, the representation method of correlation graphic model is to clearly indicate the connection between information flow and each component, and the location and serial number of the test points, so as to show the correlation between each component and test point, as shown in Fig. 3.

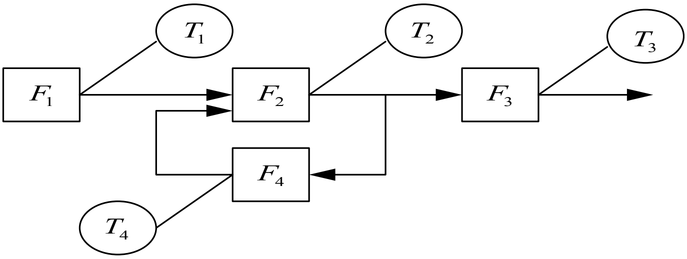


Fig. 3. Correlation graphic model

4 Sample Analysis

4.1 System Composition

Select a vehicle electronic information system as an example to verify the SDG fault diagnosis method. The system consists of 15 kinds, 40 equipment. The digraph model of system function is established as Fig. 4.

4.2 Result Analysis

After testing analysis, the result of fault detection (up to LRU level), fault coverage rate and fault isolation rate of a vehicle electronic information system are shown in Table 1, which is consistent with the actual situation.

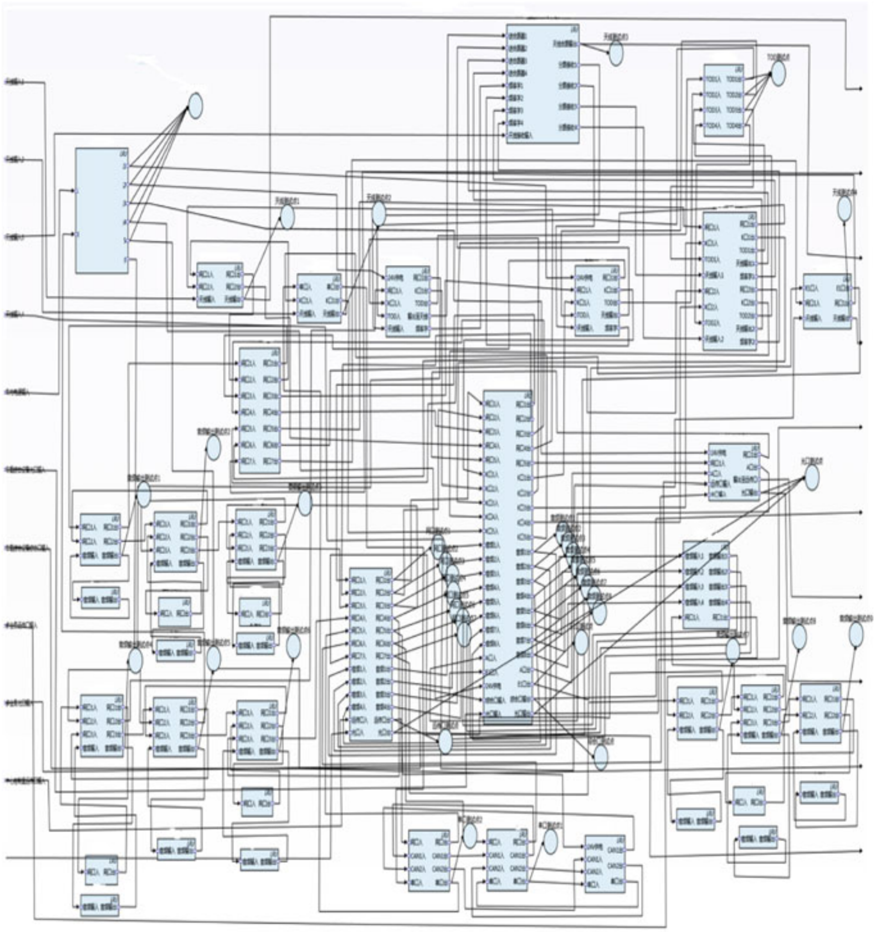


Fig. 4. Digraph model of a complex electronic system structure

Table 1. Fault detection rate

Test type	Fault detection rate	Fault coverage rate	Fault isolation rate		
			Reach one LRU	Reach two LRU	Reach three LRU
External test equipment	75.23%	79.15%	70.45%	78.89%	88.61%

5 Conclusion

Complex electronic system has complicated principal and excess amount of single equipment and malfunctions, which makes it difficult to run quick failure diagnosis. It is necessary to optimize its testability design at the beginning of the design, improve the testability of the system and improve the supportability of equipment use and maintenance. The article firstly analyzes the relationship between the testability model and fault model, as well as the current technology development status. Secondly, it introduces the definition of digraph, digraph model, fault diagnosis model based on digraph and the modeling method and process of correlation graphic model. On this basis, it proposes the integrated technology of testability design and diagnosis for complex electronic information systems. Taking vehicle electronic information system as an example, it establishes the function digraph model and analyzes its testability. The results show that the conclusion of the model analysis is consistent with the actual situation.

References

1. Mobed P, Maddala J, Pednekar P et al (2015) Optimal sensor placement for fault diagnosis using magnitude ratio. Ind Eng Chem Res 54(38):9369–9381
2. Liu YK, Su S, Yang Y et al (2013) Fault diagnosis approach for wind turbine based on signed directed graph. J Mech Strength 35(5):583–588
3. Xu XH (2010) Research on the fault diagnosis of aero-engine based on the AHP-SDG method. Nanjing University of Aeronautics and Astronautics, Nanjing
4. Li G, Gao J, Chen F (2007) Formal support for failure knowledge modeling and diagnostic reasoning using polychromatic sets. In: Proceedings of the 5th IEEE international conference on industrial informatics. IEEE Press, Piscataway, NJ, USA vol 2, pp 645–650
5. Kun Huang, Hongzhi Li (2011) Archiecture of data link integration application. Command Inf Syst Technol 2(5):15–18
6. Yuan L (2012) Information extraction of strategic intelligence research. Command Inf Syst Technol 3(1):49–52

Chinese Named Entity Recognition with Changed BiLSTM and CRF

Jie Ren(✉), Jing Liang, and Chenkai Zhao

University of Electronic Science and Technology of China, Chengdu, China
2356670185@qq.com

Abstract. This paper is aimed at improving name entity recognition (NER) accuracy. We replace the traditional bidirectional long short-term memory network (BiLSTM) with a changed BiLSTM, and then uses a CRF layer behind the changed BiLSTM layer to add the probabilistic relation of different Chinese characters.

Keywords: NER · CRF · Changed BiLSTM

1 Introduction

Name entity recognition (NER) is also known as proper name recognition. It is a fundamental task in natural language processing (NLP) and has a wide range of applications. Named entity generally refers to the entity with specific meaning or strong reference in the text, usually including person name, place name, organization name, and so on. NER is a foundation key task in NLP. At the same time, NER is also the basis of many NLP tasks such as relationship extraction, event extraction, knowledge mapping, machine translation, and question answering system.

NER has been a research hotspot in the field of NLP. At beginning, NER is based on dictionary and rule-based methods. Later, the traditional machine learning methods, especially probabilistic graph models such as hidden Markov model (HMM) [1,2], maximum entropy Markov model (MEMM), and conditional random field (CRF) [3,4], became the focus of NER's research. In recent years, NER based on deep learning such as long short-term memory (LSTM) has been become popular [5]. However, both probability graph model and deep learning model are incomplete. Probability graph model does not learn the deep hidden information in sentences and deep learning model does not consider the probabilistic relation between parts of speech and the parts of speech. Some scholars found that the combination of two methods deals with the above problems [8]. Huang Z, et al. used bidirectional long short-term memory (BiLSTM) and CRF to do NER, and the results is very good [6,7]. Then, Lample G, et al. proposed that use two LSTM and CRF can obtain deeper hidden information

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 398–406, 2020.
https://doi.org/10.1007/978-981-15-0187-6_47

and a better accuracy [9]. But their method did not consider the reverse meaning of texts. Also, Zheng S, et al. used a BiLSTM network to deal with this problem and a LSTM network to replace CRF to gain the deeper semantic [10], but they did not consider the probabilistic relationship.

To solve these problems, we propose an advanced BiLSTM network to achieve better deep semantics and probability relationships. We use two LSTM to train a sequence and return a sequence. What is more, we use two reverse LSTM to train the same sequence and return a sequence too. Then, we concatenate the two sequences to one sequence. Hence, we can obtain deep semantics and the relationships including the forward meaning and backward meaning together. Finally, we use a CRF to obtain the probabilistic relation.

In this paper, in Sect. 1, we introduce the background and the overall idea. In Sect. 2 , we introduce the important models in our algorithm. In Sect. 3 , we give out our experiment results. In Sect. 4, a conclusion is given.

2 Mathematic Models

2.1 Improved BiLSTM

The Disadvantage of RNN: Recurrent neural networks (RNN) are implemented by reusing a cell structure. Hence, the output of the current moment is related to the past.

The expression function of RNN is:

$$h_t = f(W \cdot x_t + U \cdot h_{t-1} + b), \tag{1}$$

$$y_t = V \cdot h_t, \tag{2}$$

where x_t and y_t are the input and output at time t, respectively. h_t is the memory information at time t, and $f(z)$ is an activation function, which is usually a tanh function. W, U, b, v are the parameters of the network.

Because RNN is implemented by reusing a cell structure, we just train W, U, b, v by iterations and do not need other parameters.

However, there is a disadvantage of RNN. It can not remember long time information because of gradient diffusion. The appearance of LSTM solved this problem.

The Advantage of LSTM: LSTM is a variant of RNN, which can effectively solve the gradient diffusion problem of simple RNN. It mainly improves the following two parts.

First, it adds a new internal state c_t and retains the original external state h_t. Hence, gradient diffusion is suppressed by combining linearity and nonlinearity. Second, it controls the amount of information transmitted through three gates, ensuring that the linear transmission does not lead to too much information. Therefore, its formulas are written as:

$$c_t = f_t \odot c_{t-1} + i_t \odot \hat{c}_t, \tag{3}$$

$$h_t = o_t \odot \tanh(c_t), \tag{4}$$

where f_t, i_t, o_t are forgotten gate, input gate, output gate, respectively. They are from 0 to 1. $\odot$ means the product of the value of the gate and each element in the vector. $\hat{c}_t$ is the candidate states obtained by nonlinear functions. The details are shown as follows:

$$\hat{c}_t = \tanh(W_c \cdot [x_t, h_{t-1}] + b_c), \tag{5}$$

$$i_t = \sigma(W_i \cdot [x_t, h_{t-1}] + b_i), \tag{6}$$

$$f_t = \sigma(W_f \cdot [x_t, h_{t-1}] + b_f), \tag{7}$$

$$o_t = \sigma(W_o \cdot [x_t, h_{t-1}] + b_o), \tag{8}$$

where $[x_t, h_{t-1}]$ means to concatenate x_t and h_{t-1}, together. σ is sigmoid function:

$$\sigma(x) = \frac{1}{1 + \mathrm{e}^{-x}}. \tag{9}$$

Because of these three gates and the internal state, LSTM can successfully remember the past information and delete the useless information.

Although LSTM is the most popular network in dealing with time sequence problems, it also has a small shortcoming. It just learn the forward information but does not learn the backward information, which has limitations in the language model.

The Meaning of BiLSTM: As we just mentioned above, sometimes, we need a network to study the forward information and backward information together, especially in text problem such as text classification, text translation, and NER.

BiLSTM is simple to understand. It has two LSTMs. The first one's input is from x_1 to x_T and its external state is from h_1^1 to h_T^1. The second one's input is from x_T to x_1 and its external state is from h_1^2 to h_T^2. Later, the concatenation of these two LSTMs' external state is $h_t = [h_t^1, h_t^2]$.

Our Changed BiLSTM: In this paper, in order to get the deeper information of sentences and the forward and backward text connection, we change the BiLSTM structure.

We divided the model into two parts. The first part is a forward LSTM connected to another forward LSTM. The first forward LSTM's input is the input data, and the second forward LSTM's input is the returned sequence from the first forward LSTM. The second part is a backward LSTM connected to another backward LSTM. Also, the first backward LSTM's input is the input data, and the second backward LSTM's input is the returned sequence from the first backward LSTM.

By using this structure, we can firstly get deeper meanings of forward and backward text sequences. Later, we combine the two states together and use the combinative result as the CRF's input.

2.2 Conditional Random Filed

The Model Definition: CRF is a conditional probability distribution model of one set of output variables with another set of input variables that are given. It is characterized by the assumption that the output random variables constitute Markov conditional field.

The Parametric Formalization: If $P(Y|X)$ is linear CRF and the value of random variable X, x is given, the value of random variable Y, y is shown as follows:

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_l, x, i)\right), \tag{10}$$

where t_k and s_l are characteristic functions which are always 0 or 1, λ_k and μ_l are corresponding weights, $Z(x) = \sum_y \exp\left(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_l, x, i)\right)$ is a normalization factor.

The parameters are learned by BFGS method, which is not described here.

The Prediction Algorithm: The CRF prediction problem is to find the output sequence y^* with the maximum conditional probability given the CRF $P(Y|X)$ and the input sequence x. This problem is usually solved by viterbi algorithm. Hence, the output sequence is written as:

$$y^* = \arg\max_w P_w(y|x) = \arg\max_w \frac{\exp(w \cdot F(y,x))}{Z_w(x)}, \tag{11}$$

and because $Z_w(x)$ is a constant and exp(.) is a monotone increasing function. Therefore,

$$y^* = \arg\max_y (w \cdot F(y,x)) \tag{12}$$

So the CRF prediction problem is called the optimal path problem with the largest probability of denormalization.

$$\max_y (w \cdot F(y,x)), \tag{13}$$

where the path means the tag sequence and

$$w = (w_1, w_2, ..., w_K)^{\mathrm{T}}, \tag{14}$$

$$F(y,x) = (f_1(y,x), f_2(y,x), ..., f_K(y,x))^{\mathrm{T}}, \tag{15}$$

$$f_k(y,x) = \sum_{i=1}^{n} f_k(y_{i-1}, y_i, x, i), \quad k = 1, 2, ..., K. \tag{16}$$

To solve the optimal path, (13) can be written as:

$$\max_y \sum_{i=1}^{n} w \cdot F_i(y_{i-1}, y_i, x), \tag{17}$$

where $F_i(y_{i-1}, y_i, x) = (f_1(y_{i-1}, y_i, x, i), f_2(y_{i-1}, y_i, x, i), ..., f_K(y_{i-1}, y_i, x, i))^{\mathrm{T}}$ is local characteristic vector.

Later, use viterbi algorithm. First of all, the denormalized probability of each mark $j = 1, 2..., m$ to position 1 is:

$$\delta_1(j) = w \cdot F_1(y_0 = \text{start}, y_1 = j, x), \quad j = 1, 2, ..., m. \tag{18}$$

In general, the recursive formula is used to find the maximum value of the denormalization probability of each mark $l = 1, 2, ..., m$ to position i:

$$\delta_i(j) = \max_{i \leq j \leq m} \delta_{i-1}(j) + w \cdot F_i(y_{i-1} = j, y_i = l, x), \quad l = 1, 2, ..., m, \tag{19}$$

$$\Phi_i(l) = \arg\max_{i \leq j \leq m} \delta_{i-1}(j) + w \cdot F_i(y_{i-1} = j, y_i = l, x), \quad l = 1, 2, ..., m. \tag{20}$$

When $l = n$, the maximum value of the denormalized probability is:

$$\max_y(w \cdot F(y, x)) = \max_{1 \leq j \leq m} \delta_n(j), \tag{21}$$

and the most optimal terminal:

$$y_n^* = \arg\max_{1 \leq j \leq m} \delta_n(j) \tag{22}$$

Return from the end of this optimal path:

$$y_i^* = \Phi_{i+1}(y_{i+1}^*), \quad i = n-1, n-2, ..., 1, \tag{23}$$

Finally, the most optimal path is $y^* = (y_1^*, y_2^*, ..., y_n^*)^{\mathrm{T}}$.

2.3 Our Model Structure

This part totally explains what we do and how to combine improved BiLSTM and CRF together.

Figure 1 is the whole structure of our network.

Firstly, because all the words in corpus are in the form of one-hot encoder and the dimension is very high, we use an embedding layer to find the words' low-dimension representation to reduce the learning time. Also, the low-dimension dense representation has better representation ability than the high-dimension sparse representation.

Secondly, we put the embedded words sequence into our improved BiLSTM to learn the forward and backward deep semantic meaning. Then, use a merge layer to concatenate the external state of the two-layer-forward LSTM and the external state of the two-layer-backward LSTM.

Thirdly, the combined external state is regarded as the input of CRF layer. CRF layer learns its parameters.

Finally, the output sequence of CRF layer is the name entities.

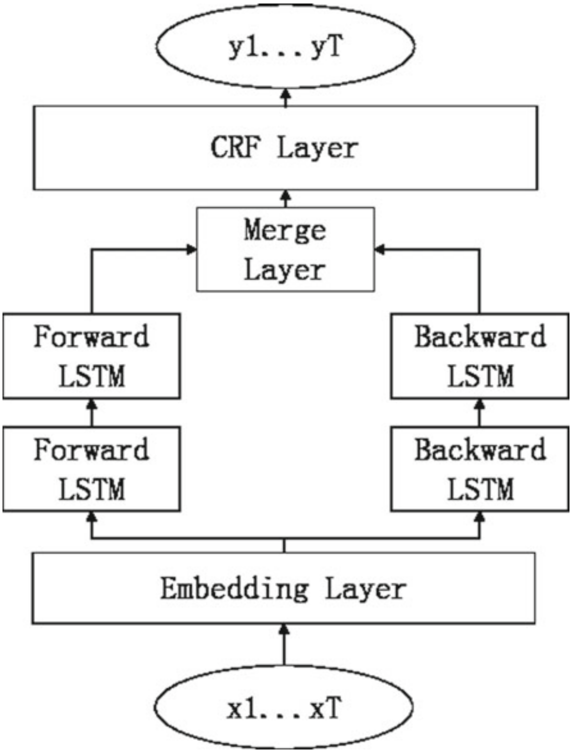


Fig. 1. Whole structure of our model

3 Experiment and Results

3.1 The Form of Experiment Data

In order to show whether our algorithm is good or not, we do an experiment with real data.

We download an open-source data of Chinese NER from the Internet. The data form is as following. Each sentence is separated by "\n\n". Each pair of Chinese character and NER label is separated by "\n". Each element (Chinese character and NER label) in the pair is separated by "\t". For example, "中\tB-LOC\n国\tI-LOC\n很\tO\n大\tO\n\n". And the labels are "B-PER", "I-PER", "B-LOC", "I-LOC", "B-ORG", "I-ORG", "O".

3.2 Data Preprocessing and Model Compile

Obviously, this data can not be used directly. There must be some preprocessing.

First of all, we combine the characters in each sentence as an input sequence, and we combine the NER labels in each sentence as an output sequence. However, the characters can not be used in a mathematic model. Hence, we count

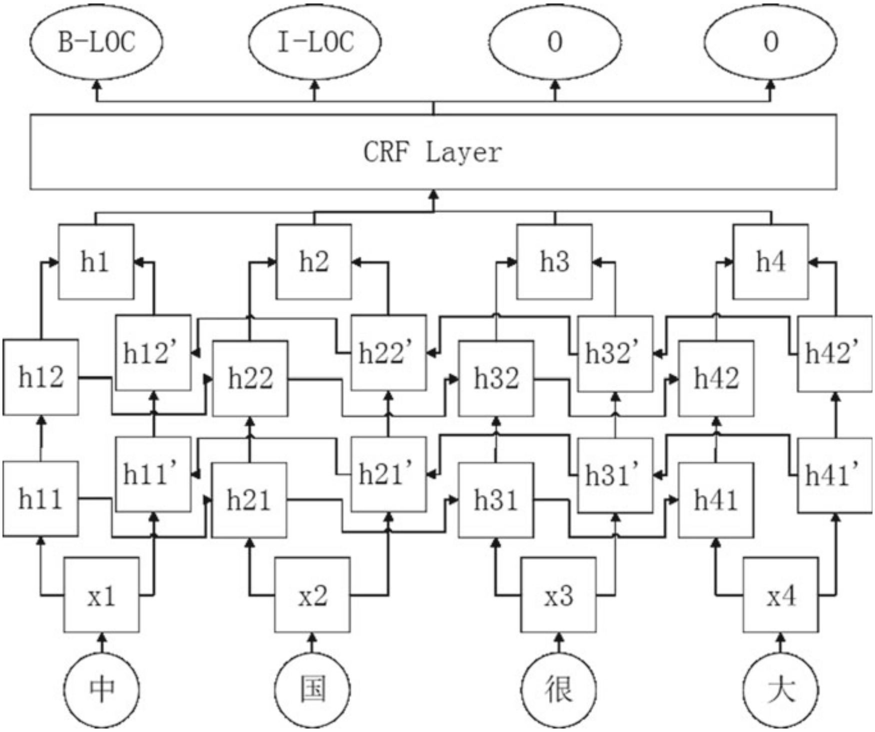


Fig. 2. Simple example of our experiment

the number of characters that appear and change each character to its one-hot encoder form.

Later, LSTM should have a fixed input dimension, which means that the input sequences should have the same length. So, we find the longest sequence and use zero-padding to make the shorted sequence has the same length as the longest one.

Finally, we use Keras to set up our model, and Fig. 2 is a simple example of our model.

In our experiment, we use 50658 training data and 4631 test data. The output embedding dimension is 200, and LSTM has 128 units. Also, there are batch normalization layer and dropout layer. The "categorical cross entropy" is the training loss and metrics, and we use "adam" as our optimizer.

3.3 Experiment Result

After training our model with, we firstly use a new text to predict whether our model has the ability to successfully analyze name entities. Here, we use a sentence "中华人民共和国国务院总理周恩来在外交部长陈毅的陪同下，连续访问了埃塞俄比亚等非洲10国以及阿尔巴尼亚", and the result is that "B-PRE" + "I-PER": ["周恩来","陈毅"] "B-LOC" + "I-LOC": ["埃塞俄比亚","非洲","阿尔巴尼亚"]; "B-ORG" + "I-ORG": ["中华人民共和国国务院","外交部"]. We can see that the result is totally correct.

Table 1. Comparison result of different algorithms.

Model	CRF viterbi accuracy (%)
Embedding + LSTM	95.71
Embedding + CRF	96.57
Embddubg + BiLSTM	96.61
Embddubg + BiLSTM+CRF	97.41
Our model	97.87

Then, we compared our algorithm with the traditional algorithms by using all test data, and the result is shown in Table 1.

From Table 1, we can find that our model has the best accuracy than other models.

However, because our model has deeper structure than traditional models, our training time is longer than other methods. Therefore, we add batch normalization layers and dropout layers to reduce training time. Also, we reduce the amount of units in LSTM layers appropriately. Finally, the comparison of training time is shown in Table 2.

Table 2. Comparison of training time

Model	Training time (mins)	CRF viterbi accuracy (%)
BiLSTM + BN + Dropout	126	96.59
BiLSTM + CRF + BN + Dropout	142	97.46
Our model	164	97.87
Our model + BN + Dropout	148	97.91

Table 2 shows that our model with BN and dropout layers can use the same training time as the traditional models to get a better CRF viterbi accuracy.

4 Conclusion

In order to solve the problem that traditional BiLSTM can not obtain deeper latent semantics, we change the BiLSTM model, and successfully combine it with a probabilistic graphic model (CRF layer) to gain a better prediction accuracy in NER problem. After experiments, we can find that it is useful in Chinese NER question, and it has the highest CRF viterbi accuracy.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (61731006, 61671138), and was partly supported by the 111 Project No. B17008.

References

1. Zhao S (2004) Named entity recognition in biomedical texts using an HMM model. In: International joint workshop on natural language processing in biomedicine and its applications. Association for Computational Linguistics
2. Ponomareva N , Pla F , Molina A et al (2007) Biomedical named entity recognition: a poor knowledge HMM-based approach
3. Koller D, Friedman N (2009) Probabilistic graphical models: principles and techniques. MIT Press
4. Lafferty J, McCallum A, Pereira F (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: International conference of machine learning
5. Zhang Y , Yang J (2018) Chinese NER using lattice LSTM
6. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. Comput Sci
7. Xu C, Xie L, Xiao X (2018 July) A bidirectional LSTM and conditional random fields approach to medical named entity recognition 90(7):1063–1075
8. Ma X, Hovy E (2016) End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF
9. Lample G, Ballesteros M Subramanian S et al (2016) Neural architectures for named entity recognition
10. Zheng S, Wang F, Bao H et al (2017) Joint extraction of entities and relations based on a novel tagging scheme

Key Problems and Solutions of the Application of Artificial Intelligence Technology

Guangxia Zhou(✉)

Science and Technology on Information Systems Engineering Laboratory,
Nanjing 210007, China
zgx8086@163.com

Abstract. Google's AlphaGo shocked the world by easily defeating Korean Go player Lee Shi-shi, thus setting off a new upsurge of artificial intelligence research and application. Currently, artificial intelligence technology is developing at a speed beyond imagination, and it has become the core and key to the leap-forward development of every industry. However, application of artificial intelligence technology also encounters many problems. Key problems of military application of artificial intelligence are analyzed emphatically, and solutions to those problems are introduced, as a reference to further research.

Keywords: Artificial intelligence technology · Machine learning · Autonomous learning · Deep learning

1 Introduction

In March 2016, AlphaGo, developed by DeepMind, a subsidiary of Google, shocked the world by easily defeating Korean Go player Lee Shi-shi, thus setting off a new upsurge of artificial intelligence (AI) research and application. AI technology is developing at a speed beyond imagination, and it has become the core and key to implement leap-forward development in every industry. The military, of course, is no exception and has higher expectation. In military, AI is a revolutionary and enabling technology capable of improving mission tasks from intelligence gathering to situation predicting, decision-making and autonomous unmanned combat. Thus, world's military powers have continuously strengthened their research and application of AI and expect to seize the strategic commanding heights. However, compared with the civil application, the application of AI in military has great particularity, such as the restraint of war ethics. This paper does not discuss such issues, but only the related technical problems of AI military application. These problems include data, security, credibility, etc., which hinder the military application of AI technology. These problems must be solved before AI technology can be widely used in the military. Through the analysis of the exploration of AI military application at home and aboard, solutions to the above problems are suggested, as a reference to further research.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 407–414, 2020.
https://doi.org/10.1007/978-981-15-0187-6_48

2 Problem Analysis

Generally speaking, the main problems of the application of AI technology in military are as follows:

First is data. Data is the lifeblood of AI. Without dataset, AI has become a passive water. Learning has neither object nor verification basis. Data problems include three aspects: Firstly, the problem of data source—in peacetime, operational data mainly comes from the aggregate of daily duty, live exercise, training and other ways. However, these aggregates are far from enough to effectively support machine learning. Even the USA, which has fought the most war in the world in recent years, has the richest aggregate of battlefield data in the world, but its Defense Innovation Unit (DIU) still thinks that the key to restricting the application of AI in the military is that there are too few machine learning datasets available in a short and strong competed environment, which makes it difficult for AI systems to carry out tasks effectively in the complicated and competed environment [1]; secondly, the problem of data comprehensiveness—the rapid development of AI technology is mainly due to the continuous development of deep learning system. By deep learning, a large amount of information can be inputted into the computer to allow deep learning system to learn and analyze data. However, the data used to train deep learning system is not comprehensive although it is constantly improving. For example, when using AI system to pick out fruits, people train visual recognition algorithms by learning a large number of image databases. The purpose is usually to identify the "natural" appearance of fruits. However, compared with the bright photographs of intact fruits, there are relatively few pictures of rotten fruits. And unlike humans, AI systems tend not to calculate or ignore them, while the human brain tends to pay attention to these abnormal groups and react strongly to them [2]. In military application scenarios, this situation is even more serious; and thirdly, the problem of data annotation—current machine learning systems acquire a large amount of data through sample learning, which has been independently labeled by analysts to generate the required output. With the development of the system, deep neural network (DNN) has become the latest technology in machine learning model. DNN can provide power for tasks such as machine translation and speech or object recognition with higher accuracy. However, training DNN requires a lot of labeling data. The process of accumulating and labeling a large amount of information is expensive and time-consuming. In addition to the challenge of accumulating labeled data, most machine learning models are fragile and vulnerable to collapse when their operating environments change slightly. For example, in speech recognition systems, if the acoustic environment or the sensor of the microphone changes, it may be necessary to retrain on a completely new dataset. The time and effort required to adjust or modify the model are almost equivalent to recreating the model.

Second is the problem of autonomous learning. At present, AI takes machine learning technology as its core and relies on huge dataset. However, the battlefield environment is complicated and changeable, the competition is violent, the situation is changing rapidly, full of fog and accident, and the AI system cannot predict all possible factors or scenarios when it is developed. In order to carry out its mission under any circumstances, the military AI system must have the ability of self-learning and quickly react to sudden and extreme situations on the battlefield.

Third is the problem of security. AI system is facing many threats in battlefield, including physical damage and cyber attack. Comparing with the hard damage caused by physical attack, the soft damage caused by cyber attack may be more serious and more difficult to prevent. The US's "Resilient Military Systems and Advanced Cyber Threat" describes the situation of missile control system being hacked and may be directed against own troops [3]. The software nature of AI system determines the inevitability of cyber attack. This is because the intelligent function of AI system comes from the software composed of various branch logics, variable tables and parameter tables [4]. The more complex the task and the more diverse the environment, the higher the complexity of the software, the wider the scope involved, the more vulnerabilities and weaknesses, the greater the possibility of being broken.

Fourth is the problem of credibility. In civil AI products, as long as the algorithm is effective, the others are not so important. How to allocate drivers and determine routes for Uber? How does Google generate search results? How does Tesla's autopilot work? Many users are not particularly concerned [5]. In military applications, commanders often need to know the reasons for AI system decision-making, because of the problems of life and death, war ethics and so on, before they can make up their minds. But AI cannot explain its thoughts and actions to commanders, and commanders cannot fully understand the decision-making process of AI system and cannot distinguish the logic behind a specific action of AI system, which often leads commanders to distrust the decision-making and action made by intelligent system. Therefore, credibility is the key to the wide application of AI system in battlefield.

3 Solution Analysis

In view of the above problems in the application of AI in military, through the analysis of the research status at home and aboard, especially the USA, solutions to the above problems are suggested as follows.

3.1 Using War Games to Simulate, Generate and Collect Data to Solve Data Source Problems

Game AI has developed rapidly in recent years. AI has defeated humans in most games, including Atari, StarCraft, Dota and so on. The rapid development of intelligent game technology has attracted the attention of the USA and taken a series of related measures. At present, the US Marine Corps University has been experimenting with the "Athena" deduction tool, a simulation and deduction platform dedicated to training, education and testing future AI applications [6]. Using Athena's war game, an air assault task can be planned to control the interception position to support amphibious landing. The game requires the user to complete the planning process when talking to the voice assistant. The program will prompt defensive forms, definitions of different tactical tasks and related historical cases. As the game progresses, an AI application captures data, compares the user's use of cover and range crossing areas, then evaluates the user's performance based on other data and records these data into a database on the combat methods of US military professionals. Finally, Athena evaluates the data and provides constructive suggestions for users, comparing their performance with that of top-level users.

Through this game environment, the USA seeks competent commanders and obtains the data needed to build future AI applications. Once enough data is available, Athena can simulate modern military operations and propose new tactics.

Athena provides a test platform for exploring how to integrate AI into military decision-making process. Using war games to observe how military experts make decisions, benchmark data is established for testing a series of applications that enhance warfighter judgment.

3.2 Explore a New Learning Algorithm to Reduce the Dataset Needed for Model Training and Solve the Problem of Data Annotation

In order to reduce the cost and time associated with training and adjusting machine learning models, DARPA launched the project of Learning with Less Labels (LwLL) [7]. Through the LwLL project, DARPA will study new learning algorithms to greatly reduce the dataset for model training or updating. This project aims to make the process of training machine learning model more efficient. DARPA said that through the LwLL project, the dataset needed to build a model will be reduced by six orders of magnitude or more, and the label precedents used to label the model will be adjusted from millions to hundreds; that is, a million images are needed to train a system now, and only one image or about 100 label precedents will be needed in the future.

To achieve this goal, the LwLL project will explore two technical areas. The first technology area focuses on building efficient learning and adaptive learning algorithms. The main approaches are innovation in meta-learning, transfer learning, active learning, K-shot learning and supervised/unsupervised adaptation. The second technical area explores the normative description of machine learning problems, including the difficulty of decision-making and the real complexity of the data used to make decisions. In this regard, DARPA said that it is difficult to assess the efficiency of building machine learning systems or the basic limitations of the accuracy level of the model. Through LwLL project, we hope to find the theoretical limit of machine learning possibilities and use this theory to promote system development and enhance capabilities.

3.3 Enhancing AI System's Autonomous Learning and Environmental Adaptability by Bionics

At present, AI takes machine learning technology as its core and relies on huge data support. But the real world is full of contingency, and programmers cannot predict all possible factors or situations. When these machine learning systems encounter special situations that are not included in programs and databases, they will be at a loss. If we want to expand the capability of machine learning system in this new environment, we must stop its service and retrain it with additional data, but this method relies on human intervention and is inefficient. In contrast, biological systems can conduct independent training, learn from past experience and adapt to the accumulated knowledge even in the face of a new environment. From here we can see that even the most advanced AI systems at this stage are still far from adaptive biological intelligence. In 2017, DARPA launched a project called Lifelong Learning Machines (L2M) [8].

The learning mechanism proposed by L2M does not need to provide the system with a large number of knowledge sets about applications as it does now. It needs limited domain knowledge and methods to start its behavior and learning. L2M tries to apply biological learning mechanism to machine learning system, breaking the dependence of existing machine learning system on pre-programming and training samples, so that artificial intelligence system, like biological system, can make decisions based on experience. Even in the new situation without experience and training before, it can also make scientific decisions quickly according to "experience" accumulated in "life" and improve its operation. Active autonomy enhances the ability to adapt to the environment. The L2M project will use the principles of computer science and biological learning to build learning paradigms and evolutionary networks, and continue learning through external data and internal goals. At the same time, L2M project will set the limit of system capability and build a security mechanism for AI system.

3.4 Setting Up Machine Commonsense Research to Enhance Reasoning Ability of AI System

Common sense is a basic ability of human beings to perceive and understand the world. Typical AI systems lack a general understanding of how the physical world works, a basic understanding of human motivation and behavior, and a general understanding of things like adults. The lack of common sense hinders the AI system's understanding of the world, its natural communication with humans, its rational behavior in unpredictable situations and its learning of new experiences.

At present, in all machine learning applications, machine reasoning is relatively narrow and highly specialized. Researchers must train and program AI systems for each situation, but commonsense reasoning is still missing. To this end, in 2018, DARPA launched a project called Machine Common Sense (MCS) [9]. The MCS project will explore the latest developments in cognitive understanding, natural language processing, in-depth learning and other areas of artificial intelligence research to find answers to commonsense questions.

The MCS project plans to adopt two strategies to develop two different kinds of commonsense services and to design a special evaluation method for the two services. The technical area of research includes the following three aspects: First is the foundation of human common sense, learning experience to build computational models that can simulate human perception of objects (intuitive physics), agents (intentional actors) and locations (space navigation). These computational models will assess and identify cognitive development milestones in developmental psychology research and the literature to determine their learning effectiveness at three levels (prediction/expectation, empirical learning and problem solving). The second is the basic experimental environment of human common sense. The basic test environment of human common sense will be used when testing the research results in the field of technology, i.e., the foundation of human common sense. Third, there is a wide range of common sense. Learn from online reading to construct answers to natural language questions about commonsense phenomena and image-based questions. The service will simulate the general knowledge of American adults in 2018 and will be judged through the benchmark test of the Allen Institute of Artificial Intelligence (AI2). DARPA predicts that researchers will use manual construction, information extraction, machine learning, crowd sourcing and other computational methods to build the knowledge base.

3.5 Developing Cyber Resilience to Actively Address the Security Problem of AI System

In October 2016, the USA released two strategic documents related to the development of artificial intelligence, the National Strategic Plan for Research and Development of Artificial Intelligence and Preparing for the Future of Artificial Intelligence. In the area of AI security, the National AI Research and Development Strategic Plan of the USA emphasizes the new challenges that traditional network security may bring in AI environment [10]. It is considered that AI systems embedded in key systems must be durable and secure to cope with large-scale deliberate network attacks. It is believed that the application of AI system will bring some new and unique threats. At the same time, the report also emphasizes that AI systems may eventually adopt "cyclic self-improvement"; that is, a large number of software modifications will be made by the software itself, rather than by human programmers. In order to ensure the security of self-modifying systems, additional research is needed to develop self-monitoring frameworks such as checking the consistency of the system's behavior through the original goal of the designer or proven value frameworks that can resist self-modifying.

In the Report "Summer Study on Autonomy" released by the Defense Science Board of the United States Department of Defense, the answer to the cyber security problem of intelligent military applications is to enhance the cyber resilience of intelligent systems [4].

Cyber resilience can effectively deal with the security problems of AI system because the focus of cyber resilience shifts from preventing cyber attacks to minimizing the impact of cyber attacks. It does not seek to establish a perfect system to deal with all kinds of incidents, including anti-attack. Instead, it adopts a series of technical means to improve the system's self-active protection ability, so that the key tasks of the system can still be carried out normally in the "toxic carrier" environment, under attack or even under partial control, and maintain the normal function of the system, so as to take the next step. The measures lay the foundation [3].

In January 2013, the US Department of Defense released the report "Resilient Military System and Advanced Cyber Threat," which proposed the construction of resilient military system. In August 2014, the US Air Force issued a "Cyber Resilience" research guide, which calls for accelerating the research of resilience mechanism and key technologies, survival and recovery of core mission functions, cyber deception, cyber agility, resilience and agility of embedded systems [11].

3.6 Revealing the Transparency of AI Decision-Making and Enhancing Credibility by Improving the Interpretability of AI System

Interpretability of AI systems has become a key obstacle to human–computer cooperation. Explainable Artificial Intelligence (especially explainable machine learning) will be indispensable if we want human beings to understand, trust and effectively manage the new generation of AI systems. To this end, in October 2016, DARPA launched the Explainable Artificial Intelligence (XAI) project [12]. The goal is to establish new or improved machine learning technology, generate explainable models and integrate effective interpretation technology to enable end users to understand, trust and effectively manage future AI systems. Through this project, the new machine learning system will be able to explain its own logic principles, describe its own advantages and disadvantages, and explain future behavior.

Explainable AI faces three challenges: how to generate explainable models, how to design interpretive interfaces and how to understand users' psychological needs for effective interpretation.

In response to the first challenge, the project will develop a series of new or improved machine learning technologies to generate more interpretable models; in response to the second challenge, it hopes to integrate the latest human–computer interaction technologies (such as visualization, language understanding, language generation and session management) with new principles, strategies and technologies for effective interpretation; in response to the third challenge, the project will plan to summarize, expand and apply current psychological theories of interpretation.

At present, the project has achieved some results. At the IJCAI/ECAI 2018 Workshop on Explainable Artificial Intelligence (XAI) conference held in July 2018, several papers funded by the project were published, including depth adaptive programming through reward decomposition interpretation, natural language interpretation of visual question and answer using scene diagrams and visual attention generation, finite state representation of learning cycle strategy network, etc.

4 Conclusion

Artificial intelligence technology has shown good application prospects in all walks of life and has become the core and key to achieve leap-forward development of every industry. Military field is no exception. According to the characteristics of military field, four key problems of AI application in military are analyzed and summarized, including data problem, autonomous learning problem, security problem and credibility problem. For each of these problems, through the analysis and summary of domestic and foreign research, the corresponding solutions are introduced, which can provide reference to related research.

References

1. Jin X (2017) Status and development of intelligent command and control. Command Inf Syst Technol 8(4):10–18
2. Vanian J (2018) Artificial intelligence is too easy to be corrupted by bad examples, what should we do? http://www.fortunechina.com/business/c/2018–07/12/content_311635.htm. Cited 6 May 2019
3. Zhou GX (2017) US military cyber deterrence and cyber resilience. Command Inf Syst Technol 8(5):76–80
4. Defense Science Board (2016) Summer study on autonomy, Washington
5. Mclemore C, Lauzen H (2018) The dawn of artificial intelligence in naval warfare. https://nosi.org/2018/06/16/the-dawn-of-artificial-intelligence-in-naval-warfare/?shared=email&msg=fail. Cited 6 May 2019
6. Jensen B, Cuomo S, Whyte C, Wargaming with ATHENA: how to make militaries smarter, faster, and more efficient with artificial intelligence. http://www.warontherocks.com Cited 6 May 2019
7. DARPA (2018) Learning with Less Labels (LwLL), Arlington, Virginia
8. DARPA (2017) Lifelong Learning Machines (L2M), Arlington, Virginia
9. DARPA (2018) Machine Common Sense (MCS), Arlington, Virginia
10. Wang YP (2017) Information security in artificial intelligence strategy. Secrecy Sci Technol 11:27–30
11. Lan YS, Zhou GX, Wang H, Yi K (2015) Construction mechanism and implementation of resilient command information systems. J Command Control 1(3):284–291
12. DARPA (2016) Explainable Artificial Intelligence (XAI), Arlington, Virginia

Recent Advances for Smart Air Traffic Management: An Overview

Chao Jiang(✉)

The 28th Research Institute of China Electronics Technology Group Corporation,
Nanjing 210000, China
capjiangchao@163.com

Abstract. With the development of the civil aviation industry, the pressure carried by the air traffic management (ATM) system, which is the core of civil aviation operations, is also gradually increasing, and therefore, it is necessary to use the power of various emerging information technologies to build a smart air traffic management system, thereby improving operational efficiency while ensuring safe operation of air traffic management. In recent years, the research of smart air traffic management has become a hot topic in academic circles, supported by industry guidance units such as the Civil Aviation Administration and the Air Traffic Management Bureau, and scholars at home and abroad have conducted relevant explorations on smart air traffic management. The paper comprehensively analyzes the research results of smart air traffic management in recent years, and for cloud computing, big data, artificial intelligence, Internet of Things (IoT), mobile Internet, five most important emerging information technologies, the research progress of combining with smart air traffic management is analyzed, meanwhile, proposed the shortcomings of the existing research and the suggested research directions for the next step, provide reference for further research on smart air traffic management.

Keywords: Smart ATM · Cloud computing · Big data · Artificial intelligence · IoT · Mobile Internet

1 Introduction

Since 2009, US President Barack Obama has proposed the concept of "Smart Earth" for the first time at a round table with business leaders. The concept of "smart" has taken root in all walks of life and has become more and more popular with the continuous development of new generation of information technology, the civil aviation industry is no exception, put forward the development concept of "smart civil aviation". Taking it as an extension, in 2018, the Civil Aviation Administration of China put forward the "four enhancements of air traffic management" strategic objectives of "enhancing security, enhancing efficiency, enhancing smartness, and enhancing coordination," which focus on making up shortcomings and improving capabilities from aspects such as production safety, guaranteed operation, and construction developing [1]. Among them, "enhancing smartness" is an important foundation and necessary path to achieve the objectives of "enhancing security, enhancing

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 415–421, 2020.
https://doi.org/10.1007/978-981-15-0187-6_49

efficiency, and enhancing coordination." And because of the importance of smart air traffic management for the future development of air traffic management system, at home and abroad, a lot of research has been proposed for this field, based on recent research results in the field of smart air traffic management at home and abroad, the paper will comb the concept and connotation and application and expansion of smart air traffic management and discuss the development trend and priority of future research in the field of smart air traffic management.

2 Concept and Connotation of Smart ATM

With the development of the new generation of information technology and the continuous expansion of the concept of "smart," the concept of smart air traffic management came into being. Wenxian and Xinping [2] proposed the concept of smart air traffic management earlier, they believe that the smart air traffic management system is a novel air traffic management system fully incorporating new generation of information technologies, in which sensors are equipped and universally connected, forming the "Internet of things for ATM" and contributing to the real-time surveillance and control for air traffic. The heterogeneous information from "Internet of things for ATM" is fused and processed through advanced computer platform. Then, the processed information is provided to every domain application platform, fulfilling more precise and more dynamic air traffic management. The concept basically applies the relevant fusion concept in the smart earth and depicts the basic logical framework of the smart air traffic management.

Jidong [3] strengthens the importance of Internet and artificial intelligence technology in the connotation of smart air traffic management, and he believes that in the smart air traffic management system, smart refers to the use of Internet technology to give management equipment some abilities of artificial intelligence, so that it can make its own judgment quickly and accurately, with certain mechanical smartness. This connotation also highlights an important feature of the smart air traffic management, which is to improve the production efficiency and diversity of assistant decision-making of air traffic controller through technical means [4] and reduce the amount of labor for them.

Hongyu et al. [5] proposed a concept of smart air traffic management and designed the overall framework of smart air traffic management system including sensing layer, network layer, platform layer, application layer, and visualization layer (Fig. 1). The sensing layer obtains information, the network layer transmits information, and the platform layer implements information storage, information exchange, information processing, etc., and the application layer covers control command, airspace management, air traffic flow management, flight service, low-altitude/general aviation, unmanned aircraft air traffic management, etc., and the visual layer provides efficient and intelligent interaction. Flexible and scalable basic platform and air traffic management applications can be realized through layered principles.

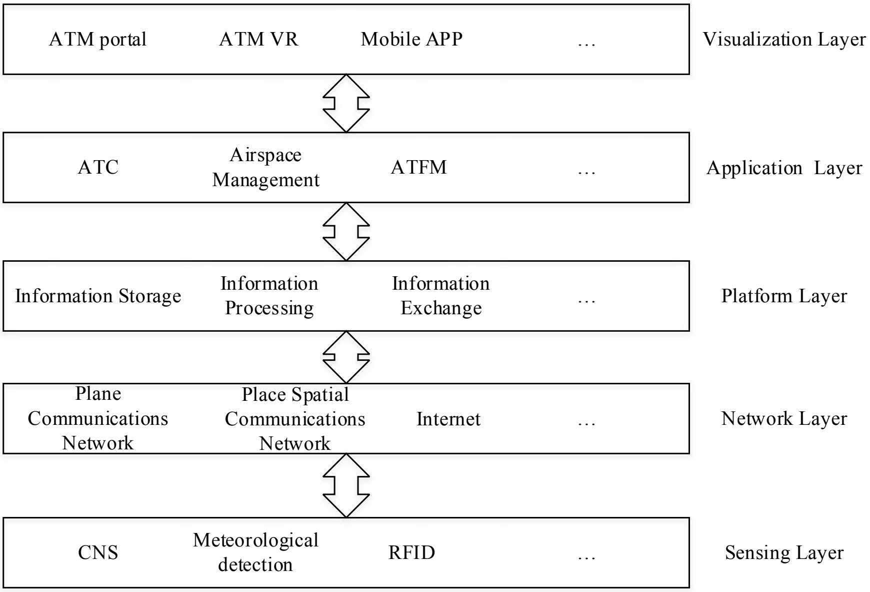


Fig. 1. Architecture of intellectualized air traffic management

3 Application and Expansion of Smart ATM

The development and implementation of smart air traffic management are inseparable from the maturity and application of a new generation of information technology. Scholars at home and abroad have conducted the research and exploration which combined air traffic management with representative technologies of a new generation of information technologies such as cloud computing, big data, artificial intelligence, Internet of Things, and mobile Internet.

3.1 Cloud Computing

Yun [6] analyzed four types of shortcomings of the current situation of data storage, processing, and analysis in air traffic management, and discussed the potential applications and advantages of using cloud computing in the management and processing of air traffic management data, and accordingly proposed improved optimization solution based on cloud computing infrastructure. She thought that data management solutions based on cloud computing can greatly improve the reliability and security of information systems in air traffic management and reduce the complexity and repeatability of data management.

Yunfei et al. [7] applied cloud computing to civil aviation collaborative decision-making system, they designed the system algorithms based on cloud computing topology, and carried out comparative algorithms analysis. Then, the optimal algorithm was used to complete the preliminary overall design and simulation test of the cloud computing platform of system, which proved the feasibility and advancement of using cloud computing in air traffic management information systems.

Yuan [8] summarized the application experience of using cloud computing technology in the construction of air traffic management information systems at home and abroad. Analyses in the new generation of air traffic management information systems architecture based on cloud computing technology, air traffic management cloud computing verification project, air traffic flow management system based on cloud computing were focused on. And by setting up an air traffic flow management prototype system based on cloud computing technology, the feasibility of constructing an air traffic management system in the form of cloud was verified.

3.2 Big Data

With the development of information technology, big data technology has been widely used in the field of air traffic management as a trend of the current information industry. Chen and Fan [9] have combed the research progress of big data using the USA's air traffic management which has more developed civil aviation industry. Currently, the USA has achieved certain results in the System Wide Information Management (SWIM), the FAA operation and performance data system, the US National Flight Data Center, the enhanced traffic management system (ETMS), and other aspects. Overall, a big data application management model in air traffic management for statistical analysis, collaborative operation, and planning decision has been formed. The storage and hierarchical management of related data are becoming more and more standardized, and the secondary development and efficient use of data have become a hierarchy that is worth learning from.

Jingwei et al. [10] discussed the construction path of air traffic management big data based on information fusion. They proposed the basic idea of constructing China's air traffic management big data to build a national core database, establish an interactive interaction mechanism, and provide analytical and evaluation data products, by fully integrating and mining the data to optimize China's air traffic management information systems.

Ziqi [11] believed that the current application of big data in the field of air traffic safety management is still in its infancy, but it has achieved certain applications and breakthroughs in promoting the transformation of safety management mode, safety hazard investigation, operational risk warning, and unsafe incident investigation and case study.

Yongjie and Hui [12] integrated the previous research results and proposed a four-layer architecture of air traffic management big data based on the data, technology, platform, and application from the perspective of the application of the air traffic management network information system. Meanwhile, in terms of coordination, security, efficiency, and management, the application direction and development trend of air traffic management big data were given.

3.3 Artificial Intelligence

Xiaohong et al. [13] proposed the composition and implementation method of the artificial intelligence assistant decision-making system for air traffic management from the reality. The system aimed at solving different problems of air traffic management at all levels based on the prediction of flight conflict, the resolution of flight conflict, and the intelligent management of flight flow so as to finally realize the full application of artificial intelligence in air traffic management.

Yabin [14] believed that artificial intelligence technology can be applied to air traffic flow management assistant decision-making and flight conflict detection and detachment assistant decision-making, thus forming an artificial intelligence assistant system for air traffic management.

Tiansheng [15] believed that Automatic Dependent Surveillance Broadcast (ADS-B) can be regarded as a special artificial intelligence technology. Compared with other technologies, ADS-B provides a more accurate supervision method, enabling air traffic controllers can control air traffic management information systems as a whole and enhance the safety of air traffic, effectively solve congestion and re-push problems, improve the air traffic operating environment, and ensure flight safety.

3.4 Internet of Things (IoT)

Xinping [16] proposed a new air traffic management system based on IoT technology. The core of the project is to use the Internet of Things to improve the perception ability of air traffic in a more efficient and intelligent way and to enhance the interaction ability between air traffic controllers and pilots through various data links, thereby improving the level of air traffic management.

Yudong [17] analyzed the application prospects of using IoT technology in air traffic management collaborative decision-making system, airborne interval assistant system, and flight queue management.

Rui [18] and Yuanwei [19] proposed that the two key technologies of smart air traffic management are radio frequency identification (RFID) technology and wireless sensors network, by use of the two key technologies to realize surface operations, airspace operations, collaborative air traffic flow, runway safety, and equipment status management.

3.5 Mobile Internet

As a representative application software of the era of mobile Internet, WeChat has attracted applications from all walks of life since its introduction of Mini Program, due to its high feasibility of technology and application. Duan and Xie [1] developed the “Smart Management of Air Traffic Management Projects” system with the help of WeChat Mini Program tool. The system can realize the effects of mobilization, informationization, intelligence, velocity, and visualization of the project management system of East China Air Traffic Management Bureau of CAAC. It can be said that the use of mobile Internet has improved the intelligent level of air traffic management.

4 Suggestions for Follow-up Research

Based on the results of previous research, it can be seen that scholars at home and abroad have made useful explorations of smart air traffic management from the perspective of the combination of air traffic management and a new generation of information technology. Cloud computing, big data, artificial intelligence, Internet of Things, and mobile Internet technologies all have obtained a relatively rich application, but there are still some shortcomings in the existing research.

In the aspect of combining the research of smart air traffic management with cloud computing and big data technology, the existing research focuses on solving the storage optimization problem of large amounts of data, and the business application is a little limited. As the potential value of the air traffic management data continues to increase, it is recommended to carry out in-depth research of intelligent data processing of air traffic management. By studying the technologies of acquisition, processing, transmission, interaction, and intelligent mining of various air traffic management data, the capability promotion of the value of air traffic management data to the operation of air traffic management is further enhanced.

In the aspect of combining the research of smart air traffic management with artificial intelligence technology, the existing research has proposed to apply it to air traffic flow management assistant decision-making and flight conflict detection and detachment assistant decision-making. It also proposed the idea of developing auxiliary decision-making system. It is suggested that the follow-up study can further expand the scope of application on this basis and carry out more exploration in intelligent conflict management, intelligent air traffic flow management [20], intelligent planning management, and intelligent entry and exit sorting, and intelligent airport operation. At the same time, there have been many runway incursions in recent years. The hidden problem of information deviation in ground–air interaction needs to be solved urgently. For the important role of ground–air interaction talks in air traffic management, artificial intelligence technology can be introduced to study the applications of some important operation process such as voice read-back of air traffic controllers. What is more worthy of expectation is that, until the application of future artificial intelligence and other technologies in air traffic management is mature and reliable, we can completely use the robots with the ability of air traffic management, planning, reasoning, and action to complete the air traffic management business. It will bring about tremendous changes in the development of the entire air traffic management.

5 Conclusion

The importance of smart air traffic management for the future development of air traffic management has reached a general consensus. Researches have also been carried out in this field, and rich research results have been obtained. Combining the research results in the field of smart air traffic management at home and abroad, the paper combs the concept and connotation of smart air traffic management and analyzes the application expansion of smart air traffic management for five most important emerging information technologies (cloud computing, big data, artificial intelligence, Internet of

Things, and mobile Internet). And the development trend and priority of future research on smart air traffic management are discussed in view of the shortcomings of the existing research, which provides a reference for the further research of the subsequent smart air traffic management.

References

1. Changmiao D, Shijia X (2018) The development of smart ATC engineering system based on WeChat Mini Program. Civ Aviat Manag 335(09):41–44
2. Wenxian S, Xinping Z (2013) Smart air traffic management system and its application. J Nanjing Univ Aeronaut Astronaut (Soc Sci) 15(3):51–55
3. Jidong B (2014) Smart air traffic management system and its application. Sci Technol Inf 12 (20):22
4. Tak-Kuen John K, Yi M, George JP et al (1997) Smart ATMS: a simulator for air traffic management systems. In: Proceedings of the 1997 winter simulation conference
5. Hongyu Y, Bo Y, Xiping W et al (2018) Research and prospect of intellectualized air traffic management technology. Adv Eng Sci 50(04):16–25
6. Yun L (2014) Application prospect of cloud computing in data processing of air traffic management. Comput Knowl Technol 10(12):2704–2706
7. Yunfei Y, Xiaojian C, Weiqing C et al (2015) Research on infrastructure of civil aviation collaborative decision making system based on cloud computing. Softw Ind Eng 4:36–41
8. Yuan S (2018) Research on architecture of ATM information system and verification of air traffic flow management system based on cloud computing technology. Telecom World 341 (10):46–47
9. Chen Z, Fan L (2017) Research of the big data applications of the American ATM. Innov Sci Technol 208(6):94–96
10. Jingwei T, Weijun P, Zhilin L et al (2018) Research on the construction of air traffic management big data based on information fusion. Innov Sci Technol 18(3):65–69
11. Ziqi L (2018) Talking about the application of big data in the safety management of ATM. PC Fan 2018(18):57
12. Yongjie Y, Hui D (2019) Big data value embodiment of air traffic management in business operation. Command Inf Syst Technol 10(01):7–12
13. Xiaohong L, Cunru B, Jiong W et al (2007) Application of artificial intelligence technique in air traffic management. Air Traffic Manag 12:25–27
14. Yabin C (2013) Application of artificial intelligence technique in ATM. Silicon Valley 2013 (7):118
15. Tiansheng F (2018) Practice of artificial intelligence in ATM (ADSB). Sci Technol Innov 2018(11):79–80
16. Xinping Z (2013) A novel air traffic management system based on the Internet of Things. J Transp Inf Saf 31(5):145–149
17. Yudong W (2014) Exploration of new air traffic management system based on Internet of Things. Sci Technol Vis 2014(31):73
18. Rui W (2017) Smart ATM system and its applications. Technol Innov Appl 2017(18):270
19. Yuanwei L (2017) Smart ATM system and its applications. Dual Use Technol Prod 6(2):15
20. Ping C (2016) Global air traffic management (ATM) operational concept and its enlightenment. Command Inf Syst Technol 7(6):13–16

Application of GRIB Data for 4D Trajectory Prediction

Yibing Yin(✉) and Ming Tong

State Key Laboratory of Air Traffic Management System and Technology,
Nanjing 210016, China
yinyibing1992@163.com

Abstract. The accuracy of 4D trajectory prediction is the foundation of trajectory-based operation and decision support tools of ATC. Meteorological modeling, especially the wind modeling, is the key element for improving the precision of trajectory prediction. Through analyzing the wind data in GRIB (GRIdded Binary) format, wind speed and direction at each levels can be obtained. Meanwhile, the 4D trajectory prediction model is constructed based on the wind effect to aircraft movements, especially ground speed and heading of aircraft. Taking arrival flights to Shanghai Pudong International Airport as an example, a simulation is conducted through the comparison of prediction results and the radar information. And the simulation results indicate the effectiveness of the proposed approach.

Keywords: Trajectory prediction · GRIB data · Wind data · Meteorological modeling

1 Introduction

The aircraft 4D track is an accurate description of the spatial position (longitude, latitude, and altitude) and time of each point in an aircraft track in space and time [1]. The accuracy prediction of 4D trajectory is the technical basis of a new generation of air traffic control automation system based on track operation. It is widely used in flight sorting [2], early warning and cancelation of flight conflicts [3], and track optimization.

At present, there are two main methods for aircraft flight trajectory predicting: First, based on the hybrid estimation theory [4] to achieve track prediction, after all, aircraft motion not only has multi-modal properties, but also needs to consider its three-dimensional state in the horizontal, vertical and lateral directions. Therefore, track prediction can be regarded as an estimation problem of stochastic linear hybrid systems [5]. The second is based on aircraft dynamics and kinematics models, using various performance parameters of the aircraft to achieve track prediction [6].

For aircraft dynamics and kinematics based on track prediction studies, accurate data support can improve the effectiveness of predictions, and meteorological data is one kind of important data. The influence of wind data in meteorological data on track prediction is particularly important. If wind speed and direction are lacking, the accuracy of aircraft four-dimensional trajectory prediction will be reduced. However, due to the difficulty in obtaining wind data, in the current domestic 4D track prediction

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 422–430, 2020.
https://doi.org/10.1007/978-981-15-0187-6_50

research, the influence of wind on track prediction is theoretically described, or the wind speed and wind direction are assumed in the form of vector diagrams. Simple modeling: Since the above method cannot accurately obtain the real data of the meteorological department, the effect of the four-dimensional track prediction is not convincing. Therefore, it is particularly important to model the track prediction wind information and to use the meteorological model covering the wind information as an independent module to participate in the construction of the track prediction model. By analyzing the meteorological data in GRIB format, this paper realizes the storage of the specific wind data in the form of database, and then in the process of track prediction, real-time reading and calculation can effectively improve the accuracy of 4D track prediction.

2 Format and Processing of Wind Data

2.1 Wind Data Format

The European Center for Medium Range Weather Forecasts (ECMWF) provides meteorological information on a global scale, and its numerical forecast is currently at a high level in the world and has a high use value. The meteorological data provided by ECMWF can be selected according to the isobaric surface, that is, the numerical information under different levels is found. The meteorological information is stored in the file in the GRIB data type, and the forecasting center updates the data every 6 h.

The GRIB code [7] is a computer-independent compressed binary code used primarily to represent numerical weather analysis and product forecast data. For high-speed communication links using modern communication protocols, GRIB is an effective tool for delivering large volumes of gridded data. By packaging and compressing various related data into GRIB codes, the information is organized much more compactly than the character-based form, thus facilitating the storage of data and speeding up the transfer between computers.

2.2 Wind Data Processing

GRIB is an effective tool for a large number of gridded data. The global positioning of meteorological information is determined by grid points, and the grid position is uniquely represented by latitude and longitude. The process of obtaining wind data in a certain area has the following four steps:

(1) Select the required weather data. Select the data of the upper air pressure layer, including the date, time (0, 6, 12, 18 h), wind component, required pressure layer, etc., where the time is UTC time.

(2) Crop the area. Select the meteorological data of a certain area of the world, and determine the north–south area by the latitude range, and determine the east–west area by the longitude range.

(3) Determine the unit grid span. The data is defined at 0.75° × 0.75° latitude and longitude spans, you can also customize the grid span to the required accuracy.

(4) After each option is determined, download the file. The file download format is GRIB data. The GRIB data file is recorded in units. Each record contains the value of a single physical quantity of a set of grid points on a height layer.

For the wind data stored in the GRIB format on each acquired air pressure layer, the description file can only be generated after being decoded. The basic process is: First, use the tool grib2ctl. exe generates a description file.ctl of the entire file. Then, use the tool gribmap. exe generates a map file. idx; Finally, through its description file, the form and structure information of the file storage record can be extracted to process the data.

The specific process is shown in Fig. 1.

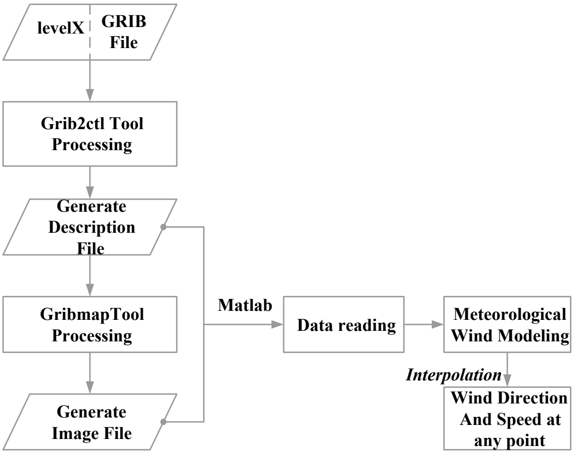


Fig. 1. Flow chart of GRIB wind data processing

3 Meteorological Modeling of Trajectory Prediction

3.1 Track Prediction

Track prediction is a process of estimating the future trajectory of an aircraft by a track predictor. The track predictor predicts the future direction of the aircraft based on the current state of the aircraft, estimated flight intent, expected environmental conditions, and aircraft performance, such as Fig. 2 shows.

Preparation process: The initial version of the flight script (aircraft flight intent) [8]. Data input requires flight planning, airline operating procedures, and air traffic control restrictions on aircraft performance.

Calculation process: The core function of the track predictor. The flight script contains a description of the aircraft's operation, including its programmed air traffic control restrictions. The track calculator integrates intent information with 4D trajectory implementation using environmental condition prediction information and aircraft performance data.

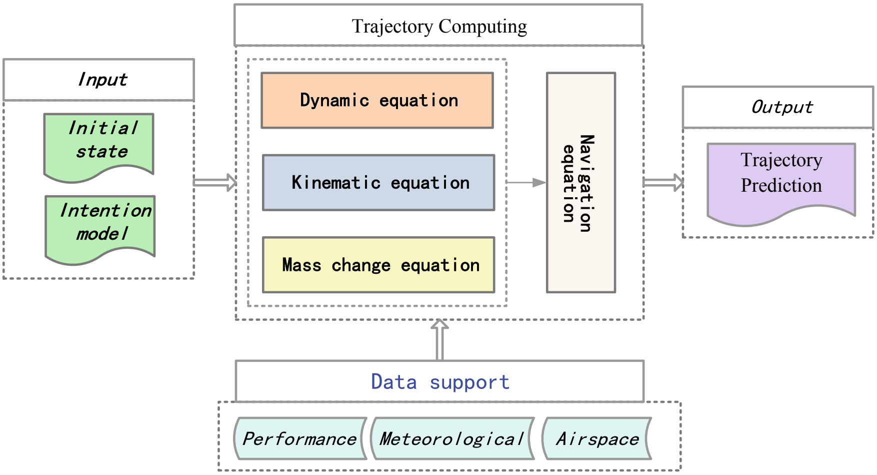


Fig. 2. Prediction process of 4D trajectory prediction

Update process: The track predictor can update the process based on time when new status or intent data appears. The update process may generate new flight scripts or may only correct the information.

Output process: The predicted track data is handed over to the customer for the application, such as collision detection and release, approach management, and departure management.

The inputs to the track predictor include basic data, meteorological data, and aircraft performance data. Among them: the basic data includes airspace data, route information, etc.; meteorological data is constructed by wind data based on GRIB format given in this paper; aircraft performance data is set by BADA [9] (engine basic data, which is developed by European Control Experimental Center) Supported by ASCII data, including data on operational performance parameters and airline program parameters for 399 aircraft types.

3.2 Wind Data Pre-processing

In the track prediction, the real-time position of the aircraft is not exactly at the grid point. Moreover, the grid formed by the GRIB data points is not a regular rectangular grid, but is an approximately trapezoidal grid. Therefore, in order to determine the wind vector information on the latitude and longitude of the aircraft, it is necessary to find the four grid points adjacent to the latitude and longitude, as shown in Fig. 3 shows.

The east–west wind component $u(M)$ of the M point wind is obtained by two-point linear interpolation:

$$u(M) = u_1 + (u_2 - u_1) \times \frac{(x_0 - x_{i+1})}{(x_i - x_{i+1})} \tag{1}$$

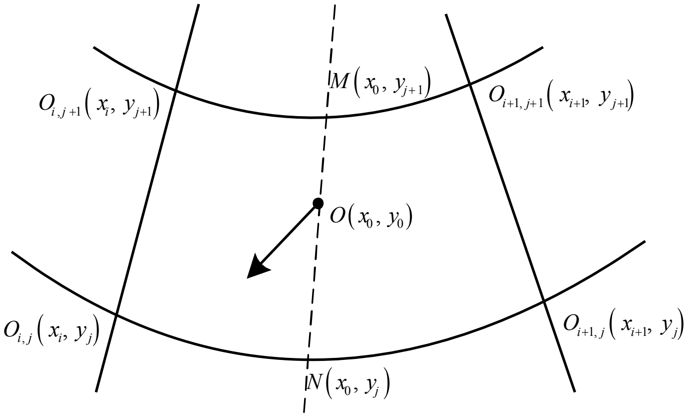

$O_{i,j+1}(x_i, y_{j+1})$
$M(x_0, y_{j+1})$
$O_{i+1,j+1}(x_{i+1}, y_{j+1})$
$O(x_0, y_0)$
$O_{i,j}(x_i, y_j)$
$O_{i+1,j}(x_{i+1}, y_j)$
$N(x_0, y_j)$

Fig. 3. Grid data of trajectory point

In the formula, u_1 is the east–west wind component of point $O_{i+1,j+1}$, and u_2 is the east–west wind component of point $\mathrm{O}_{i,j+1}$.

In the same way, by linear interpolation, the wind component v_1 of the north–south direction of $O_{i+1,j+1}$, the east–west wind component u_2 of the point wind of $\mathrm{O}_{i,j+1,}$ and the wind component v_2 of the north–south direction can be obtained. And the east–west component (u) of the point O and the north–south direction component (v) are obtained finally. Thereby, get the size and direction of the wind:

$$V_{\text{wind}} = \sqrt{u^2 + v^2} \tag{2}$$

$$\varphi_{\text{wind}} = \begin{cases} \arctan(u/v) & u > 0, v > 0 \\ 180 - \arctan(u/-v) & u > 0, v < 0 \\ 180 + \arctan(-u/-v) & u < 0, v < 0 \\ 360 - \arctan(-u/v) & u < 0, v > 0 \end{cases} \tag{3}$$

Among them, V_{wind} is the wind speed and φ_{wind} is the wind direction. North–south direction: the south wind is positive; East–west direction: the west wind is positive.

3.3 Wind Impact Analysis

In the track prediction, when the vacuum speed and the wind speed are constant, the degree of crosswind, that is the difference of the wind angle, will cause the magnitude of the bias flow and the ground speed to change [10].

Due to the influence of the crosswind, the aircraft's track will be biased toward the leeward side of the heading, resulting in a bias flow. The bias flow is the angle between the direction line of the aircraft relative to the air movement to the direction line of the ground motion, and is represented by the angle between the ground speed vector and the space velocity vector. In the left crosswind, the track is biased to the right side of the heading, and the bias flow is positive. When the crosswind is right, the track is biased to the left side of the heading and the bias flow is negative; and the magnitude of the drift is related to the vacuum speed and the wind speed and the degree of crosswind.

The angle between the ground speed vector and the wind speed vector, that is, the wind angle, indicates the extent to which the aircraft is subjected to the crosswind. Taking the flight path as the reference, when the left-side wind is used, the flight path is clockwise to the wind direction line, which is positive value, i.e. "+WA"; when the right-side wind is used, the flight line is counterclockwise to the wind direction line, which is negative value, i.e. "−WA."

From the decomposition of the wind, we can know that: $V_{\text{wind}} \times \sin(\text{WA})$ is the crosswind component, which will make the flight path deviate from the heading line and generate a bias flow, which will affect the navigation direction; $V_{\text{wind}} \times \cos(\text{WA})$ is the cis (inverse) wind component, which will affect the ground speed of the aircraft, make the ground speed not equal to the vacuum speed, and affect the sailing time.

In the track prediction, since the wind directly exerts a force on the aircraft, a bias flow and a windward wind component are generated, which affects the passing time of the aircraft. During aircraft flight, according to the known wind direction and wind speed, the bias flow and ground speed can be calculated, and the distance traveled per unit time can be obtained:

$$\begin{array}{l} \text{WA} = \varphi_{\text{wind}} - \text{MTK} \\ \text{DA} = \arcsin(V_{\text{wind}}/V_{\text{TAS}} \times \sin(\text{WA})) \\ V_{\text{GS}} = V_{\text{TAS}} \times \cos(\text{DA}) + V_{\text{wind}} \times \cos(\text{WA}) \\ \text{DIS} = V_{\text{GS}} \times \Delta T \end{array} \tag{4}$$

Among them, DIS is the distance files per unit time, and MTK is the track angle.

4 Simulation Verification

4.1 Radar Data

The radar data from Shanghai terminal airspace is copied, and the radar track is transferred to a computer-readable file (.txt) by controlling the playback function of the automation system.

Extract historical radar data for a particular day's flight, including location and speed information for each flight at different times. According to the route and sector division of the navigation data in Pu Dong Airport, the radar trajectory of the arrival flight from the northwest on January 2, 2013, is analyzed, as shown in Fig. 4 (the thin-dotted line indicates the sector boundary and the thick-dotted line indicates the approach route, the solid line indicates the radar trajectory).

4.2 Wind Vector

The GRIB grid data of the required Shanghai terminal area is tailored and processed according to the flow chart shown in Fig. 1. The wind vector is shown in Fig. 5.

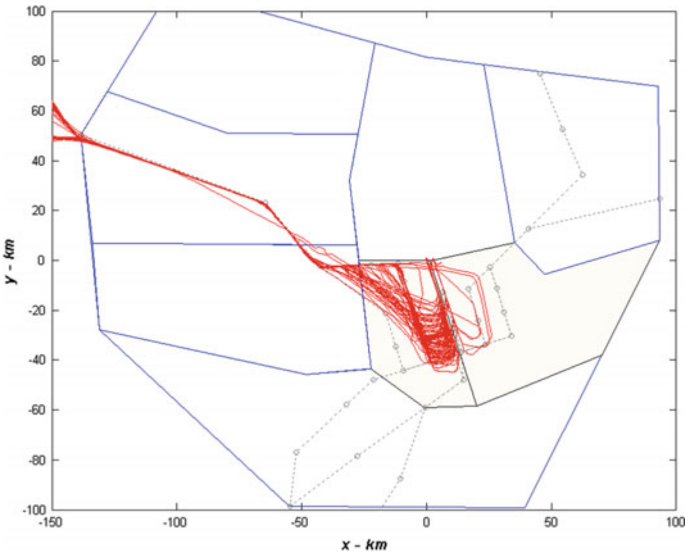


Fig. 4. Radar trajectory of Pu Dong airport arrival flight from the Northwest

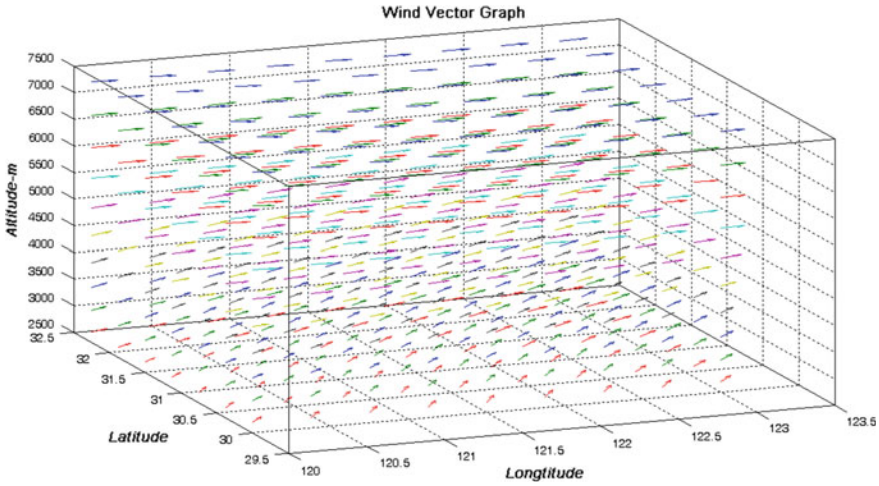


Fig. 5. Schematic of wind vector data

4.3 Simulation Verification

The flight from northwest approaching to Pu Dong Airport (CCA4503) was selected as a simulation case on January 2, 2013. In the forecasting process, the track prediction is performed on the meteorological models that are added and not added by the GRIB wind data. The comparison between the track prediction point and the actual radar point is shown in Table 1.

Table 1. Trajectory prediction analysis of flight CCA4503

Route point	Pass time (without wind data)	Actual arrival time	Estimated arrival time
VMB	7:14:35	7:14:35	7:14:35
EKIMU	7:21:00	7:20:08	7:20:50
JTN	7:24:04	7:22:58	7:23:48
PD019	7:26:11	7:24:59	7:25:54
IAF	7:27:21	7:26:10	7:27:11
PD015	7:28:17	7:26:55	7:27:57
PD014	7:29:39	7:28:10	7:29:19
PD013	7:31:43	7:30:23	7:31:23
PD012	7:33:05	7:32:00	7:33:45
PD023	7:34:33	7:33:56	7:34:25
IF	7:37:17	7:37:03	7:37:07
FAF	7:37:55	7:37:41	7:37:47

After comparative analysis, it can be seen that the track prediction result of adding wind data is more accurate, indicating the integration of wind data in the GRIB data format for meteorological modeling can effectively improve the accuracy of the track prediction tool.

5 Conclusion

In this paper, the processing flow of wind information in GRIB data format is given in the process of track prediction, and the meteorological model supported by wind data is integrated into the track prediction model to carry out four-dimensional track prediction. For more complex meteorological modeling, and the impact of weather forecast information on track prediction, it will be an important research direction in the future.

References

1. Korn B, Helmke H, Kuenz A (2006 July) 4D trajectory management in the extended TMA: coupling AMAN and 4D FMS for optimized approach trajectories. In: Silicon Valley, California: 25th International Congress of the Aeronautical Sciences (ICAS 2006)
2. Schroeder JA (2009) A perspective on NASA Ames air traffic management research. AIAA, 2009-7054
3. Ruiz S, Piera M, Pozo I (2013) A medium term conflict detection and resolution system for terminal maneuvering area based on spatial data and 4D trajectories. Transp Res Part C 26:396–417
4. Hwang I (2003) Air traffic surveillance and control using hybrid estimation and protocol—based conflict resolution. Stanford University, USA
5. Liu W, Hwang I (2011) Probabilistic 4D trajectory prediction and conflict detection for air traffic control. In: IEEE conference on decision & control

6. Fukuda Y, Shirakawa M, Senoguchi A (2010) Development of trajectory prediction model. In: ENRI international workshop on ATM/CNS (EIWAC), Tokyo, Japan
7. Hagelauer P, Mora-Camino F (1998) A soft dynamic programming approach for on-line aircraft 4D-trajectory optimization. Eur J Oper Res 107(1):87–95
8. Michael AK, Scott D, Robab S et al (2009) Improving ground-based trajectory prediction through communication of aircraft intent. AIAA, 2009-6080
9. Eurocontrol Experimental Center. User Manual for the Base of Aircraft Data (BADA), Revision 3.9. EEC Technical/Scientific Report No. 11/03, 2011
10. Alejo D, Cobano JA, Heredia G et al (2014) Collision-free 4D trajectory planning in unmanned aerial vehicles for assembly and structure construction. J Intell Rob Syst 73 (1–4):783–795

Weibo Rumor Detection Method Based on User and Content Relationship

Zhongyue Zhou, Haijun Zhang(✉), and Weimin Pan

School of Computer Science and Technology, Xinjiang Normal University,
Urumqi 830054, China
zhjlp@163.com

Abstract. In order to effectively identify the rumor information in the Weibo platform, we propose a combined model based on deep learning, which includes convolutional neural network (CNN) that incorporates the attention mechanism and combines with the neural network of long short-term memory (LSTM) to implement a microblog rumor detection method for the characteristics of user-content relations. Firstly, the convolutional neural network incorporating the attention mechanism is used to extract the fine-grained features of the user-content relationship. Secondly, the LSTM network is used for coarse-grained feature extraction. Finally, the extracted feature vectors are classified by the Softmax classifier so as to achieve a good effect of rumor detection.

Keywords: Weibo · LSTM · CNN · Attention mechanism · Softmax

1 Introduction

With the further development of the Internet era, social media has gradually increased its position in people's life, study, and work. Weibo is a broadcast social network platform that spreads short-text real-time information through interactive methods. As an open platform, Weibo has the characteristics of real-time, convenience, and rapid dissemination in information. However, Weibo is not only an effective platform for people to obtain information, but also an important tool for the breeding and dissemination of rumors.

At present, most of the rumors research based on Weibo are still using traditional machine learning methods. Most of these methods use texts, user behaviors, time series, and emotional values as rumor judgement basis. This paper leads into a framework of deep learning, using a convolutional neural network model with attention mechanism and a time-characteristic LSTM network model to effectively identify the microblog rumors by using the relationship between user features and content features.

The main structure of this paper includes the following parts: The second part introduces the related work on rumor recognition; the third part describes the rumor detection method based on composite model of user-content relations; the fourth part summarizes the proposed method.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 431–434, 2020.
https://doi.org/10.1007/978-981-15-0187-6_51

2 Related Work

At present, there are two common methods for detecting microblog rumors, one is feature classification based on traditional learning, and the other is rumor recognition based on deep learning by constructing neural network model.

The feature classification method is mainly based on content features and user features. Qazvinian et al. [1] used Bayes classifier to construct positive and negative models and extracted various potential features such as behavioral features and emotional features of the information posted or put forwarded by users. Zhang et al. [2] believe that shallow features cannot distinguish between rumor information and nonverbal information. Therefore, an automatic rumor detection method based on the newly proposed implicit feature and the shallow feature of the message is proposed.

Compared with the common feature extraction classification algorithm, the neural network model based on deep learning is more efficient and accurate for rumor recognition. Ruchansky [3] and others proposed a model called CSI, which is based on the improvement of recurrent neural network; the model is mainly divided into two modules, the former captures the time response characteristics of users and content and combines with the text, and the latter captured source features and achieved good results in false news recognition.

The above research work had carried on rumor recognition from multiple dimensions and achieved good rumor recognition effect. However, the performance of monolayer neural network model in feature extraction is not very ideal, and the extraction of some deep features cannot be completed. When the common CNN is combined with the LSTM network model, some user-content relationship characteristics are often neglected, which leads to the loss of some features, and the long-term time series rules need to be optimized. Therefore, this paper proposes a CNN with attention mechanism that combines with the LSTM network model to achieve the purpose of rumor recognition.

3 Rumor Recognition Based on Composite Model of User-Content Relations

3.1 Traditional CNN-LSTM

The convolutional neural network model based on deep learning has the ability of learning characteristics hierarchically. Research [4] shows that the features learned by convolutional neural networks have stronger discriminative ability and generalization ability than artificial design features. The LSTM is a time recurrent neural network [5]. The neural network has three main gates (input gate, forgetting gate, and output gate) and a cell unit component, which can realize time series prediction.

Traditional rumor detection method based on deep learning mainly extracts the local features of short texts through the convolutional layer of CNN and constructs new vectors by convolution kernel operation. On the characteristics of time dimension, using LSTM network model with the ability of feature extraction for long-distance time series, data can solve the characteristics of long-distance context-dependent feature learning and store information of different time periods and construct deep learning model for rumor detection.

3.2 CNN with Attention Mechanism Combined with LSTM

In recent years, neural networks based on the attention mechanism [6] have made important progress in the tasks of natural language processing. We integrate the attention mechanism into the convolutional neural network, mainly by multiplying the output features of the CNN module and the salient features output by the corresponding attention mechanism module. The feasibility and effectiveness of the fusion method have been verified in the literature written by Wang et al. [7].

$$e_{ji} = \alpha(\text{query}_i, \text{key}_j) \tag{1}$$

Compared with the traditional rumor detection method, the combined model learning method based on user-content relationship proposed by us, Which comprehensively considers the relationship between users and content, extracts the relevant features between them, and achieves better rumor identification effect through model training. Inspired by the CNN-based neural multitask model, we use the CNN model with attention mechanism to extract significant fine-grained features and extract coarse-grain features based on text and emotion features. At the same time, each dimension is refined and processed, and the output is carried out through the last hidden layer in LSTM network, which can prevent the memory loss and gradient exploding problems caused by excessive step length. Finally, Softmax classifier is used for feature classification.

Finally, we will introduce the characteristic feature parameters used in the simulation module. The characteristic parameters between the user and the content are given in Table 1.

Table 1. User and content characteristics

Number	Content-based feature	Number	User based
x_1	String length	y_1	Number of followers/fans
x_2	Semantic feature	y_2	Impact/number of tags
x_3	Punctuation/emoji	y_3	Registration time
x_4	Emotional polarity	y_4	Forward/comment
x_5	Positive/negative	y_5	member
x_6	Message relevance	y_6	User credibility
x_7	Label/connection	y_7	Comment
x_8	Graphic combination	y_8	Registration time/location

4 Conclusion

We firstly use the CNN neural network that incorporates the attention mechanism. Compared with the single neural network model, it can better explore the deep user characteristics and extract significant fine-grained user and content relation features. Then, LSTM is used to mine the coarse-grained features of long memory information, so as to realize the purpose of rumor recognition. The method is simulated in the deep learning framework—TensorFlow. The experiment proves the feasibility of the proposed model strategy, and the CNN integrates attention mechanism that combines with LSTM network model and fully exploits the characteristics between the user and the content relationship, which can effectively improve the accuracy and time efficiency of rumor detection.

Acknowledgements. This work is supported by Xinjiang Joint Fund of National Science Fund of China (U1703261).

References

1. Qazvinian V, Rosengren E, Radev DR et al (2011) Rumor has it: identifying misinformation in microblogs. In: Proceedings of the conference on empirical methods in natural language processing, Edinburgh, pp 1589–1599
2. Zhang Q, Zhang S, Dong J et al (2015) Automatic detection of rumor on social network. In: Natural language processing and chinese computing, Springer, Cham, pp 113–122
3. Ruchansky N, Seo S, Liu Y (2017) CSI: a hybrid deep model for fake news detection
4. Zhou C, Sun C, Liu Z et al (2015) A C-LSTM neural network for text classification. arXiv preprint arXiv:1511.08630
5. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8): 1735–1780
6. Abedinia O, Amjady N, Zareipour H (2016) A new feature selection technique for load and price forecast of electrical power systems. IEEE Trans Power Syst 32(1):62–74
7. Wang P, Li W, Gao Z et al (2017) Scene flow to action map: a new representation for RGB-D based action recognition with convolutional neural networks
8. Yang S, Shu K, Wang S et al (2019) Unsupervised fake news detection on social media: a generative approach. In: Proceedings of 33rd AAAI conference on artificial intelligence
9. Gao L, Guo Z, Zhang H et al (2017) Video captioning with attention-based LSTM and semantic consistency. IEEE Trans Multimedia 19(9):2045–2055
10. Kwon S, Cha M, Jung K (2017) Rumor detection over varying time windows. PLoS ONE 12(1):e0168344
11. Ma J, Gao W, Wong KF (2018) Detect rumor and stance jointly by neural multi-task learning. In: Companion of the web conference 2018, International World Wide Web Conferences Steering Committee, pp 585–593

Improved HyperD Routing Query Algorithm Under Different Strategies

Xiaoyu Tao, Bo Ning(✉), Haowen Wan, and Jianfei Long

Dalian Maritime University, Dalian 116026, China
ningbo@dlmu.edu.cn

Abstract. A hypercube labeling model HyperD was introduced to solve the problem of excessive bandwidth consumption and long query time between various nodes. By analyzing and adopting different strategies to improve the routing query algorithm of HyperD, a relatively more efficient routing query algorithm is proposed, which reduces the cost of communication and accelerates the data transmission between nodes. In order to verify the effectiveness of the algorithm, experiments were performed on randomly selected HyperD networks of various sizes. The improved routing algorithm was compared with the known routing algorithms, the results show that the improved algorithm is more efficient, and it can find the optimal solution in valid time.

Keywords: Bandwidth consumption · Routing query algorithm · HyperD

1 Introduction

Reducing unnecessary message transmission and the overall bandwidth utilization in P2P networks is a very interesting issue [1–4]. Toce et al. [5–7] used hypercube in dynamic label mode (HyperD) to prove that the label mode can reduce the bandwidth utilization in the network under certain conditions. Compared with the traditional Dijkstra algorithm [8–10], the HyperDRouter algorithm given by Andi Toce et al. greatly reduces the search time, but it cannot find a better path in a valid time for non-complete hypercube cases. This paper continuously improved the routing query algorithm with different strategies. We, respectively, proposed MNR, MMNR, and MPPR algorithm. Then, we compared and analyzed the three algorithms with the existing Dijkstra algorithm and the HyperDRouter algorithm given. The results showed that the MPPR algorithm was the best overall, especially in the case of a high sparse rate.

2 Routing Query Algorithm

In routing algorithm HyperDRouter, the path between two nodes is determined by the Hamming distance between their labels and the path of any intermediate nodes [8, 11]. The time complexity is close to $O(k^2)$, and k is the dimension. The followings are the improved routing algorithm of HyperD with different strategies, which can find the optimal solution in an effective time.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 435–441, 2020.
https://doi.org/10.1007/978-981-15-0187-6_52

2.1 Maximum Neighbor Priority Routing Algorithm

This section adopts the strategy of preferentially selecting the node with the largest neighbor set and controls the selection range of the node. The improved heuristic maximum neighbor priority routing algorithm (MNR) is proposed. Different from the HyperDRouter algorithm, Eq. (2.1) further controls the range of selected nodes, which makes it more likely to find the best path.

$$\mathrm{Path}_{\mathrm{MNR}} = \sum_{t=u}^{v} \max\{\mathrm{Neis}(p)|p \in \mathrm{All\_min}\{\min\{\mathrm{difference}(l_t, l_v)\}\}\} \tag{2.1}$$

2.2 Multipath Minimum Routing Algorithm

The multipath minimum routing algorithm (MMNR) uses the enumeration search strategy to find the path from the starting point to the destination node by finding all the nodes that meet the conditions, and finally finds the optimal path through statistics. The path formula is shown in Eq. 2.2.

$$\mathrm{Path}_{\mathrm{MMNR}} = \min(\sum_{t=u}^{v} \mathrm{All\_max}\{\mathrm{Neis}(p)|p \in \mathrm{All\_min}\{\min\{\mathrm{difference}(l_t,l_v)\}\}\}). \tag{2.2}$$

Algorithm 1 selects all nodes that satisfy the cyclic iteration condition (lines 7–11). When the target node is found (lines 3–5), the path is stored in the path of statistical path set. If the path is divided into m_1 paths, the time complexity is $O(m_1 \times k^2)$.

```
Algorithm 1    MMNR (D,u,v,Current_Path)
Input：HyperD network D, source node u ∈ $V_D$, goal node v ∈ $V_D$, Current_Path
Output：min{Paths} // The shortest path among all paths from u to v, Paths ⊆ $E_D$.
1: if   Current_Path is Null
2: Init Paths ⇐ Ø    // Paths is a collection of paths, global variables.
3:   if   v ∈ neighborhood (u) {
4:           Current_Path ⇐ Current_Path ∪ {edge(u,v) }
5:           Paths ⇐ Paths ∪ Current_Path
6:   } else
7:Select all node t ∈ neighborhood(u) such that q ∈ max{ Neis(p)| p ∈ all_min(distance(t,v))}. arr(t) add q.  //arr(t) is the current array of t that satisfies the condition q.
8: for each arr(t)
9:     Current_Path ⇐ Current_Path ∪ {edge(u,t) }
10:    MMNR(D,t,v,Current_Path)
11: Return   Paths
```

2.3 Multipath Maximum Probability Path Algorithm

In this section, the multipath maximum probability path algorithm (MPPR) is proposed. Each node finds a label that satisfies the topK condition to cycle, which is basically very close to the optimal path. The time complexity of the algorithm is $O(m_2 \times (\log_2 n)^2)$ (m_2 is the number of branches, and n is the number of labels included in HyperD). The path formula is described by Eq. (2.3).

$$\text{Path}_{\text{MPPR}} = \min\left(\sum_{t=\text{u}}^{v} \text{topK}\left(\frac{\text{Neis}(t)}{\min(\text{difference}(l_t, l_\text{v}))}\right)\right) \cdot \left(e_t = \sum_{i=1}^{\text{topK}} p_i \geq 0.9\right) \tag{2.3}$$

```
Algorithm 2  MPPR (D,u,v,Current_Path,flag)
Input: HyperD network D, source node u ∈ $V_D$, goal node v ∈ $V_D$, Current_Path
Output: Shortest path from u to v in Min_Path, Min_Path ⊆ $E_D$
if  Current_Path is Null{
    Init  H=distance(u,v), P(t)= $\frac{Neis(t)}{distance(t,v)}$ , Min_Path ⇐ Ø  }
if  v ∈ neighborhood (u)  {
        Current_Path ⇐ Current_Path ∪ {edge(u,v) }
        Min_Path ⇐ Current_Path
} else if (flag = =0){
      Select all node t ∈ neighborhood(u) such that meet top2(P(t)), t ∈ arr(t)  }
  else { Select all node t ∈ neighborhood(u) such that meet top4(P(t))-top2(P(t)),
t ∈ arr(t) }  // top2 is all the first two nodes that satisfy the highest P(t)ratio
    for each arr(t)    //arr(t) is the array of the currently selected t
    Current_Path ⇐ Current_Path ∪ {edge(u,t) }
    if  (Min_Path=Ø ∪Min_Path.Length > Current_Path.Length+1)< H ∩
        Current_Path.Length < H
    MPPR(D,t,v,Current_Path,0)
    end for
if  Min_path= Ø ∩ Current_Path has repeated points {
    Back Current_Path to the first node(q) will be repeated.  //backtrack to the
first repeat point of the current path.
    Current_Path=Current_Path(u,q).    // Intercept time segment of the current
path to the node q .
    MPPR(D, q, v, Current_Path, 1)}
    Return  Min_Path
```

Algorithm 2 selects a set of neighbors (line 8) that satisfy top 2 at each step, and recursion is performed on all the nodes that satisfy the criteria (lines 11–14). $P(t)$ is a set of maximum values. In extreme cases (line 16), the algorithm has been in an infinite loop and cannot find a suitable path. Backtracking to the starting node that caused the infinite loop, the nodes other than the selected node are to be re-filtered once (line 10).

3 Experimental Analysis

This experiment mainly uses sparse rate, fault tolerance, etc., as indicators to compare and analyze the results of several algorithms.

$$\text{Sparse Rate} : R_{\text{sparse}} = \frac{\text{Node}_{\text{Hypercube}} - \text{Node}_{\text{HyperD}}}{\text{Node}_{\text{Hypercube}}} \times 100\% \tag{3.1}$$

$$\text{Fault Tolerance Rate} : R_{\text{error}} = \frac{\text{Path}_{\text{current}} - \text{Path}_{\text{best}}}{\text{Path}_{\text{best}}} \times 100\% \tag{3.2}$$

As shown in Fig. 1, each algorithm has a certain degree of optimization compared to the HyperDRouter. MPPR is further optimized for MNR and MMNR, almost identical to the Dijkstra. As shown in Fig. 2, the time requirements of the HyperDRouter, MNR, and MMNR are almost the same, MPPR is only 5–10 times, and Dijkstra is hundreds or even thousands of times.

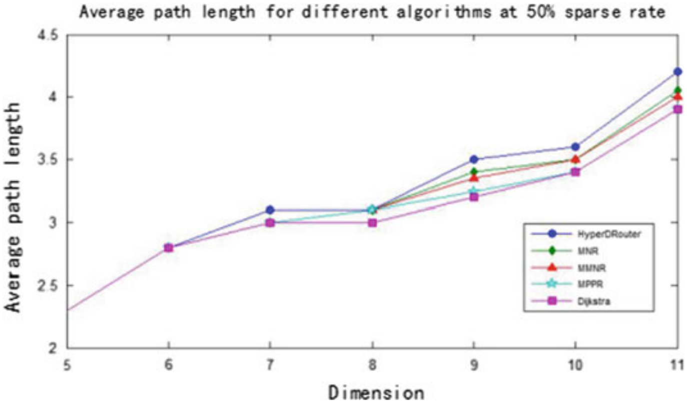


Fig. 1. Dimension change corresponding to average path length

As shown in Fig. 3, compared with the HyperDRouter, the path lengths of MNR and MMNR have a smaller reduction, the MMNR is relatively smaller, and both MPPR and Dijkstra have a larger reduction and are similar. As the sparse rate increases, these gaps gradually increase and then decrease. Figure 4 shows that at the case of lower

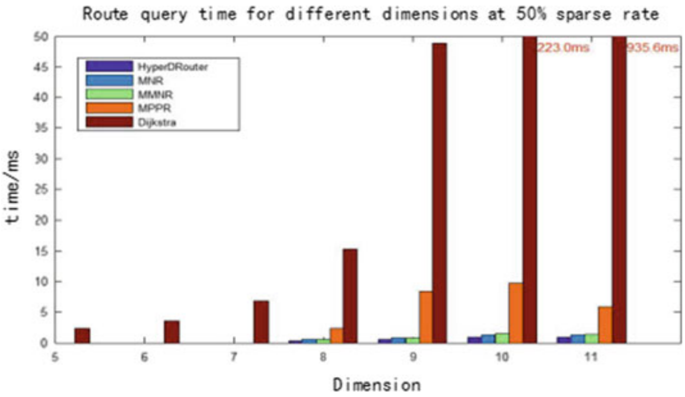


Fig. 2. Dimension change corresponding query time

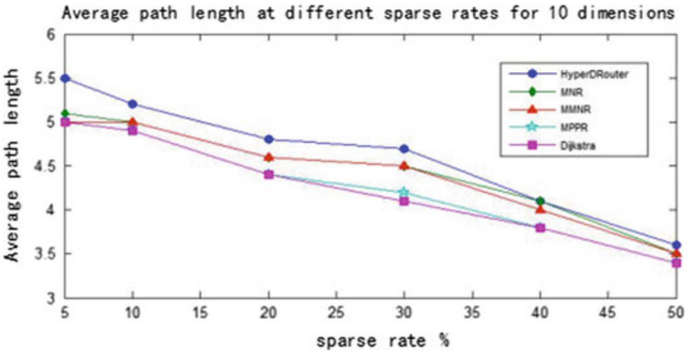


Fig. 3. Sparse rate change corresponding to average path

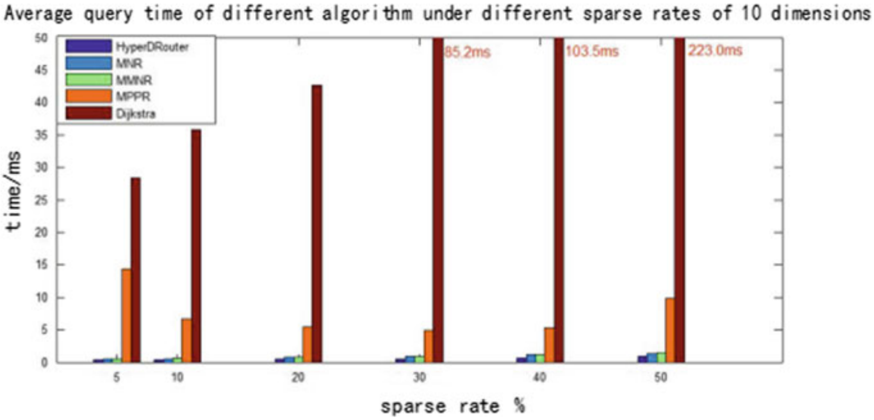


Fig. 4. Query time required for the change of sparse rate

sparse rate, the time requirements of the MPPR and Dijkstra algorithms are not much different. However, when the sparse rate is high, the MPPR algorithm is better, and the search time required is significantly less than the time required by the Dijkstra.

Figures 5 and 6 can be seen that the MPPR is the closest to the Dijkstra as a whole. The MNR and MMNR also have some improvement on the path length. Simultaneously, compared to HyperDRouter, MPPR has reduced fault tolerance by nearly 10 percentage points, which is a big improvement.

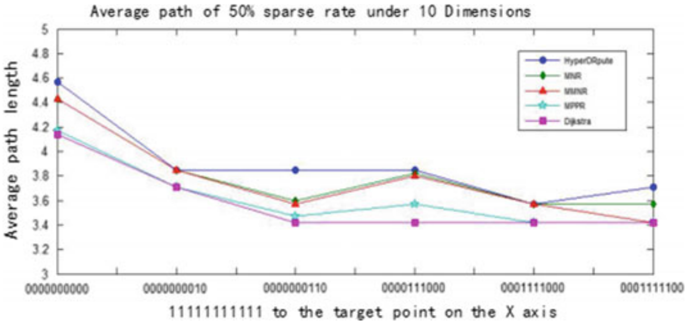


Fig. 5. Corresponding path length of different destination nodes

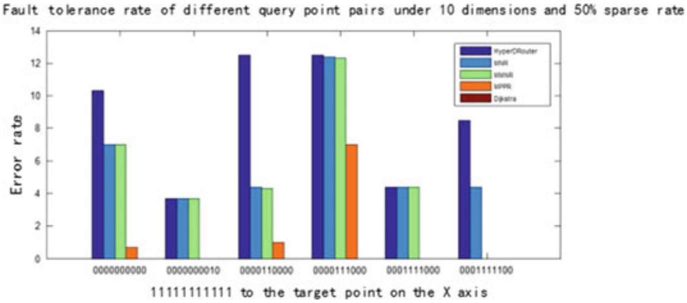


Fig. 6. Corresponding fault tolerance rate of different destination nodes

4 Conclusion

We have continuously improved the routing query algorithm of HyperD under different strategies. Through a series of experiments for comparative analysis, the improved routing algorithm is more effective. It ensures that the optimal solution can be found within a certain time range.

References

1. Autry CW et al (2011) The new supply chain agenda: a synopsis and directions for future research. Int J Phys Distrib & Logist Manag 41(10):940–955
2. Smoot ME et al (2011) Cytoscape 2.8: new features for data integration and network visualization. Bioinformatics 27(3):431–432
3. Chuang PJ et al (2007) Hypercube-based data gathering in wireless sensor networks. J Inf Sci Eng 23(4):1155–1170
4. Balakrishnan H et al (2003) Looking up data in P2P systems. Commun ACM 46(2):43–48
5. Toce A et al (2011) HyperD: a hypercube topology for dynamic distributed federated databases. In: The fifth annual conference of the international technology alliance, UMD, College Park, Maryland, USA. https://www.researchgate.net/publication/245535766
6. Toce A et al (2012) HyperD: analysis and performance evaluation of a distributed hypercube database. In: Proceedings of the sixth annual conference of ITA, Maryland, USA
7. Toce A et al (2017) An efficient hypercube labeling schema for dynamic peer-to-peer network. Parallel Distrib Comput 102:186–198
8. Broumi S et al (2017) Application of Dijkstra algorithm for solving interval valued neutrosophic shortest path problem. In: Computational intelligence
9. Chen XB (2009) Some results on topological properties of folded hypercubes information. Process Lett 109(8):395–399
10. Qiao Y, Yang W (2017) Edge disjoint paths in hypercubes and folded hypercubes with conditional faults. Appl Math Comput 294:96–101
11. Broumi S et al (2016) Shortest path problem under bipolar neutrosphic setting. Appl Mech Mater 859:59–66

Computer-Aided Diagnosis of Mild Cognitive Impairment Based on SVM

Sichong Chen, Wenjing Jiang, Bingjia Liu, and Zhiqiong Wang(✉)

College of Medicine and Biological Information Engineering,
Northeastern University, Shenyang, China
{20175672, 20175664, 20175656}@stu.neu.edu.cn,
wangzq@bmie.neu.edu.cn

Abstract. Recent years, resting-state functional magnetic resonance imaging (rs-fMRI) and complex network analysis are guiding new directions in studies of brain disease. Here, this study aims to discriminate mild cognitive impairment (MCI) from cognitive normal (CN) by computer-aided diagnosis. For each subject, the functional brain network was constructed by thresholding with sparsity. Features were extracted through network measures and selected via the least absolute shrinkage and selection operator (LASSO). Features of two forms, computed separately under each threshold and integrated into area under the curve (AUC), were extracted. With features in the form of AUC, the support vector machines (SVM) classifier achieved an accuracy of 87%, which raised to 90% with threshold of 0.3. LASSO without stability selection was examined, achieving similar performance in classification to LASSO with stability selection while being less time consuming.

Keywords: Computer-aided diagnosis · Mild cognitive impairment · Functional brain network · Feature extraction and selection · Support vector machines

1 Introduction

Mild cognitive impairment (MCI) is a kind of disease which precedes Alzheimer's disease (AD) and develops widely in elderly people. Research [1] has shown that MCI patients are of higher risk converting into AD than normal aging people. Due to the unclear etiology and pathogenesis of AD, there is no effective treatment for AD at present. However, for patients with MCI, drug intervention and other methods are available for effectively delaying and even curing the disease [2]. Therefore, exploring an accurate-aided diagnosis method for MCI is the key to curing MCI and reducing the conversion from MCI to AD.

The human brain a complex network, which continuously processing and transporting information among different regions, not only structurally, but functionally [3]. Therefore, the connectivity could be explored by utilizing complex

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 442–451, 2020.
https://doi.org/10.1007/978-981-15-0187-6_53

network analysis, which is important for analyzing the pathogenesis and prevention of brain diseases. Researchers have recognized the use of complex network analysis and machine learning in the diagnosis of cognitive impairment. Khazaee et al. [4] proved that abundant original features and effective feature selection are both conducive to obtaining the optimal feature set in the classification of MCI, AD and healthy subjects. In the research on the identification of whether MCI is converted to AD or not by Ye et al. [5], the method of stability selection with sparse logistic regression was introduced for feature selection.

This study aimed to develop a complex network-based machine learning method of accurate-aided diagnosis for MCI in order to reduce the conversion from MCI to AD. This study adapted the advantages of the previous study and was carried out in the following procedures. Pearson's partial correlation and thresholding by defining sparsity were used to construct the functional brain network. From brain networks, features in two different forms were extracted, one was computed separately under each threshold and the other was integrated into AUC. The optimal feature set was selected by LASSO, and the effectiveness of it when combined with stability selection was explored. Finally, SVM with leave-one-out cross-validation (LOOCV) was used to construct and evaluate the classifier.

2 Materials and Methods

The materials and the overall procedure of this research is shown in Fig. 1.

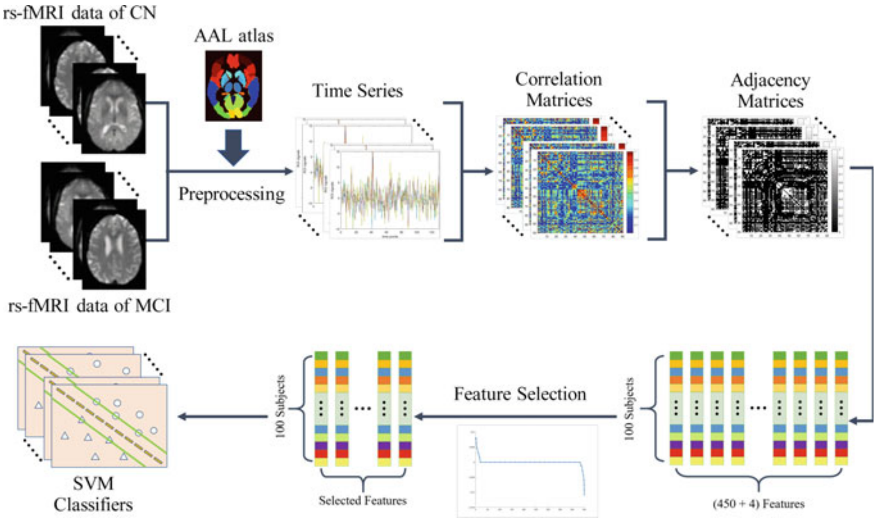


Fig. 1. Overall procedure of this study.

2.1 Brain Network Construction

Data acquisition and preprocessing: Data of rs-fMRI using three tesla (3T) scanners were selected from Alzheimer's Disease Neuroimaging Initiative (ADNI, http://adni.loni.usc.edu/). The rs-fMRI data of each subject were acquired with the following parameters: repetition time (TR) = 3000 ms; echo time (TE) = 30 ms; flip angle = 80°; slice thickness = 3.312 mm; and 48 slices. By utilizing the packages of the DPABI [6] and the SPM12 [7], the preprocessing was carried out, including slice timing, realign, denoising, normalization and smoothing.

Construction of functional brain network: The procedure of functional brain network construction is shown in Fig. 2. For each subject, the time series of 90 regions of interest (ROIs) defined by the automated anatomical labeling atlas [8] (AAL) were extracted from the preprocessed re-fMRI data. Each ROI was considered as one node of the network, and the edge of two nodes was defined by the Pearson's partial correlation coefficient between time series of the two ROIs. Then, the partial correlation coefficient matrix was constructed. In thresholding, the matrix was first transformed into absolute value and eliminated the diagonal elements, then converted into a binary adjacency matrix as the functional brain network. The threshold was determined by defining sparsity value of the network, and the optimum threshold would be determined depending on classification results.

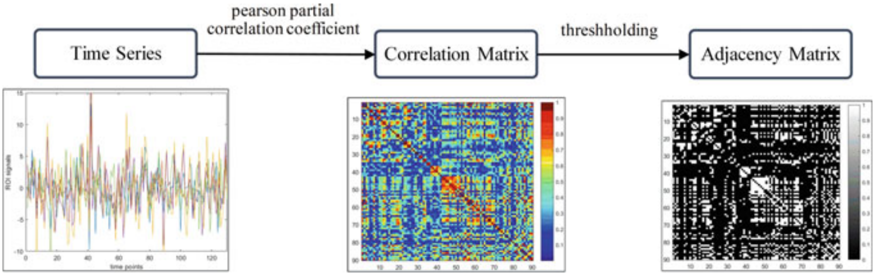


Fig. 2. Procedure of the functional brain network construction.

2.2 Feature Extraction and Selection

In order to reduce the complexity as well as increase the effectiveness of data, feature extraction based on network measures and feature selection via LASSO were conducted. Feature extraction gives a relatively low dimensionality representation of input data by computation based on some typical measures, and thus, the original feature set is generated. For further dimensionality reduction, feature selection is needed to generate optimal feature subset from the original feature set.

Feature extraction based on network measures: Adjacency matrices of each subject constructed under various thresholds are different, resulting in the difference of the network features. Therefore, it is necessary to find a comprehensive measurement of all thresholds. In this study, except for using features from each threshold, AUC of each series feature was calculated to improve the generalization ability of features. Totally, nine network measures (including both local and global measures) were used to generate features from adjacency matrix. For all subjects, each local network measures had 90 features (represent the value in 90 brain areas), and each global network measures had one feature. Network measures employed are:

(1) Betweenness centrality [9]
Betweenness of node i characterizes its effect on information flow between other nodes:

$$\mathrm{BC}_i = \sum_{s \neq i \neq t} \frac{n_{st}^i}{g_{st}} , \tag{1}$$

where g_{st} is the number of all shortest path of node s and t. n_{st}^i is the number of paths containing node i in the g_{st} paths.

(2) Degree centrality
The degree of a node means the number of edges that the node connects to other nodes, which could be normalized to the degree centrality:

$$\mathrm{DC}_i = \frac{D_i}{N-1} , \tag{2}$$

where D_i is the degree of node i, N is total number of nodes, which equals 90.

(3) Clustering coefficient [10]
In local measure, clustering coefficient of node i suggests the closeness of node i with the neighbor nodes:

$$\mathrm{CC}_i = \frac{2R_i}{k_i(k_i-1)} , \tag{3}$$

where CC_i is the sum of edges of sub-network composed by node i and its k_i neighbors. When applied to the entire network, clustering coefficient is the mean value of the local clustering coefficient over all nodes.

(4) Efficiency [11]
In local measure, efficiency of node i quantifies how efficient the communication is among its neighbors when it is removed. Formally:

$$E_i = \frac{1}{N_{G_i}(N_{G_i}-1)} \sum_{i \neq k \in N_{G_i}} \frac{1}{d_{jk}} , \tag{4}$$

where G_i is sub-network of the neighbors of node i. N_{G_i} is the number of nodes within the sub-network d_{jk} is the shortest path between node j and k. On a global scale, efficiency quantifies the communication efficiency among all nodes.

(5) PageRank centrality [12]

PageRank centrality was originally used to measure the value of a website. This feature can measure the importance of a node from a probabilistic perspective. Formally:

$$\mathrm{PR}_i = \frac{1-\alpha}{N} + \alpha \sum_{j \in N_{G_i}} \frac{\mathrm{PR}_i}{D_j^{\mathrm{out}}} , \tag{5}$$

where α is a constant which equals to 0.85 and D_j^{out} is the out-degree of node j.

(6) Assortativity [13]

Assortativity is a global network feature which calculates the Pearson correlation coefficient of all nodes in the network:

$$A = \frac{\left(\sum_{i \in N} D_i\right)\left(2\sum_{i \neq j \in N} D_i D_j\right) - \left(\sum_{i \in N} D_i^2\right)^2}{\left(\sum_{i \in N} D_i\right)\left(\sum_{i \in N} D_i^3\right) - \left(\sum_{i \in N} D_i^2\right)^2} , \tag{6}$$

where D_i is the degree of node i. In addition, the z-score of assortativity was calculated in order to eliminate the individual random differences of brain networks.

Feature Selection via LASSO: In this study, the feature selection algorithm based on the sparse linear regression with stability selection was adopted to select the feature subset of high discrimination. This study utilized SLEP [14] tool package based on MATLAB for sparse linear regression. LASSO is an effective model for sparse linear regression, through which the weight values of many original features will become zero, and the preserved features with nonzero weight will be the effective ones. The principle of stability selection is to conduct repeated random sampling N times and select the features in each sample. This study divided the original feature set into several feature subsets, according to the property of features of the functional brain network. Then combined those subsets according to the binomial coefficient. In each subset combination, employed LASSO with stability selection (by bootstrapping), and selected the features whose selection frequencies in stability selection are larger than a given threshold *thd* as the optimal features in one combination. In this process, the appropriate value of the regularization parameter λ were selected from a sequence $\{\lambda_1, \lambda_1, \lambda_M\}$. Finally, the feature selection frequency in all combinations was computed as the selection score, and features with selection score higher than *thd* were selected into the final optimal feature subset.

2.3 Classification

Support vector machine (SVM) is a supervised learning model based on strict mathematical logic which is appropriate for classifying small samples with high accuracy. In order to discriminate MCI from CN, SVM combined with LOOCV method was used in this study to construct the classification model with features in the form of different thresholds and in form of AUC. The toolbox of LIBSVM provided by Prof. Lin [15] was utilized.

3 Results and Discussion

3.1 Subjects

In this study, rs-fMRI data of 47 CN subjects and 53 MCI subjects were analyzed (Table 1). The data were acquired from the database of ADNI, and all subjects were professionally diagnosed [16].

Table 1. Basic information of subjects.

	CN	MCI
Number	47	53
Male/female	19/28	24/29
Age (mean ± STD) years	75.74 ± 4.62	72.11 ± 4.22
MMSE[a] score (mean ± STD)	29.18 ± 1.19	26.98 ± 1.86
CDR[b] score (mean ± STD)	0.06 ± 0.14	0.52 ± 0.27

[a] *MMSE* Mini-Mental State Examination

[b] *CDR* Clinical Dementia Rating

3.2 The Small-World Property of Funtional Brain Networks

In the process of construction of functional brain network, the partial correlation coefficient matrix was transformed into a binary adjacency matrix to represent functional brain network of each subject. It is noteworthy that although the brain of each subject is unique, setting different thresholds would result in different adjacency matrices. In this research, to ensure the small-world property is within the appropriate range (sigma $\geq$ 1.1), the range of threshold varied from 0.12 to 0.4 with a step size of 0.02. Entirely, 15 thresholds were considered in this research, so there were 15 adjacency matrices for each subject.

Considering the functional brain network is a kind of small-world network, the small-world property (sigma) was employed to analyze the difference between the brain of MCI and CN. Figure 3 shows that if the threshold lower, the sigma value would become higher, both in networks of CN and MCI. Moreover, at any threshold within this range, the average sigma of the CN is larger than MCI. It suggests that compared to MCI, the brain of CN is more effective in information transmission at a lower wiring cost.

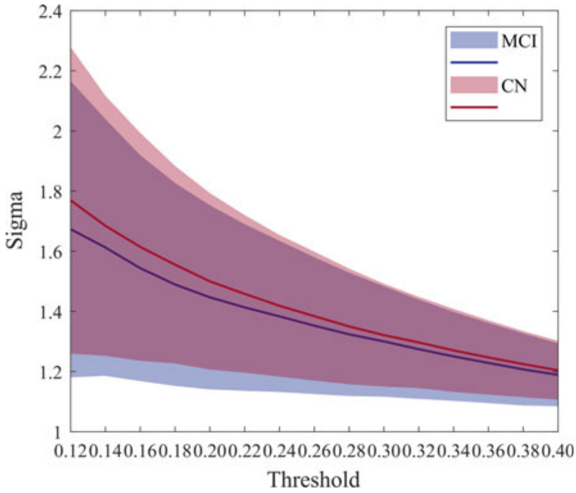


Fig. 3. Comparison of sigma between two groups in different thresholds. The shading represents the distribution of the sigma of all subjects at each threshold. The line indicates the mean value.

3.3 Feature Selection Results

Feature selection was conducted separately in local and global feature sets. The local feature set was divided into five subsets according to different measures to determine the importance of each feature in all subset combinations. As shown in Fig. 4, among all local features selected, the majority was from the measure of betweenness centrality, which indicates that betweenness centrality is a major determinant in the discrimination of CN and MCI. For global features, the z-score of assortativity was the selected one.

3.4 The Classification Results

The effect of two feature forms on classification: This study extracted two different forms of features: features in different threshold values and features in AUC. It can be seen in Fig. 6 that the classification accuracy reached the peak level (90%) at the threshold value of 0.3 and reached lowest level (75%) at the threshold value of 0.22. The accuracy with features in AUC was 86%, which is higher than most of accuracy values with features in threshold. It is noteworthy that the number of selected features maintained high synchronicity with the change of accuracy with features in the threshold, especially when the threshold was within intervals of [0.12, 0.2] and [0.24, 0.38]. In conclusion, classification accuracy could reach an optimal level by choosing an appropriate threshold value, and the number of features has relevance to it. However, the form of features in AUC showed better performance in classification than the form of features in threshold in most cases.

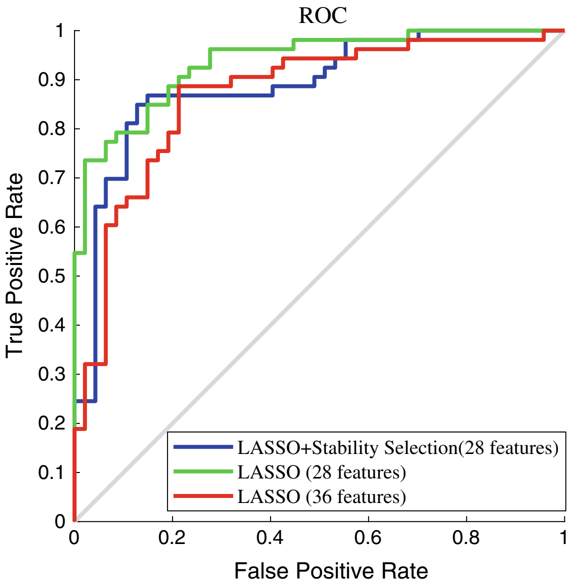


Fig. 4. Selection scores of features from five local network measures. In x-axis, each 90 values represent a measure. They are betweenness centrality, degree centrality, local clustering coefficient, local efficiency and PageRank centrality (from left to right). Features with selection score higher than *thd* were selected. This study set $N = 100$, $thd = 0.5$, and the scope of λ from 0.01 to 0.2 with step length of 0.005 ($M = 39$).

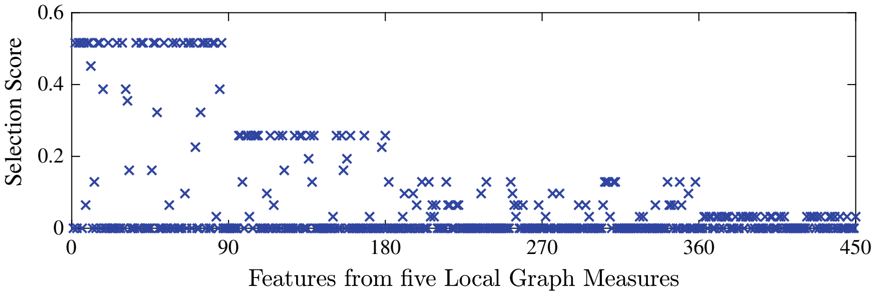


Fig. 5. Receiver operating characteristics (ROC) curves of 3 feature sets selected by different methods.

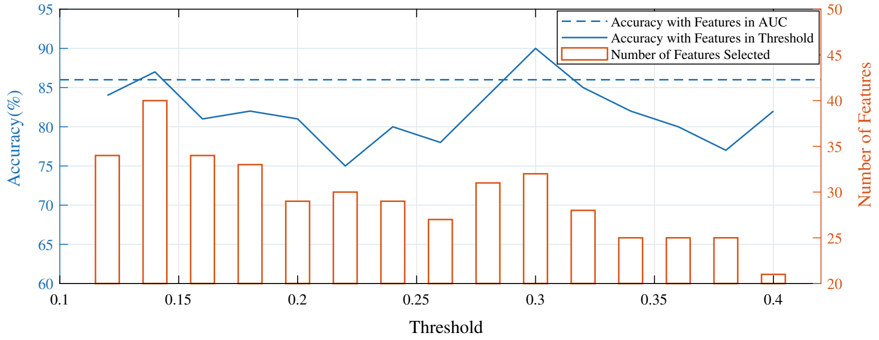


Fig. 6. Comparison between features in threshold and AUC. For each group of features computed under 15 different threshold values, the classifier was constructed and evaluated.

The applicability of stability selection for feature selection: It was demonstrated that logistic regression had a better performance in MCI classification when combined with stability selection [5]. In light of that, whether stability selection would improve the performance of LASSO in feature selection with the data of this study was examined. The area under the ROC curve and the accuracy of the three curves shown in Fig. 5 are: LASSO + Stability Selection (28 features), 0.9755 and 86%; LASSO (28 features), 0.9783 and 87%; and LASSO (36 features), 0.9603 and 84%. The result indicates that the area under the ROC curve when combining LASSO with stability selection for classification was slightly larger than that when using LASSO only (36 features were selected). It is noteworthy that when employing top 28 features with the highest weight value to control the variable of the number of features the same as LASSO with stability selection, using LASSO showed even larger area under the ROC curve. Combining LASSO with stability selection is time consuming for calculation in loops, while could not bring obvious improvement to the classification. Therefore, LASSO has higher applicability in this study than which been combined with stability selection.

4 Conclusion

In this study, based on complex brain networks, a machine learning method was explored to discriminate MCI from CN. By calculating sigma values, it was discovered that the brain of CN is more effective in information transmission than that of MCI. Features in the form of AUC showed better performance in classification than features under most thresholds. LASSO was demonstrated as an effective method in feature selection, with an accuracy of 87.0% and an area under ROC curve of 0.9783 by 28 optimal features in the form of AUC.

5 Ethics and Permissions

Ethical statements and the informed consent from participants are present in the Alzheimer's Disease Neuroimaging Initiative, and we confirmed that all experiments were performed in accordance with relevant guidelines and regulations of access. And we confirmed that all experimental protocols were approved by the College of Medicine and Biological Information Engineering, Northeastern University, China.

Acknowledgements. This work was supported by the Fundamental Research Funds for the Central Universities (N182410001) and National Training Program of Innovation and Entrepreneurship for Undergraduates (201910145101).

References

1. Petersen RC, Negash S (2008) Mild cognitive impairment: an overview. CNS Spectr 13(1):45–53
2. Patterson C (2018) World Alzheimer Report 2018–the state of the art of dementia research: new frontiers. Alzheimer's Disease International (ADI), London, UK
3. Van Den Heuvel MP, Pol HEH (2010) Exploring the brain network: a review on resting-state fMRI functional connectivity. Eur Neuropsychopharmacol 20:519–534
4. Khazaee A, Ebrahimzadeh A, Babajani-Feremi A, Initiative ADN et al (2017) Classification of patients with MCI and AD from healthy controls using directed graph measures of resting-state fMRI. Behav Brain Res 322:339–350
5. Ye J, Farnum M, Yang E, Verbeeck R, Lobanov V, Raghavan N, Novak G, DiBernardo A, Narayan VA (2012) Sparse learning and stability selection for predicting MCI to AD conversion using baseline ADNI data. BMC Neurol 12(1):46
6. Yan CG, Wang XD, Zuo XN, Zang YF (2016) DPABI: data processing & analysis for (resting-state) brain imaging. Neuroinformatics 14(3):339–351
7. Statistical parametric mapping. https://www.fil.ion.ucl.ac.uk/spm/
8. Tzourio-Mazoyer N, Landeau B, Papathanassiou D, Crivello F, Etard O, Delcroix N, Mazoyer B, Joliot M (2002) Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain. Neuroimage 15(1):273–289
9. Kintali S (2008) Betweenness centrality: algorithms and lower bounds. arXiv preprint arXiv:0809.1906
10. Fagiolo G (2007) Clustering in complex directed networks. Phys Rev E 76(2):026107
11. Latora V, Marchiori M (2001) Efficient behavior of small-world networks. Phys Rev Lett 87(19):198701
12. Boldi P, Santini M, Vigna S (2009) PageRank: functional dependencies. ACM Trans Inf Syst 27(4):19-1
13. Newman ME (2002) Assortative mixing in networks. Phys Rev Lett 89(20):208701
14. Liu J, Yuan L, Ye J (2011) SLEP: sparse learning with efficient projections. Technical report, Computer Science Center, Arizona State University
15. Chang CC, Lin CJ (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol (TIST) 2(3):27
16. Aisen PS, Petersen RC, Donohue MC, Gamst A, Raman R, Thomas RG, Walter S, Trojanowski JQ, Shaw LM, Beckett LA et al (2010) Clinical core of the Alzheimer's disease neuroimaging initiative: progress and plans. Alzheimer's Dement 6(3):239–246

Details for Person Re-identification Baseline

Zhong Zhang[1,2](✉), Haijia Zhang[1,2], and Shuang Liu[1,2]

[1] Tianjin Key Laboratory of Wireless Mobile Communications
and Power Transmission, Tianjin Normal University, Tianjin, China
haijia27zhang@gmail.com, shuangliu.tjnu@gmail.com
[2] College of Electronic and Communication Engineering,
Tianjin Normal University, Tianjin, China
zhong.zhang8848@gmail.com

Abstract. In this paper, we evaluate the performance of person re-identification (Re-ID) baseline under different implementation details including resize ratio of the pedestrian image, batch size and basic learning rate. To this end, we employ ResNet-50 as the classification model by modifying the original FC layer and apply a classifier to compute identity prediction values. We perform amounts of experiments to assess the effects of these implementation details on Market-1501 and experimental results show that these implementation details are very important for person Re-ID.

Keywords: Person re-identification · Baseline · Implementation details

1 Introduction

Person re-identification (Re-ID) targets to spot and return an image of pedestrian with the same identity and different camera view from the gallery set. Due to complex variations of pedestrian images in pose, camera conditions and background, hence, person Re-ID is a challenging task. It has important applications in surveillance system [1] and human activity analysis [2], and therefore, both academic and industry pay more and more attention on person Re-ID. Before researchers of the image classification and object recognition fields turn to convolutional neural network (CNN) and deep learning, handcrafted features are used to perform the feature representation [3–6]. Many approaches are proposed for person Re-ID [7–11]. Recently, CNN and deep learning have been widely applied in person Re-ID task, and some large-scale person Re-ID datasets including Market-1501 [12], CUHK03 [13] and DukeMTMC-reID [14] are issued to train deep model. Deep features learned from CNN are robust to variances of pedestrian images, and therefore, CNN-based methods have dominated person Re-ID

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 452–457, 2020.
https://doi.org/10.1007/978-981-15-0187-6_54

community [15–18]. Specifically, these approaches are based on the deep classification model which is treated as the baseline.

In this paper, we perform amounts of experiments to assess influence of different implementation details on the baseline including resize ratio of the pedestrian image, batch size and basic learning rate. We make two contributions as follows:

- We detail the architecture of the baseline to obtain a clear understanding of the classification model.
- We verify various implementation details could have some effects on the performance of the baseline.

We divide the rest of paper into three sections. Firstly, we detail the framework of the baseline in Sect. 2. Afterwards, experimental results conducted on Market-1501 is provided in Sect. 3. Finally, we sum up our work in Sect. 4.

2 Approach

In this section, we utilize the ResNet-50 [19] as the baseline because of its excellent characteristics as well as relatively succinct framework.

2.1 ResNet

The residual network (ResNet) is proposed by He et al. [19], and they participate in the ILSVRC with it. The performance of ResNet in image classification, target detection and other tasks greatly surpasses the performance level of the previous year's competition. The obvious feature of ResNet is that it has a fairly deep depth, from 34 to 152 levels. The depth of ResNet is amazing, and the deep depth makes the network very powerful. The ResNet has several characteristics as follows:

- Firstly, it has compact structure without increasing extra parameters.
- Secondly, it increases the number of feature maps layer by layer to retain more useful information.
- Thirdly, BN and global average pooling layers are utilized for regularization to accelerate model convergence.

2.2 From ResNet-50 to the Baseline

The 50-layer ResNet (ResNet-50) has the suitable network architecture for person Re-ID, and therefore, many approaches [20–22] employ ResNet-50 as backbone to modify and match the person Re-ID task. We just make one modification on the ResNet-50 to match the classification task. Concretely, the number of neurons in the original fully connected (FC) layer of ResNet-50 is changed from 1000 to 512. Hence, we could obtain a feature vector f with the dimension of 512 from the modified ResNet-50. Afterwards, we design a classifier which is composed of one FC layer and the softmax function to compute identity prediction values.

In the stage of training, we firstly resize each image into a fixed shape, and then the resized image is randomly cropped and fed into the modified ResNet-50. During the test stage, we only resize the pedestrian image, and then the feature vector f could be learned from the trained deep model to represent each query and gallery image for computing distance between them.

2.3 Loss Function

We compute loss using the cross-entropy loss function:

$$\text{Loss} = -\sum_{k=1}^{K} p_k(f) \log q_k(f) \tag{1}$$

where K is the total number of identity, $q_k(f)$ is the identity prediction for the kth identity, and $p_k(f)$ is ground-truth value. If the target label of f is the tth identity, then $p_t(f) = 1$; otherwise $p_k(f) = 0$. We compute the prediction values $q_k(f)$ by softmax function:

$$q_k(f) = \frac{e^{a_k}}{\sum_{m=1}^{K} e^{a_m}} \tag{2}$$

where a_k is the activation value of the kth neuron in the FC layer.

3 Experiments

In this section, we list some experiment settings and provide the experimental results.

3.1 Experiment Settings

During the training stage, we utilize the pre-trained ResNet-50 to speed up the model convergence, and set 120 epochs 32 batch sizes, respectively. As for the basic learning rate, it is initialized to 0.01 and decayed by a factor of 0.1 each 40 epochs. We set values of the weight decay and momentum to be 0.0005 and 0.9, respectively.

3.2 Performance Evaluation

The performance of the baseline is evaluated under different implementation details mentioned in Sect. 1, i.e., resize ratio of the pedestrian image, batch size and basic learning rate. We report mAP and the CMC at rank-1, rank-5 and rank-10 on Market-1501.

Resize ratio of the pedestrian image. We evaluate the effects of the resize ratio by resizing the pedestrian image from 264 × 132 to 280 × 140. In Table 1, the baseline achieves the best performance when resizing each image to 272 × 136.

Table 1. The influence of the resize ratio of the pedestrian image

Resize ratio	Rank-1	Rank-5	Rank-10	mAP
264 × 132	88.12	95.01	96.79	70.61
268 × 134	88.09	95.22	96.97	71.16
272 × 136	**88.84**	**95.64**	**97.39**	**71.72**
276 × 138	88.12	95.25	96.79	71.10
280 × 140	88.03	95.43	96.97	70.88

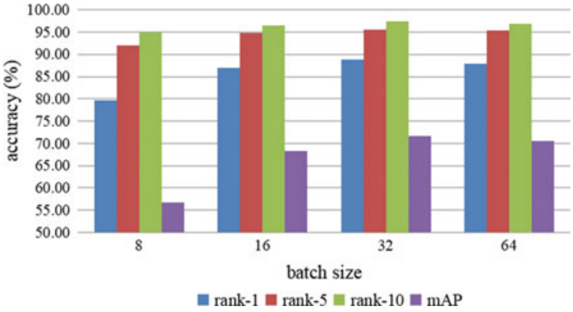


Fig. 1. The influence of the batch size.

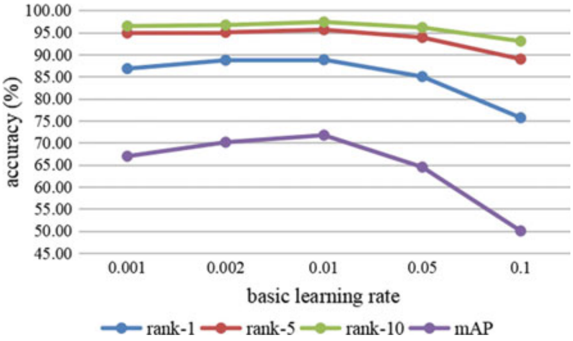


Fig. 2. The influence of the basic learning rate.

Batch size. We test different batch sizes from 8 to 64, and Fig. 1 shows the detailed results. From Fig. 1, we can see that it is suitable for the baseline to set 32 batch sizes and obtain the best results. It is harmful for the performance of the baseline when setting small batch size.

Basic learning rate. We initialize the basic learning rate with different values, such as 0.001, 0.005, 0.01 and so on. Figure 2 shows experimental results and we could see that the baseline may learn worse feature representations when the basic learning rate is initialized with large value or small value.

4 Conclusion

In this paper, we have evaluated the performance of person Re-ID baseline under different implementation details. We modify ResNet-50 as the person Re-ID baseline by changing original FC layer. We have shown that these implementation details could have influence on the performance of the baseline. Our work is helpful and meaningful for training deep model in the person Re-ID community.

Acknowledgements. This work was supported by National Natural Science Foundation of China under Grant Nos. 61501327 and 61711530240, Natural Science Foundation of Tianjin under Grant No. 17JCZDJC30600, the Fund of Tianjin Normal University under Grant No. 135202RC1703, the Open Projects Program of National Laboratory of Pattern Recognition under Grant No. 201800002, and the Tianjin Higher Education Creative Team Funds Program.

References

1. Wang X (2013) Intelligent multi-camera video surveillance: a review. Pattern Recognit Lett 34(1):3–19
2. Loy CC, Xiang T, Gong S (2009) Multi-camera activity correlation analysis. In: IEEE conference on computer vision and pattern recognition, Miami, pp 1988–1995
3. Zhang Z, Wang C, Xiao B, Zhou W, Liu S (2012) Action recognition using context-constrained linear coding. IEEE Signal Process Lett 19(7):439–442
4. Zhang Z, Wang C, Xiao B, Zhou W, Liu S, Shi C (2013) Cross-view action recognition via a continuous virtual path. In: IEEE conference on computer vision and pattern recognition, Portland, pp 2690–2697
5. Gao S, Wang C, Xiao B, Shi C, Zhang Z (2014) Stroke bank: a high-level representation for scene character recognition. In: International conference on pattern recognition, Stockholm, pp 2909–2913
6. Zhang Z, Mei X, Xiao B (2015) Abnormal event detection via compact low-rank sparse learning. IEEE Intell Syst 31(2):29–36
7. Gheissari N, Sebastian TB, Hartley R (2006) Person reidentification using spatiotemporal appearance. In: IEEE conference on computer vision and pattern recognition, vol 2, New York, pp 1528–1535
8. Farenzena M, Bazzani L, Perina A, Murino V, Cristani M (2010) Person reidentification by symmetry-driven accumulation of local features. In: IEEE conference on computer vision and pattern recognition, San Francisco, pp 2360–2367

9. Cheng DS, Cristani M, Stoppa M, Bazzani L, Murino V (2011) Custom pictorial structures for re-identification. In: British machine vision conference, Dundee, pp 1–11
10. Liao S, Hu Y, Zhu X, Li SZ (2015) Person re-identification by local maximal occurrence representation and metric learning. In: IEEE conference on computer vision and pattern recognition, Boston, pp 2197–2206
11. Tan S, Zheng F, Shao L (2015) Dense invariant feature based support vector ranking for person re-identification. In: IEEE global conference on signal and information processing, Orlando, pp 687–691
12. Zheng L, Shen L, Tian L, Wang S, Wang J, Tian Q (2015) Scalable person re-identification: a benchmark. In: IEEE international conference on computer vision, Santiago, pp 1116–1124
13. Li W, Zhao R, Xiao T, Wang X (2015) DeepReID: deep filter pairing neural network for person re-identification. In: IEEE conference on computer vision and pattern recognition, Boston, pp 152–159
14. Ristani E, Solera F, Zou R, Cucchiara R, Tomasi C (2016) Performance measures and a data set for multi-target, multi-camera tracking. In: European conference on computer vision, Amsterdam, pp 17–35
15. Ding S, Lin L, Wang G, Chao H (2015) Deep feature learning with relative distance comparison for person re-identification. Pattern Recognit 48(10):2993–3003
16. Su C, Li J, Zhang S, Xing J, Gao W, Tian Q (2017) Pose-driven deep convolutional model for person re-identification. In: IEEE international conference on computer vision, Venice, pp 3980–3989
17. Geng M, Wang Y, Xiang T, Tian Y (2018) Deep transfer learning for person re-identification. In: IEEE international conference on multimedia big data, Xian, pp 1–5
18. Yao H, Zhang S, Zhang Y, Li J, Tian Q (2019) Deep representation learning with part loss for person re-identification. IEEE Trans Image Process 28(6):2860–2871
19. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, Las Vegas, pp 770–778
20. Cheng D, Gong Y, Zhou S, Wang J, Zheng N (2016) Person re-identification by multi-channel parts-based CNN with improved triplet loss function. In: IEEE conference on computer vision and pattern recognition, Las Vegas, pp 1335–1344
21. Zheng Z, Zheng L, Yang Y (2018) A discriminatively learned CNN embedding for person re-identification. ACM Trans Multimed Comput 14(1):13
22. Sun Y, Zheng L, Yang Y, Tian Q, Wang S (2018) Beyond part models: person retrieval with refined part pooling. In: European conference on computer vision, Munich, pp 480–496

An Ontology for Decision-Making Support in Air Traffic Management

Yin Sheng[1](✉), Xi Chen[1], Haijian Mo[1], Xin Chen[2], and Yang Zhang[1]

[1] State Key Laboratory of Air Traffic Management System and Technology, 1 Yongzhi Road, Qinhuai District, Nanjing City 210000, Jiangsu Province, China
shengyin1218@qq.com, chenxizhongqiu@126.com, aiwin28@163.com, nwpu_zhangy@163.com

[2] Nanjing University of Finance and Economics, 3 Wenyuan Road, Qixia District, Nanjing City 210000, Jiangsu Province, China
msxinchen@nufe.edu.cn

Abstract. Decision-making processes of Air Traffic Management (ATM) need to organize information from different data sources. For example, reroute requires four-dimensional (4D) trajectory and weather information. Future ATM Decision-Support Systems (DSS) are expected to perform trustworthy complex decision-making automatically. This paper proposes an approach to DSS in ATM based on an ontology including concepts and instances of trajectories and meteorology. Temporal and spatial relationships are then set in this ontology. Reasoning rules are also build to represent knowledge in DSS. A case study shows a reroute scenario during a thunderstorm. In this scenario, information of the flight and the weather is combined to support decision-making by the proposed approach.

Keywords: Ontology · Air Traffic Management · Decision-Support System

1 Introduction

Systems in Air Traffic Management (ATM) are complex. Various data sources, e.g., weather information, flight plans, and airport information, must be organized to support the decision-making in the ATM. Crew members and air traffic controllers must process complex information to guarantee the safety of flights.

Utilizing data from different sources in a consistent way is challenging for enterprises, e.g., governments, airlines, or information service providers. Usually,

This work is supported by National Key R&D Program of China (No. 2016YFB0502405) and The Youth Program of Humanities and Social Sciences Foundation of the Ministry of Education of China (No. 18YJC630013).

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 458–466, 2020.
https://doi.org/10.1007/978-981-15-0187-6_55

data models are developed by an organization to facilitate the data to be generated, used, and stored [1]. A data model includes the framework that represents real-world entities, format of encoding, and restrictions on the range of data. However, aviation data providers may adopt a variety of data models. Furthermore, another key requirement for designing future Decision-Support Systems (DSS) for ATM is to perform tasks agilely and reliably with little human effort [2].

Ontology models can describe concepts and their relations, and reason based on description logic. Thus, ontology facilitates decision-making in ATM by improving data combination and reasoning according to ATM rules. This paper proposes an ontology consisting of 4D trajectory and thunderstorm information, as well as a reasoning architecture and an automatic decision-making process in future ATM systems.

This paper is organized as follows. Section 2 gives an literature review of current technologies. Backgrounds of ontology-based decision-making in aviation are introduced in Sect. 3. Sections 4 and 5 discusses the ontology models of four-dimensional (4D) trajectories and thunderstorms. A detailed case study for the proposed approach is shown in Sect. 6. Section 7 concludes this paper and presents future research topics.

2 Related Work

2.1 Ontologies

An ontology [3] is a kind of data framework that is widely used in the many research areas, i.e., artificial intelligence (AI), knowledge engineering, and semantic web [4]. Gruber [5] thinks that an ontology represents "an explicit specification of a conceptualization," where conceptualization is an abstract, simplified view of the world. Borst and Akkermans argue that an ontology is a formal, explicit specification of a shared conceptualization [6]. Formal indicates that the ontology should be understood by computers. Explicit requires that the concepts and the relations between them are defined explicitly. Shared means that a group of people can use an ontology to express knowledge.

An ontology is based on "triples" that are often expressed by a subject-predicate-object structure. Triples are usually expressed by Resource Description Framework (RDF) [7] and RDF Schema (RDFS) [8], and both of them form the foundation for the Web Ontology Language (OWL) [9]. Currently, OWL is one of the most popular languages for semantic modeling today.

In this paper, an ontology is the explicit description of the set of concepts, properties, and relationships in a certain domain. For our purpose, the domain is decision-making in ATM. The concepts are the key components about the ATM domain, e.g., 4D trajectories, airports, and thunderstorms [10].

Properties describe the characteristics of the concepts. Location is a property of a thunderstorm that will be common across all thunderstorms. The location of the thunderstorm may differ, but we can expect each thunderstorm to have one. Each 4D trajectory of a flight, regardless of its instantiation, will have a group of route points related to one flight process. Route points can be further generalized as a structural property of a trajectory.

Relationships demonstrate how concepts interact with others. For example, an aircraft will depart from or arrive at an airport, and the airport is its origin or destination. This airport may provide local weather forecasts and thunderstorm data to support the aircraft to fly in certain airspace. In this paper, we mainly focus on the relationship between the flights and weather.

2.2 Ontologies for Decision-Making in Air Traffic Management

The National Aeronautics and Space Administration (NASA) proposes an ontology-based data integration system that shows the feasibility of integrating, querying, and searching over different sources of heterogeneous ATM data. These data generate from FAA, related enterprises, NASA, etc. In this system, a shared ontology enables stakeholders to bridge the gap of various types of data models in aviation and query date across data sets [11,12].

Traffic Management Initiatives (TMIs) also leverage an ontology to reduce the inconsistent between systems. An ontology is designed as a module of the Traffic Advisory Systems (TAS). TAS manages to reduce traffic imbalances in the National Airspace System (NAS) [13]. The TMI ontology defines a set of shared concepts and relationships, which promote the interoperations and support the TAS to reach its goal.

FAA uses an ontology as a controlled vocabulary for documents. Thus, the terminology used in FAA has a standard, and agencies in FAA search documents with less effort make decision more timely. This ontology is also used to tag semistructured text documents [14].

The above-mentioned papers notice the effectiveness and importance of ontology in ATM, but do not build ontology with spatio-temporal information, neither set-related reasoning rules.

3 Spatio-Temporal Ontology Model

3.1 Temporal Ontology

James Allen and George Ferguson proposed temporal logic in their paper "Actions and Events in Interval Temporal Logic" in July 1994. They hope to present a way of representing temporal information on events and actions based on logic. Compared to previous AI research, interval temporal logic is more expressive and can be easily understood.

Time can be expressed by intervals and instant points. Intervals are time span between two instant points. Usually, intervals have positive time length, time interval with zero length can also be considered as an interval because it has two instants overlapping with each other. Instants are independent points on the timeline that have zero time length.

There are 11 types of temporal relations between two intervals, as shown in Fig. 1. They express the temporal relations between 4D trajectories and thunderstorms.

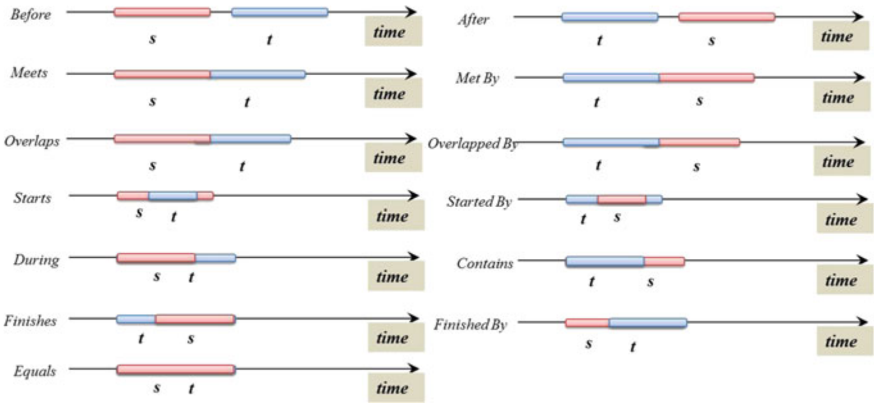


Fig. 1. Temporal ontology

3.2 Spatial Ontology

Spatial ontology depicts the relationship of objects in space. The spatial relationship of objects can be described by the region connection calculus (RCC). RCC aims at qualitative spatial representation and reasoning. In a Euclidean or topological space, RCC8 enumerates eight possible relations that may exist between two regions, as depicted in Fig. 2.

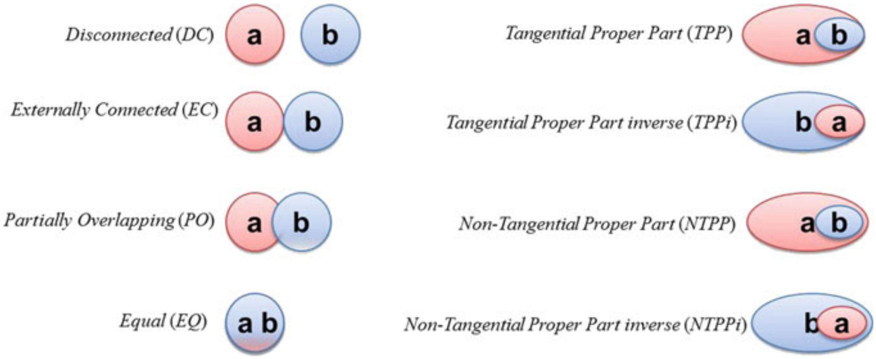

Disconnected (DC)
a
b
Tangential Proper Part (TPP)
a
b
Externally Connected (EC)
a
b
Tangential Proper Part inverse (TPPi)
b
a
Partially Overlapping (PO)
a
b
Non-Tangential Proper Part (NTPP)
a
b
Equal (EQ)
a b
Non-Tangential Proper Part inverse (NTPPi)
b
a

Fig. 2. Spatial ontology

By defining the temporal and spatial relations of 4D trajectories and thunderstorms, we can tell if a route point in a 4D trajectory is covered by a thunderstorm at some time.

4 Ontological Models of 4D Trajectory and Thunderstorm

4.1 Ontological Model of 4D Trajectories

Four-dimensional track management controls flights as much as possible in accordance with the ideal state of airspace users, such that the air traffic flow can be optimized. In the early planning phase of operations, 4D tracks will be shared between airspace users, air navigation service providers, and airport operators. Subsequent improvements will be made considering a number of constraints on airspace and airport capacity.

The project not only provides synchronized track data, but also develops new tools to improve the availability of accurate and uniform aviation and weather information. In this way, any unexpected situation that can cause delays and requires adjustments to scheduled flight operations which can be monitored in advance. Four-dimensional track management allows aircraft to choose direct flights to reduce fuel consumption, carbon dioxide emissions, and costs. In addition, controllers have access to more accurate information for planning operations, smoother management of air traffic, and can handle the expected growth of traffic flow in the next few years.

The ontology model of 4D trajectories is shown in Fig. 3. This ontology is developed according to the trajectory model of FIXM, consisting of necessary elements for describing 4D trajectories.

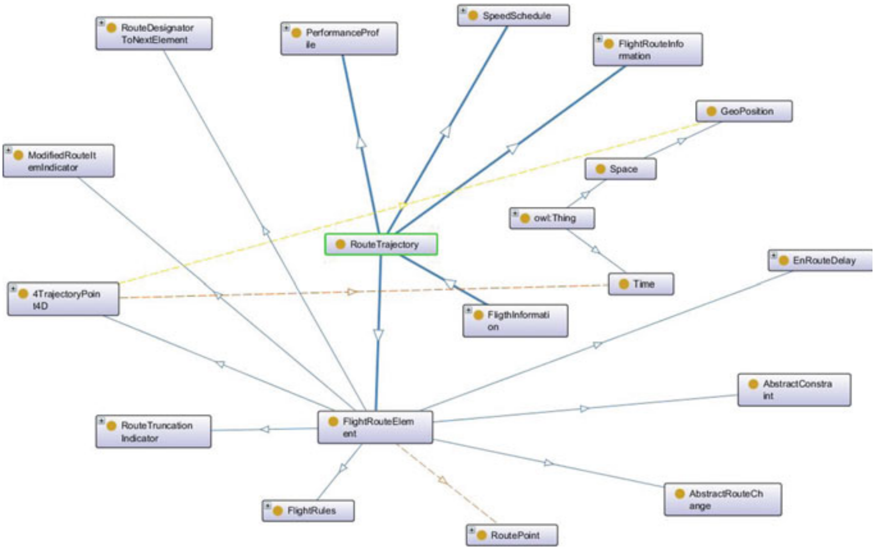


Fig. 3. Four-dimensional trajectory ontology

4.2 Ontological Model of Thunderstorms

Thunderstorms, a natural phenomenon, are localized strong convective weather that occurs in tropical and temperate regions. Thunderstorms can be accompanied by lightning strikes, lightning, strong winds, and significant precipitation, such as rain or hail. The thunderstorm is dangerous for flights to get through. Thus, flights should reroute if its initial trajectory enters a thunderstorm.

The ontology model of a thunderstorm is built to describe the possible conflicts with 4D trajectories, as shown in Fig. 4.

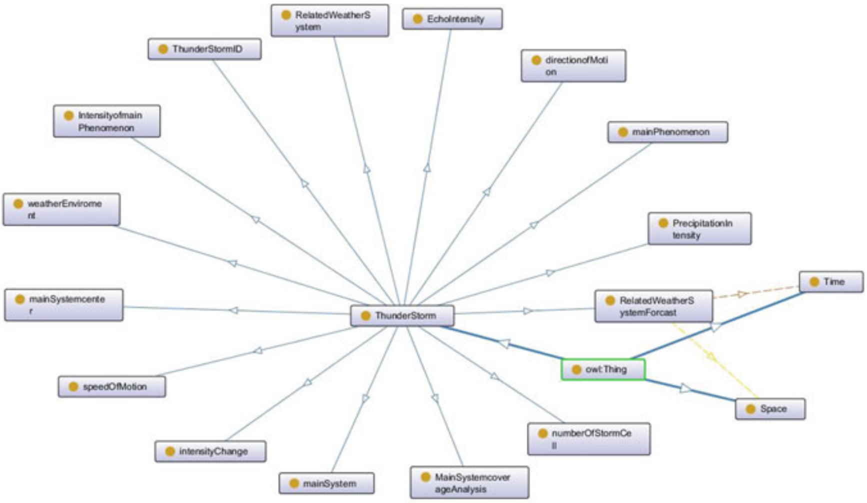


Fig. 4. Thunderstorm ontology

5 Reasoning with Ontology

Description logic is a logical language used for knowledge representation and a reasoning method for it. It is mainly used to describe the relationship between concept classification and its concepts. Most of the description logic methods are used in applications that involve knowledge classification, such as web-oriented information processing.

Description logic is mainly composed of two parts: TBox and ABox. TBox defines the structure of a particular domain of knowledge and contains a set of axioms that can form new concepts through existing concepts. ABox contains instances of concepts in TBox. TBox has the ability to classify (taxonomy). Classification is the first step to systematically build knowledge. By classifying, the ontology of things can be grouped into common concepts, which can be combined into a broader concept. ABox is a collection of instance assertions that indicate individual's attributes or relationships between individuals. As a kind of knowledge representation method, description logic can not only represent the knowledge in the domain but also have the mechanism of reasoning, which can derive the implicit knowledge. The reasoning in the description logic is divided into two types: conceptual reasoning and instance detection. Conceptual reasoning includes determining the consistency (satisfaction) of concepts, inclusion relationships, equivalence relations, and disjointness; instance detection refers to determining whether a given instance belongs to a concept.

Released by Stanford University, Protege software is a software for creating ontologies based on Java. It is also a knowledge-based editor. It is an open-source software. This software is mainly used for the construction of ontology in the Semantic Web. Protege provides the construction of concepts, relationships, attributes, and instances, and users do not need to focus on the concrete ontology description language. Users can construct the domain ontology model at the conceptual level. The Protege tool supports some different reasoners and allows users to choose one from several options. These reasoners can perform reasoning based on description logic. Reasoners are the engine to run the queries based on logical rules. These reasoners can not only execute reasoning rules but also examine semantic consistency. They can also answer logical questions according to concepts and their relationships defined in existing ontologies.

6 Case Study

We demonstrate the process of the proposed approach by a case study in this section. Suppose there is a flight from Shanghai to Beijing, as depicted in Fig. 5. A *4DTrajectory* has a *RoutePoint* and the *time* that the flight passes it. A *Thunderstorm* has a group of *Forcast* results, and each result consists of the affected *area* and *time*. We can add the relations between affected areas and route points by analyzing the geographic position.

Then, Semantic Web Rule Language (SWRL) is used to define reasoning rules. *Overlaps*, *Overlapped By*, *Starts*, *Started By*, *During*, *Contains*, *Finishes*, *Finished By*, and *Equals* are defined as sub-properties of *TemporalConflict*. While *PO*, *EQ*, *TPP*, *TPPi*, *NTPP*, and *NTPPi* are sub-properties of *SpatialConflict*. We define the following reasoning rules to support decision-making in ATM:

$$\begin{gathered}
FourDTrajectory(?tr1) \wedge hasFlightRouteElement(?tr1, ?re1) \wedge \\
hasTrajectoryPoint(?re1, ?pd1) \wedge hasTime(?pd1, ?t1) \wedge hasRoutePoint(?pd1, ?r) \\
\wedge\, ThunderStorm(?th2) \wedge hasForcast(?th2, ?fc2) \wedge hasTime(?th2, ?t2) \\
\wedge\, hasAffectedArea(?th2, ?aa) \wedge TemporalConflict(?t1, ?t2) \\
\wedge\, SpatialConflict(?aa, ?r) \rightarrow conflictWith(?tr1, ?th2).
\end{gathered}$$

Then, we run the reasoner and the results are depicted in Fig. 6.

Based on the proposed ontological model, we can also define reasoning rules to detect the conflicts between two trajectories. That is, if the origin trajectory is infeasible, we generate a new one. With these reasoning rules, we can tell if the new trajectory conflicts with a thunderstorm or other trajectories.

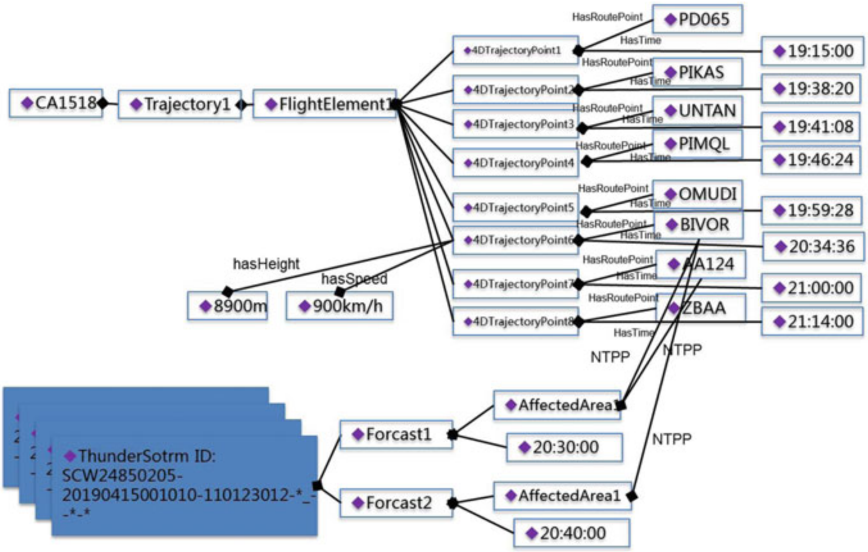
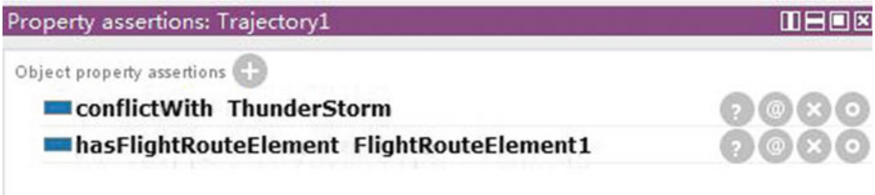


Fig. 5. Trajectory from Shanghai to Beijing


Property assertions: Trajectory1

Object property assertions

conflictWith ThunderStorm

hasFlightRouteElement FlightRouteElement1

Fig. 6. Result of the case study

7 Conclusion

This paper proposes an ontology-based approach to decision-making in Air Traffic Management. Reasoning rules are defined within this ontology. Conflicts between trajectories and thunderstorms can be detected by running these rules. The results show the effectiveness of the proposed approach. The proposed approach can also be used in different scenarios in ATM [15].

The future work may be conducted along with the following topics:

- ontology for multi-criteria decision-making;
- fuzzy ontology models of ATM; and
- ontology models of the process of thunderstorms.

References

1. Keller RM (2016) Ontologies for aviation data management. In: 2016 IEEE/AIAA 35th digital avionics systems conference (DASC). IEEE, pp 1-9
2. Blasch E, Kostek P et al (2015) Summary of avionics technologies. IEEE Aerosp Electron Syst Mag 30(9):6–11
3. Chandrasekaran B, Josephson JR, Benjamins VR (1999) What are ontologies, and why do we need them? IEEE Intell Syst 1:20–26
4. Fensel D (2001) Ontologies. Springer
5. Gruber TR (1993) A translation approach to portable ontology specifications. Knowl Acquis 5(2):199–220
6. Borst WN, Akkermans JM (1997) Engineering ontologies. Int J Hum-Comput Stud 46(2/3):365–406
7. W3C (2014) RDF 1.1 concepts and abstract syntax. Available from: http://www.w3.org/TR/rdf11-concepts/
8. W3C (2014) RDF schema 1.1. Available from: http://www.w3.org/TR/rdfschema/
9. W3C (2014) OWL 2 web ontology language document overview, 2nd edn. Available from: https://www.w3.org/TR/owl2-overview/
10. Miller D (2017) An ontology for future airspace system architectures. In: 2017 IEEE/AIAA 36th digital avionics systems conference (DASC). IEEE, pp 1–8
11. Keller RM (2015) NASA's ATM ontology. Paper presented at the Air transportation information exchange conference, Silver Spring, MD
12. Keller RM (2015) Data integration using the NASA air traffic management ontology. Paper presented at the Semantic web for air transportation (SWAT-2015), Washington, DC
13. FAA (2009) Traffic flow management in the National Airspace System. Available from: https://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf
14. Cowell D, Buchanan CL, Eberhardt J (2015) Enterprise information management (EIM) at the FAA: translating semantic technologies into direct user benefit. Paper presented at the Semantic web for air transportation (SWAT-2015), Washington, DC
15. Yan Y, Ding H (2019) Big data value embodiment of air traffic management in business operation. Command Inf Syst Technol 10(1):7–12

Flight Conflict Detection and Resolution Based on Digital Grid

Yang Yi[1](✉), Ming Tong[1], and LiuXi[2]

[1] State Key Laboratory of Air Traffic Management System and Technology,
Nanjing 210000, China
nuaa_yang@nuaa.edu.cn
[2] Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Abstract. For high-altitude control area, in order to assist the ground control personnel to monitor short-term flight conflicts in real-time, and to solve the complex flight conflicts that may occur in multi-aircraft from a global perspective, this paper first establishes a spatial digital grid model based on flight safety intervals, and reasonably simplifies the motion model. Secondly, the short-term reachable domain of the aircraft is transformed into grid coordinates, and the numerically dimensioned method is used to obtain the conflicting reachable domain grid coordinates. On the basis of conflict detection, dynamic programming method is used to select the mutually exclusive grid coordinates which are obtained by traversing grid coordinates, and finally the conflict resolution decision of each aircraft is determined by the performance index. The simulation results show that the proposed method can effectively solve the complex flight conflict problem and provide the resolution decision that meets the safety interval and performance constraints before the TCAS (Traffic Collision Avoidance System), while the algorithm time can meet certain real-time requirements.

Keywords: Flight conflict detection and resolution · Grids model · Dynamic programming

1 Introduction

With the rapid development of civil aviation industry, air traffic flow has increased dramatically [1]. In civil aviation alone, in 2018, 610 million passengers were transported, which has an increase of 10.9% over the previous year [2]. Meanwhile, considering that military aviation, airfreight, and other air activities have becoming more frequent, high altitude control areas in China gets more crowded, which leads to higher properties of flight conflict [3].

At present, there are three main ways to prevent flight conflicts: air traffic control deployment (ATC), Airborne Collision Avoidance (TCAS) and visual obstacle avoidance [4, 5]. ATC can command and deploy aircraft from a global perspective, while TCAS can only achieve local conflict resolution. Considering that the present ATC system is a typical man-in-loop decision system [6], in which the control decisions rely on both established control rules and controllers' experience [7]. In order to

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 467–479, 2020.
https://doi.org/10.1007/978-981-15-0187-6_56

detect the flight conflict fast and efficiently and give optimal conflict resolution in complex conditions, many scholars have done relevant research works. In 1964, Reich established the aircraft collision model theory [8], in which the aircraft was assumed to be a cuboid, thus the flight conflict detection only needs to detect whether the point collided with the cuboid. Folton proposed the idea of computational geometry to solve the problem of multi-aircraft collision detection [9], which uses the proximity of Voronoi polygon to reduce the number of detection between two aircraft. Anderson and Lin built a cross-route collision avoidance model based on the conflict area [10]. Erdmann proposed the collision-free path to prevent collisions between aircrafts [11]. The above researches mainly focus on the medium and long-term conflict detection, without considering the maneuverability of multiple aircraft in the short-term detection process. The above methods mainly focus on the medium and long-term conflict detection of aircraft, without considering the maneuverability of multiple aircraft in the short-term detection process.

While this paper focuses on resolving the short-term flight conflict and represents a conflict detection and resolution method based on digital grid partition, which established the spatial grid model to discretize the kinematics model into decision-making model. Through the model transformation, the short-term reachable region of the aircraft can be represented by grid numbers, which can reduce the complexity of the algorithm for multi-aircraft conflict detection. Besides, the dynamic programming algorithm based the grid model is used for solving conflict resolution problems. This method can obtain the conflict resolution in real-time, as a terminal control means, it can reduce the burden of personnel as well as human errors, and provide guarantee for flight safety.

2 Mathematical Model

2.1 Spatial Grid Model

Spatial division is a static division of airspace structure. The standard of flight interval is mainly defined in three directions: vertical interval, longitudinal interval and lateral interval [12], and the safe standard could be the minimum spatial interval or time interval. Referring to RVSM airspace, the safe interval in the vertical direction is 300 m, the safe interval in the longitudinal plane is 20 km. According to the safe interval standard, the airspace can be divided into several grid cells as shown in Fig. 1.

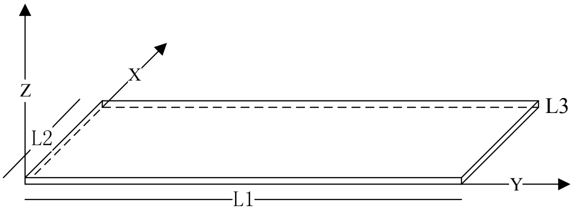


Fig. 1. Grid cell diagram

Figure L_1, L_2, L_3 shows the safety distance in each direction. The entire airspace is divided according to the cell grid shown in the above figure, and the coordinates of each aircraft position can be converted into grid coordinates. Since the length, width and height of the grid are determined by the safety distance, any two aircraft can satisfy the conflict-free flight only by satisfying the following equation.

$$|X_1 - X_2| > 1 \parallel |Y_1 - Y_2| > 1 \parallel |Z_1 - Z_2| > 1 \tag{1}$$

where X, Y, Z, respectively, represent the grid coordinates of the aircraft, and the subscripts represent the aircraft number. It can be seen that maintaining a safe distance between the aircraft only needs to ensure that at least one cell grid is arranged between the aircraft as shown in (1).

Under the conflict-free condition of the airspace model, the minimum grid coordinate difference between the aircraft is 2, The ultimate distance of the real position of the aircraft appears at the boundary of the unit grid, the minimum distance L_1, L_2, L_3, the maximum distance $3L_1$, $3L_2$, $3L_3$, and the distance interval is 100–300% of the safety distance. The model guarantees the safety of multi-aircraft flight and the rapidity of conflict detection by sacrificing part of the airspace.

2.2 Aircraft Model

Simplify the model by combining civil aviation flight and ATC characteristics:

1. The speed of the aircraft remains the same.
2. Radar control is applied to the ground [13], assuming that the position and velocity direction information of all aircraft can be obtained.
3. In order to ensure the airworthiness of the aircraft, the available overload can be limited to a certain range.
4. Only for short-term conflicts within the reachable domain of each aircraft.

3 Conflict Detection and Solution Algorithm

3.1 Conflict Detection

Based on the above model, this paper mainly studies short-term conflict detection, that is, conflicts occurring in the next few seconds to several minutes [13]. The existing airborne TCAS system can provide traffic consultation 40 s in advance and provide decision consultation 20 s in advance [14]. However, the TCAS system needs to ensure that the answering machines of both parties do not malfunction and are not interfered by other signals. In the event of an abnormal situation or when TCAS has not yet provided the decision-making consultation stage, the ATC-B can obtain the positional speed information of the aircraft using the broadcast automatic correlation monitoring (ADS-B) to participate in the short-term collision detection and release process of the aircraft [15].

Due to the influence of the wake of the aircraft, considering the safe distance between the aircraft and the flight in the same direction, the detection time of the aircraft is defined as:

$$T = n\Delta t = n\frac{d}{V} \tag{2}$$

where, $n \in N$ represents the predicted step size, and the speed of the aircraft is determined. To ensure the applicability of the algorithm, the maximum prediction time $\boldsymbol{T}$ is not less than the consultation time provided by the TCAS system for 20 s, thereby determining that the maximum of n is $n_{\max} = 6$, and the area that can be reached at each period time is shown in Fig. 2.

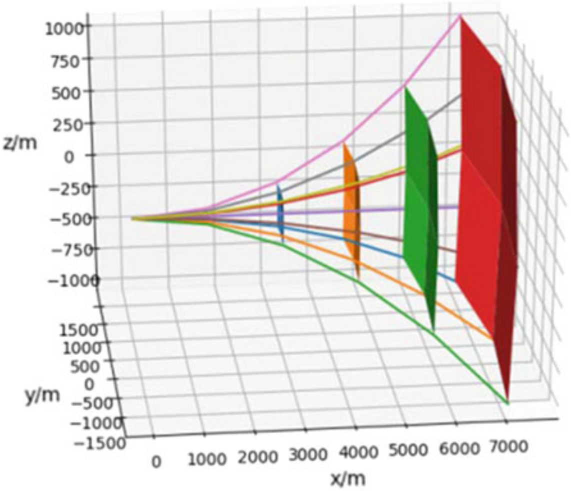


Fig. 2. Aircraft arrival area at different times

For the m-frame aircraft in the high-altitude control area, the traditional conflict detection method is used to compare the positions of every two aircrafts using different strategies at different moments. The comparison times are at least:

$$O_1(m) = n_{\max} A_{\text{num}}^{m} \tag{3}$$

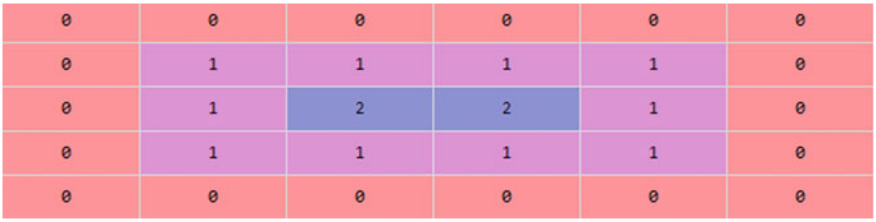
The amount of computation of the above equation increases significantly with the increase in the number of aircraft. In this paper, the spatial grid model is proposed, and the coordinates of the $n_{\max}$ nodes corresponding to each trajectory are converted into corresponding grid coordinates. The conversion is as follows:

$$\begin{cases} X = \coprod \left(x \middle/ L_2 \right) \\ Y = \coprod (y/L_1) \\ Z = \coprod \left(z \middle/ L_3 \right) \end{cases} \tag{4}$$

X, Y, Z denotes the grid coordinates, numerically annotate the grid coordinate boundary of each node and its internal region, set the initial value $N = 0$ of the non-forbidden flight zone grid, the initial value of the no-fly zone grid $N = 1$, perform the following operations on the values corresponding to the grid coordinates:

$$N = N + 1 \tag{5}$$

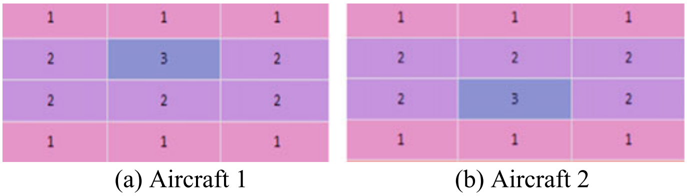
According to the above method, the cross-section of the numerical labeling of a single aircraft at the node position is shown in Fig. 3.



0	0	0	0	0	0
0	1	1	1	1	0
0	1	2	2	1	0
0	1	1	1	1	0
0	0	0	0	0	0

Fig. 3. Single aircraft collision-free numerical labeling cross-section

When multiple aircrafts collide, the numerical labeling cross-sections at the conflict position are shown in Fig. 4.



1	1	1
2	3	2
2	2	2
1	1	1

(a) Aircraft 1

1	1	1
2	2	2
2	3	2
1	1	1

(b) Aircraft 2

Fig. 4. Multi-aircraft conflict numerical labeling cross-section

The node grid coordinates of (a) and (b) are located at adjacent positions, and after summing with each other, the value is an outlier with a numerical value greater than 2. If the coordinate corresponding to the value is not the node grid coordinate, it may be composed of the boundary 1 of multiple aircrafts, which has no practical significance.

The outliers shown in the above figure correspond to the node grid coordinates, so the two aircraft cannot simultaneously adopt the maneuvering mode to reach the coordinates.

Similarly, for the flight conflict caused by the airspace limitation, the no-fly zone grid is processed by the numerical labeling method in the initialization process, and the form is the same as above, as shown in Fig. 3.

The basic flow chart for short-term collision detection of aircraft is as shown in Fig. 5.

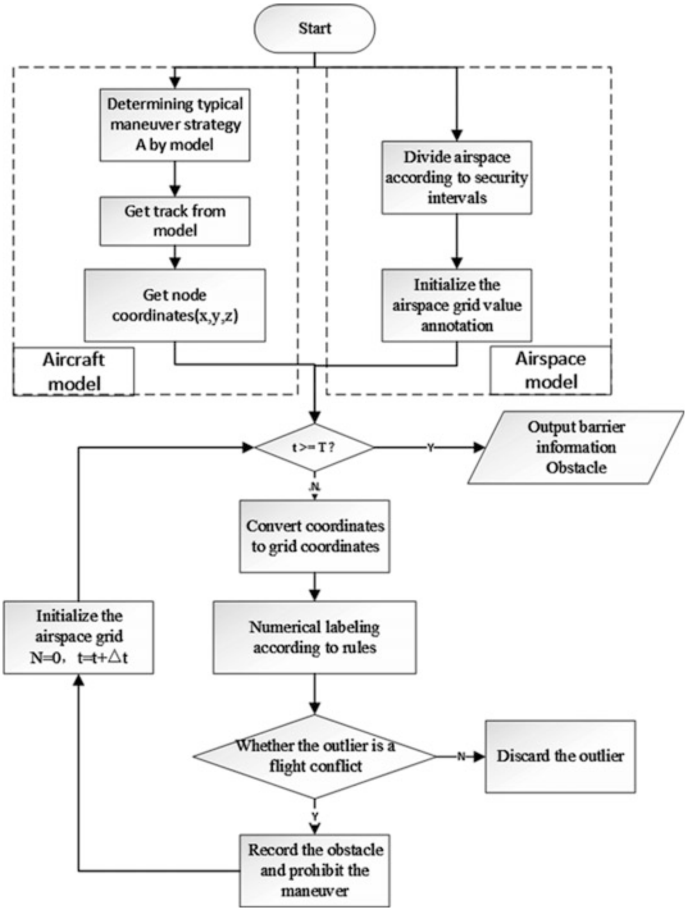

Start
Determining typical maneuver strategy A by model
Get track from model
Get node coordinates(x,y,z)
Aircraft model
Divide airspace according to security intervals
Initialize the airspace grid value annotation
Airspace model
t>=T?
Y
Output barrier information Obstacle
N
Convert coordinates to grid coordinates
Numerical labeling according to rules
Initialize the airspace grid N=0, t=t+Δt
Whether the outlier is a flight conflict
N
Discard the outlier
Y
Record the obstacle and prohibit the maneuver

Fig. 5. Flow chart of aircraft conflict detection

Referring to the above process, according to the aircraft model, maneuver strategy and environment model, the predicted position set at each time can be obtained. Considering the flight conflict detection of m aircraft, the detection complexity can be calculated as follows:

$$O_2(m) = 2\sum_{n=0}^{n_{\max}} mP(n\Delta t) \tag{6}$$

where, the complexity is mainly caused by P. The reference (11) is related to the choice of maximum time and maneuvering strategy, which mainly affects the reachable region boundary. Coefficient 2 detects the outliers after assignment, its complexity increases linearly with the number of aircraft, which is much less than the exponential growth of (3), and the complexity of summation calculation is much less than that of calculating relative distance.

Traditional conflict detection aims at mid-and long-term flight conflicts and only relieves them based on rule base, which can't guarantee the effectiveness of relief in the case of large number of aircraft. This paper mainly focuses on the conflict in the reachable area of short-term aircraft. The method of digital grid can not only greatly reduce the detection complexity of the conflict in the reachable area of short-term aircraft, but also provide a reliable basis and guarantee for conflict resolution.

3.2 Conflict Resolution

In this paper, conflict resolution is based on the aforementioned conflict detection methods. The conflict information *Obstacle* obtained in the detection process mainly includes the node grid coordinates X_e, Y_e, Z_e where the conflict occurs and the corresponding maneuvering strategies of each aircraft. However, for the complex conflict situation of multi-aircraft, firstly, we need to analyze the relationship between the grid coordinates of the collision nodes. Reference (1) compares the grid coordinates of each node, determines the grid coordinates of the collision nodes and classifies them into the same collision. At the same time, we can get the number of the aircraft and the maneuvering mode. When satisfying the constraints of dynamics and velocity dip angle, the conflict resolution is transformed into the selection process of strategy, that is, According to the above conflict information, the maneuvering decision of each aircraft is reasonably selected by planning method to maximize the total performance index. The flow chart for distinguishing collision node grids is shown in Fig. 6.

In the figure above, *Connect* is a multivariate combination leading to conflict, the elements in which is corresponded to the discriminant between the aircraft number $\boldsymbol{F}$ as well as the decision number $\boldsymbol{O}$ of arriving at the conflict location, which can be obtained by (1). By traversing to conflicting combinations (F_1O_1, F_2O_2), the maneuvering strategies in the combinations will not conflict in different moments. Therefore, there is at least one of the decisions $\boldsymbol{a}$ is negated in all combinations (F_1O_1, F_2O_2), and all decisions $\boldsymbol{a}$ can be combined into action sets *action* as follows:

$$\text{action}(t) = \left\{ \begin{array}{l} (a, \ldots)|a = (FO_1, \ldots FO_i),\ FO_i \in (FO_{i1}, \\ FO_{i2}),\ (FO_{i1}, FO_{i2}) \in \text{Connect}(t) \end{array} \right\} \tag{7}$$

After obtaining the action set without conflict, the initial state of the aircraft can be set to execute all the decisions included in the decision.

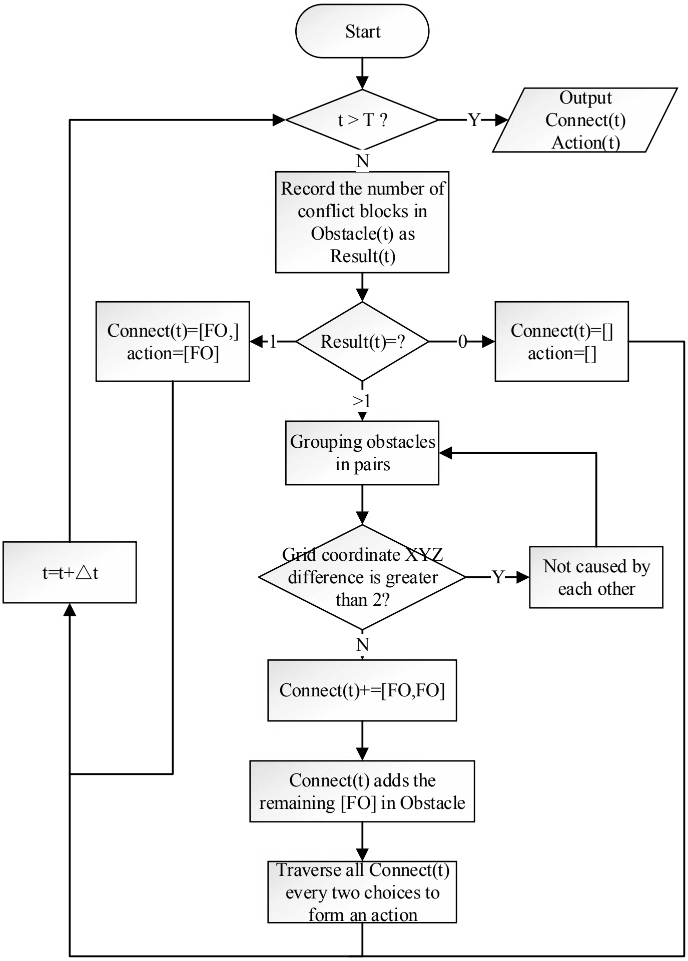

Start
t > T ?
Y
Output Connect(t) Action(t)
N
Record the number of conflict blocks in Obstacle(t) as Result(t)
Result(t)=?
1
Connect(t)=[FO,] action=[FO]
0
Connect(t)=[] action=[]
>1
Grouping obstacles in pairs
Grid coordinate XYZ difference is greater than 2?
Y
Not caused by each other
N
t=t+△t
Connect(t)+=[FO,FO]
Connect(t) adds the remaining [FO] in Obstacle
Traverse all Connect(t) every two choices to form an action

Fig. 6. Flight conflict matching and action output

$$S = \{(s_1, \ldots s_m)|s_i = (A_0, \ldots A_{\text{num}})\} \tag{8}$$

The initial state will change under *action*.s_i represents the executable strategy set of aircraft i. Considering the performance constraints of aircraft, the executable strategy of aircraft is further restricted. That is, when the velocity inclination angle is too large (too small), the strategy of upward (downward) maneuver should not be adopted. The executable strategy S of all aircraft satisfies no aftereffect, that is, the future is only related to the current state, but not to the past state. The state transition process can be expressed as follows:

$$S_t \underset{\coprod}{\xrightarrow{\text{action}(t)}} \theta S_{t+\Delta t} \tag{9}$$

where Цθ represents the constraint of satisfaction, S_t represents the set of policies that can be adopted at the current time, and $S_{t+\Delta t}$ represents the set of policies that can be adopted at the next prediction time. In the process of state transition, *action*(t) functions as follows:

$$S_{t+\Delta t} = \coprod_{\theta}(S_t - S_a) \tag{10}$$

where,

$$S_a = FO, FO \in a \tag{11}$$

where, FO represents the maneuvering strategy corresponding to the O-th decision-making number of the F-th aircraft, that is, the maneuvering mode that the aircraft cannot adopt.

According to the state transition process (10), the state of each aircraft at the next prediction time is updated continuously, and finally S_T is obtained by iteration. At this time, the executable strategy set of any two aircraft has met the safety requirement (1). Design optimization performance indicators as following:

$$J = \underset{a \in \text{action}}{\text{opt}} \sum_{i=0}^{m} \min K_i(s_i)M, \quad s_i \in S_T \tag{12}$$

where, S_T represents the set of executable strategies of the aircraft at the maximum predictive time, which is determined by the state transition in (16), and K in (20) represents the performance indicators of each aircraft under the current executable strategy, which is specifically expressed as:

$$K = \sqrt{(x_t - x_{\text{end}})^2 + (y_t - y_{\text{end}})^2 + (z_t - z_{\text{end}})^2} \tag{13}$$

The end subscript represents the position coordinates of the target point of the aircraft, and t is determined by the following equation:

$$t = \text{step} * \Delta t \tag{14}$$

where, Δt is the predicted time interval and *step* is the updated step in the simulation process, which affects the release decision-making and simulation operation efficiency of the aircraft. Equations (12), (13), and (14) give the final release decision for each aircraft:

$$(F_0 \ldots F_m) = \underset{(A_0 \ldots A_m)}{\arg\min} \left(\underset{S}{\arg\min} J \right) \tag{15}$$

By seeking the smallest performance index J to obtain optimal parameters S and combining (13), the nearest decision (A_0, …, A_m) from the target point can be obtained. F represents aircraft in (15). Each aircraft updates its position, velocity, and direction by the decision, and finally reaches the target point.

The complexity of the algorithm above is mainly embodied in the use of obstacle information to form a mutually exclusive combination of aircraft and obstacle numbers, and the final feasible strategy is obtained according to these combination iterations. Multiple traversal combinations can greatly reduce the dimension of combinations and the number of iterations. Further, dynamic programming performance indicators are only related to the final state and a large number of unreasonable states can be eliminated in the iteration process.

4 Simulation Verification

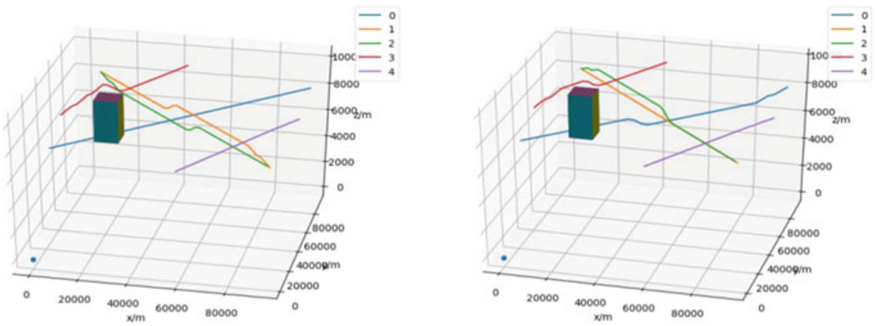
A typical flight conflict is designed and simulated in this paper to validate the effectiveness of the proposed flight conflict detection and resolution algorithm based on digital grid.

The spatial grid and coordinate system are shown in Fig. 1, in which the size is determined by the safe interval distance with $L_1 = L_2 = 5000$ m, $L_3 = 300$ m. The executable strategy of an aircraft is shown in Fig. 2. In order to ensure that the maximum predicted distance is larger than the final RA range of TCAS, the maximum predicted step size $n_{\max} = 6$ is set. The starting and ending points of five aircraft satisfy the following formats:

$$\begin{aligned}
fl &= [V, [x, y, z], [x_{\text{end}}, y_{\text{end}}, z_{\text{end}}], \theta, \psi] \\
fl_0 &= [280, [5, 5, 8], [95, 95, 8], 0, pi/4] \\
fl_1 &= [280, [5, 95, 8], [95, 5, 8], 0, \text{ - } pi/4] \\
fl_2 &= [280, [95, 5, 8], [5, 95, 8], 0, 3pi/4] \\
fl_3 &= [280, [45, 95, 9], [5, 25, 9], 0, \text{ - } 3pi/4] \\
fl_4 &= [280, [55, 5, 7], [95, 75, 7], 0, pi/4]
\end{aligned} \tag{16}$$

in which, the position coordinates are in KM units. According to the calculation, the first collide can be obtained after 22.5 s in simulation time (the 20th simulation cycle in the flight step). While the grid coordinates [3, 10, 30, −20] are set as the no-fly zone, and the aircraft 3 encounters the no-fly zone. The air-craft 4 is added to verify the rapidity of the algorithm for multi-aircraft. The flight trajectory of the collision detection and resolution algorithm presented in this paper is shown in Fig. 7.

It can be seen that two different methods can ensure the safety interval between aircraft, aircraft and no-fly zone. Different maneuvering methods have different maneuvering modes and the same performance index. Because of the setting of safe distance and maneuverability, all obstacle avoidance is from altitude, and the XY plane can't meet the safety requirements. Vehicle 3 avoids the no-fly zone effectively and finally reaches the target point. The mesh spacing difference and maximum of vehicle 012 are studied separately as shown in Fig. 8.



(a)Dynamic Programming Method

(b)Violent Search Method

Fig. 7. The flight trajectory diagram of 5 aircraft

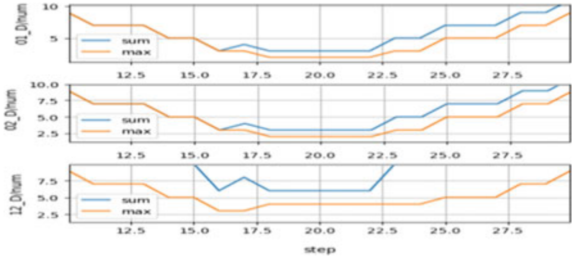


Fig. 8. Aircraft mesh distance map

The graph mainly shows the maximum mesh distance of 012 aircraft during the whole simulation process. From the initial parameter setting, it is known that collision will occur in the 20th simulation cycle. It can be seen from the figure that the safety Formula (1) is satisfied by any two aircraft in the 20th step of collision prediction, i.e., the flight safety interval requirement. The distance of their real trajectory is shown in Fig. 9.

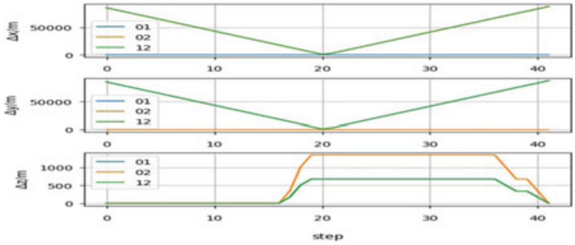
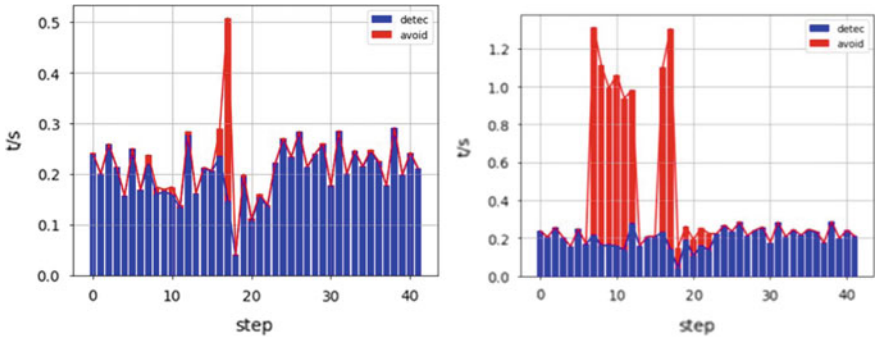


Fig. 9. Distance map of aircraft

To ensure real-time conflict detection and resolution, the whole flight conflict detection and resolution time of the two methods is as shown in Fig. 10.



(a) Dynamic programming method (b) Violent search methods

Fig. 10. Detection release time

Flight conflicts are detected by aircraft in the 7th and 18th cycles. The detection time is shorter than 0.3 s on Python platform. In post-conflict detection, the violent search method takes longer time to extricate, which does not meet the real-time requirements, while the dynamic programming method takes shorter time. It can ensure a certain real-time performance on other computing platforms for the dynamic programming method.

5 Conclusions

In order to ensure air traffic safety and reduce the situation of controllers in the complex conflict conditions of multiple aircraft, this paper proposes a method based on digital grid to detect and release flight conflicts. This method improves the automation performance of the air traffic control system. The main advantages are: (1) The prediction range is larger than the recommended decision range of the aircraft TCAS system, auxiliary air traffic controllers make command decisions on multiple aircraft; (2) Provides conflict resolution optimal solutions for TCAS system failures or unreliable conditions. The conflict detection and liberation in this paper are based on the digital grid model of the airspace which sacrifices part of the space and greatly simplifies the calculation of the safety interval. The motion model of the aircraft is fully considered to obtain the reachable area for collision detection, and the dynamic planning method is used to obtain all the non-conflicting maneuver state combinations of each aircraft. Finally, the optimal conflict resolution decision is obtained according to the performance index. At the same time, the current liberation algorithm still has some shortcomings, and the related problems of the simulation algorithm data structure and optimal parameters need to be further optimized.

References

1. He SUN, Yueru CAI, Wei ZHOU (2016) Prediction of aircraft multi-machine flight conflict status. Comput Simul 33(6):59
2. Qifeng Q (2010) Research on anti-collision strategy in air traffic management. Southwest Jiaotong University, Chengdu
3. Minggong W, Zekun W, Xiangxi W (2019) Geometric optimization model for flight conflict relief. Syst Eng Electr
4. Chen Z (2016) Technical challenges faced by future air traffic control system development. Command Inf Syst & Technol 7(6):1–5
5. Yan Y, Cao G (2018) Operational concepts and key technologies of next generation air traffic management system. Command Inf Syst & Technol 9(3):8–17
6. Reich PG (1966) Analysis of long-range air traffic system: separation standards. J Ins Navig 19(1, 2, 3)
7. Aurenhammer F (1991) Voronoi diagrams—a survey of fundamental geometric data structure. ACM Comput Surveys 23(3):9
8. Anderson D, Lin XG (1996) Collision risk model for a crossing track separation methodology. J Navig 49(2):337–349
9. Mao ZH, Feron E, Bilmoria K (2000) Stability of intersecting aircraft flows under decentralized conflict avoidance rules. In: AIAA guidance navigation and control conference, Denver, CO, pp 14–17
10. Air Traffic Control Committee of the Central Military Commission of the State Council (2002) Flight Interval Standard 5, 31
11. ICAO (1998) Handbook on airspace planning methods for determining interval standards. ICAO document, Doc. 9689-AN/953
12. Bicchi A, Marigo A, Pappas G, Pardini M, Parlangeli G, Tomlin C, Sastry S (1998) Decentralized air traffic management systems: performance and fault tolerance. In: Proceedings IFAC workshop on motion control
13. Li P (2013) Research on the detection and release strategy of flight conflicts. China Civil Aviation Flight Academy
14. Zhang A (2018) Research on simulation technology of TCASII collision avoidance model. China Civil Aviation College
15. Luo WT, Zhao ZR, Zhang DY (2011) Investigation of aircraft collision avoidance and early warning algorithm based on ADS-B. Control Eng China 4:559–563

Abnormality Diagnosis of Musculoskeletal Radiographs Combined with Shallow Texture Features

Yunxue Shao and Xin Wang(✉)

College of Computer Science, Inner Mongolia University, Inner Mongolia,
People's Republic of China
csshyx@imu.edu.cn, 13234751492@163.com

Abstract. In this paper, a new algorithm for anomaly classification combined with shallow texture features is proposed for radioactive musculoskeletal images. The classification algorithm consists of three steps. The radioactive image is first preprocessed to enhance image quality. The local binary pattern (LBP) features of the image are then extracted and merged. Finally, the merged dataset is sent to the DenseNet169 convolutional neural network to determine whether it is abnormal. The method presented in this paper achieved an accuracy of 79.64% on the musculoskeletal radiographs (MURA) dataset, which is superior to the method that does not combine texture features. The experimental results show that the shallow texture features of the combined image can more fully describe the difference between the lesion area and the non-focal area in the image and the difference between different lesion properties.

Keywords: Local binary pattern (LBP) · Musculoskeletal radiographs (MURA) · DenseNet

1 Introduction

Work-related musculoskeletal disorder (WMSDs) refers to systemic muscle, bone, and nervous system disorders caused by occupational factors. During the work process, the musculoskeletal system needs to bear the ergonomic load such as posture load and strength load, because the operator needs to maintain repetitive movements or a compulsory position to carry or lift heavy objects. And, the musculoskeletal system of the limbs is damaged, and there are irritations such as acid, numbness, swelling, and pain [1]. More than 1.7 billion people worldwide are affected by musculoskeletal diseases, so it is important for the abnormal detection of musculoskeletal.

In recent years, deep learning has become a hot spot in the field of machine learning research. Image features extracted by deep convolutional neural networks (DCNN) have proven to be effective in image classification, segmentation, or retrieval applications. Based on its advantages in non-medical images, DCNN is beginning to be gradually applied to medical image classification and detection problems. For example, Spanhol proposed the use of DCNN to classify breast cancer pathology images [2]. Li proposed a DCNN-based pulmonary nodule classification system [3]; Roth proposed

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 480–486, 2020.
https://doi.org/10.1007/978-981-15-0187-6_57

an algorithm for developing a lymph node detection system using DCNN [4]. Based on the above successful experience of DCNN applied to medical images, we tried to use DCNN to classify radioactive musculoskeletal. However, the images currently used to train the DCNN model have the following drawbacks:

1. The quality of the training images is not high.
2. For radioactive image datasets, DCNN does not extract its texture features very well.

Therefore, this paper proposes to preprocess the image to improve the image quality, then fuse the shallow texture features and then send them into the Densnet169 network for classification, and compare the classification effects without using the shallow texture features.

2 Musculoskeletal Abnormality Diagnosis Based on Densenet

The method proposed in this paper is shown in Fig. 1 including image preprocessing, image fusion, and training depth neural network.

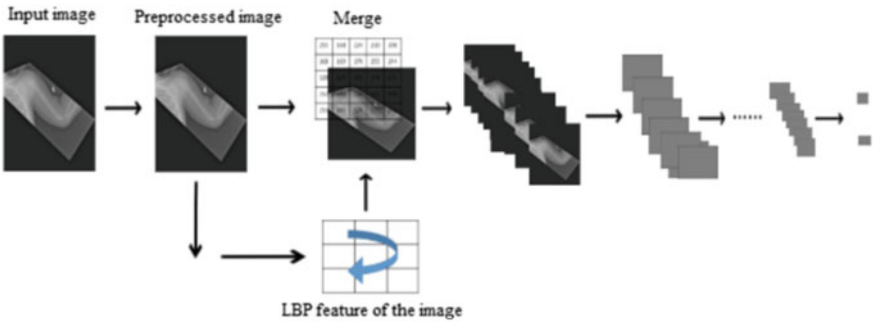


Fig. 1. The structure of the algorithm

2.1 Preprocessing

Histogram equalization is a method of enhancing image contrast (image contrast). In experiments, we used contrast limited adaptive histogram equalization (CLAHE) [5] for image enhancement. The histogram value of the CLAHE method is:

$$\mathrm{Hist}'(i) = \begin{cases} \mathrm{Hist}(i) + L & \mathrm{Hist}(i) < T \\ H_{\max} & \mathrm{Hist}(i) \geq T \end{cases}$$

where Hist(i) is the derivative of the cumulative distribution function of the sliding window local histogram, and $H_{\max}$ is the maximum height of the histogram. We cut off the histogram from the threshold T and then evenly distribute the truncated portion over the entire grayscale range to ensure that the total histogram area is constant, so that the entire histogram rises by a height L. The preprocessed image is shown in Fig. 2.

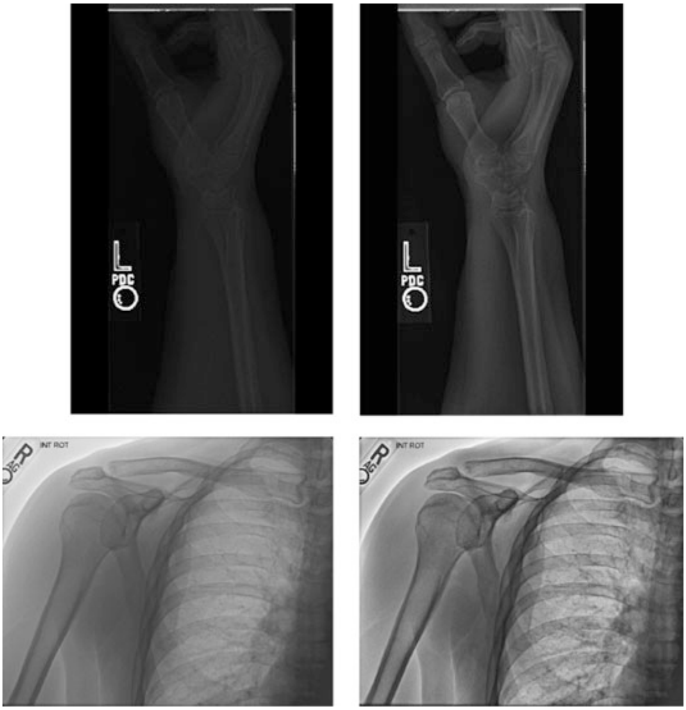


Fig. 2. The original image on the left and the preprocessed image on the right

2.2 DenseNet Neural Networks

In the field of imagery, deep learning has become a mainstream method. We chose DenseNet [6] as our model because it has better performance than ResNet in the case of less parameter and computational cost.

DenseNet's structure is to interconnect all layers, specifically each layer will accept all of its previous layers as its additional input, as shown in Fig. 3, and DenseNet directly merges feature maps from different layers, which can be achieved feature reuse to improve efficiency. We experimented with the DenseNet 169 layer network structure, adjusting the number of neurons in the last layer to 2.

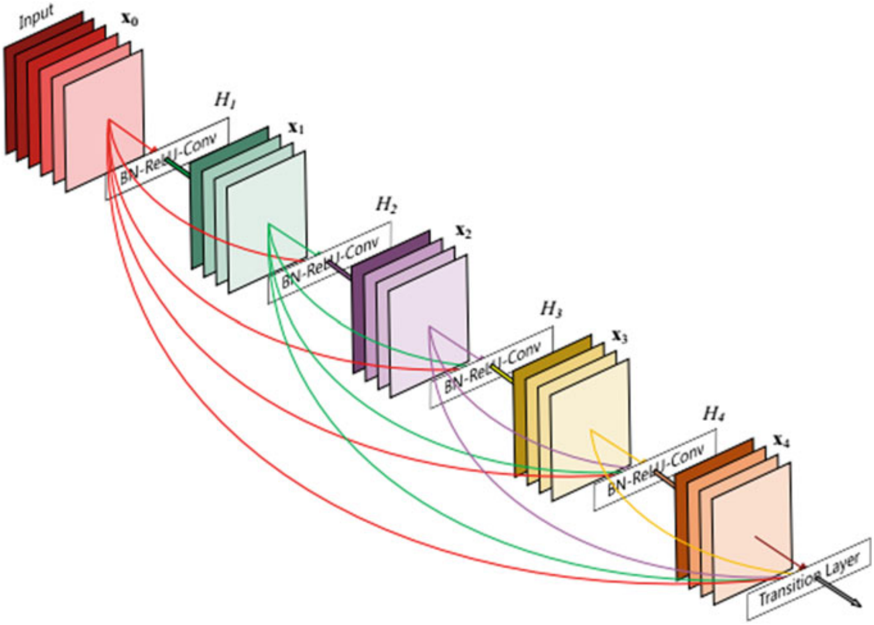


Fig. 3. DenseNet structure

2.3 Merge Texture Features

Deep neural network can directly classify images, but it does not use the underlying features of images. Considering that there are a large number of underlying features such as texture in medical images, this paper integrates the original image and texture features and then sends them into the deep neural network for classification.

2.3.1 Extract the LBP Texture Features of the Image

The local binary mode is a texture metric in the gray range. In order to improve the limitations of the original LBP, it is impossible to extract the texture features of large-size structures. Ojala et al. [7] modified the LBP to obtain the LBP value that uniquely represents the local texture features:

$$\mathrm{LBP}_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i \tag{1}$$

where $\mathrm{S}(\cdot)$ is defined as:

$$\mathrm{s(x)} = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{2}$$

2.3.2 Merging LBP Features

Performing mean variance normalization on the grayscale image and the corresponding LBP feature image, and then combining the grayscale image with the LBP image in the channel dimension, for example, the shape of the grayscale image is (h, w), the shape of the LBP image Also (h, w), the combined shape is (h, w, 2).

3 The Experiment

3.1 Experimental Dataset

MURA is a dataset of musculoskeletal radiographs, which contains a total of 14,863 studies of 12,173 patients and 40,561 multiview radiographs. Each was one of seven types of standard upper extremity radiology studies: fingers, elbows, forearms, hands, humerus, shoulders, and wrists [8].

MURA contains 9045 normal and 5818 abnormal musculoskeletal radiographic studies as shown in Table 1.

Table 1. Composition of the MURA dataset

Study	Train		Validation		Total
	Normal	Abnormal	Normal	Abnormal	
Elbow	1094	660	92	66	1912
Finger	1280	655	92	83	2110
Hand	1497	521	101	66	2185
Humerus	321	271	68	67	727
Forearm	590	287	69	64	1010
Shoulder	1364	1457	99	95	3015
Wrist	2134	1326	140	97	3697
Total No. of studies	8280	5177	661	538	14,656

3.2 The Experiment

We trained a 169 layer convolutional neural network to diagnose musculoskeletal abnormalities.

3.2.1 Parameter Settings

During training, we resize the image to 320 × 320, and we use a random horizontal flip and a random rotation of 30° for augmentation. All parameters of the network are randomly initialized. The optimizer chooses Adam, where β_1 is equal to 0.9 and β_2 is equal to 0.999. We trained the model using minibatches of size 16. The initial learning rate is chosen to be 0.00625. When iterating to the 60th and 80th, the learning rate is attenuated 10 times.

3.2.2 The Experimental Setup

To demonstrate the effectiveness of our proposed method, we conducted three sets of experiments:

- Experiment 1: Training with grayscale image
- Experiment 2: Training with grayscale + lbp (radius 1, sampling 8 points)
- Experiment 3: Training with grayscale + lbp (radius 2, sampling 8 points).

3.3 The Experimental Results

The performance of the experimental results on the validation set is shown in Fig. 4. We can see that combining LBP features when the model converges is better than using only grayscale images, which proves the effectiveness of the proposed fusion method between gray images and shallow texture features.

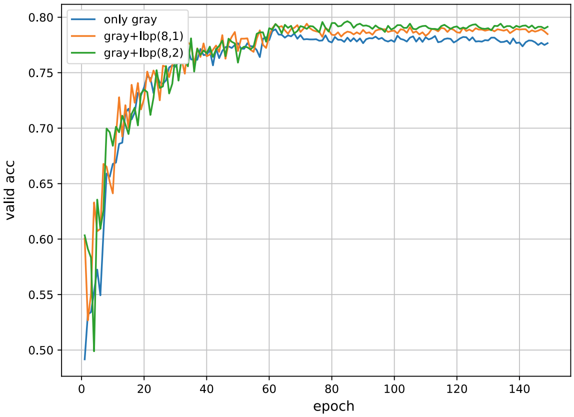


Fig. 4. The performance of the three experiments on the validation set

Table 2 shows the highest precision achieved in the validation set for Experiment 1, Experiment 2, and Experiment 3.

Table 2. The highest precision achieved by the three experiments on the validation set

Model	Only gray	Gray + LBP (8, 1)	Gray + LBP (8, 2)
Val_acc (%)	78.91	79.42	79.64

4 Conclusion

In this paper, a new classification algorithm based on deep convolutional neural network is proposed for radioactive musculoskeletal images. Adaptive histogram equalization with limited contrast was used to improve the training image quality, and the image texture features were merged and sent to DenseNet169 for training and classification. Experimental analysis shows that the proposed algorithm can effectively improve the classification accuracy of radioactive musculoskeletal images.

Further work in this paper is to continue to optimize the algorithm and expand the dataset to train a new depth neural network sensitive to radioactive musculoskeletal images.

Acknowledgements. This study was supported by the National Natural Science Foundation of China (NSFC) under Grant No. 61563039.

References

1. Qin D, Wang S, Zhang Z, He L (2017) Advances in research on discrimination criteria for work-related musculoskeletal disorders. Chin Occupational Med (3)
2. Spanhol FA, Oliveira LS, Petitjean C, Heutte L (2016) Breast cancer histopathological image classification using convolutional neural networks. In: International joint conference on neural networks, IEEE
3. Wei L, Peng C, Dazhe Z, Junbo W (2016) Pulmonary nodule classification with deep convolutional neural networks on computed tomography images. Comput Math Methods Med 2016:1–7
4. Roth HR, Lu L, Seff A, Cherry KM, Hoffman J, Wang S et al (2014) A new 2.5 D representation for lymph node detection using random sets of deep convolutional neural network observations. Med Image Comput Comput Assist Interv
5. Pizer S (1987) Adaptive histogram equalization and its variations. Comput Vis Image Process 39(3):355–368
6. Huang G, Liu Z, Laurens VDM, Weinberger KQ (2016) Densely connected convolutional networks
7. Ojala T, Pietikäinen M, Mäenpää T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans Pattern Anal & Mach Intell
8. Rajpurkar P, Irvin J, Bagul A, Ding D, Duan T, Mehta H et al (2017) Mura: large dataset for abnormality detection in musculoskeletal radiographs

An ATM Knowledge Graph-Based Method of Dynamic Information Fusion and Correlation Analysis

Qiucheng Xu(✉), Yongjie Yan, Yungang Tian, Weiyu Jiang, and Jinglei Huang

State Key Laboratory of Air Traffic Management System and Technology,
The 28th Research Institute of China Electronics Technology Group Corporation,
Nanjing 210007, China
xuqiucheng1987@163.com

Abstract. In order to fuse information and analyze correlation more efficiently and flexibly, an air traffic management (ATM) knowledge graph-based method is proposed to reorganize the information flexibly and manage the fusion process dynamically. After that, a breadth-first and depth-first search-based correlation analysis method is designed to find deeper correlations and improve the searching efficiency.

Keywords: Air traffic management · Knowledge graph · Information fusion · Correlation analysis

1 Introduction

With the rapid development of air transport in China, the number of flights is increasing rapidly at an average annual rate of over 10% in recent years [1]. It is predicted that civil aviation transportation in China will maintain a growth rate of about 12%. By 2030, there will be more than 450 civil transport airports, more than 95% of county-level administrative regions and population in China will receive air services, and the volume of passenger traffic will reach 1.8 billion [2]. However, the rapid development has also brought more and more challenges and pressures to the safe and efficient operation of air transport, for example, flight delays have become more common than ever [3], and the aviation safety accidents demonstrate a steadily rising tendency [4,5].

Air traffic management (ATM) has taken many measures to meet the challenges brought by rapid development of air traffic, such as building more surveillance and sensing equipment; developing a variety of information systems for auxiliary control aiming at different stages and scenarios of air traffic control [6]. In addition, the air traffic control department has strengthened cooperation with diverse departments, including airports, airlines, etc., to access more and more kinds of information from different sources [7]. Above solutions can solve the specific problems of air traffic controllers, while on the other hand, they could also bring a series of new problems, such as the explosion of information scale, difficulty of integration, and adding new workload to the controllers.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 487–496, 2020.
https://doi.org/10.1007/978-981-15-0187-6_58

Many researchers in the field of ATM focus on information management and efficient use methods in the operation of air traffic control.

Some researchers [8,9] introduce the concept of System Wide Information Management (SWIM). As an information switch platform based on network technology, it has been shown that relevant data can be safely, effectively, and timely shared and exchanged between different units and information systems.

Medina et al. [10] describe a usability analysis tool which computes estimate of trials-to-mastery and the probability of failure-to-complete for each task. The information required to complete a task on the automation under development is entered into the Web-based tool via a form. Yang et al. [11] look ahead the concept of intellectualized air traffic management technology, represented by deep learning, emphasizes that judgment and decision-making based on a large number of prior knowledges, which is consistent with the decision-making process of ATM.

However, most of the proposed methods put much emphasize on the breadth while not depth of information that system can handle, and thus leading to the failure to solve the depth-related problems. For example, is there relationship between flight CCA4228 and CES2471? If yes, what is the relationship? Is Shanghai Pudong Airport related to Beijing Capital Airport? If so, how is it related?

Inspired by the research idea of applying knowledge graph in social network, which can represent the breadth of a person's social relationship and analyze the depth of the relationship between any two people, we propose an information fusion and correlation analysis method based on ATM knowledge graph [12]. The main contributions of this paper are outlined as follows.

(1) Using ATM knowledge graph to reorganize the information flexibly and manage the fusion process dynamically.

(2) Based on the ATM knowledge graph, a breadth-first and depth-first search-based correlation analysis method is proposed to get all related feature instances on breadth and get relative path on depth.

Experimental results show that, compared with the traditional fusion and search method, the proposed method can find more deeper correlation and improve the searching efficiency.

The structure of the paper is organized as follows. Section 2 describes the problem definition. Section 3 shows the ATM knowledge graph-based information fusion and correlation analysis method. Section 4 shows the experimental results, and at last conclusions are made at Sect. 5.

2 Problem Definition

The ATM feature types commonly used in ATM information system include airports, runways, routes, airspace, flights, airlines, route points, control units, etc. Most of these commonly used information uses object-oriented design method, that is, to design a type template for each kind of feature, and mainly uses tables in entity-relationship databases for storage, which finally convert each data entity into a row of records.

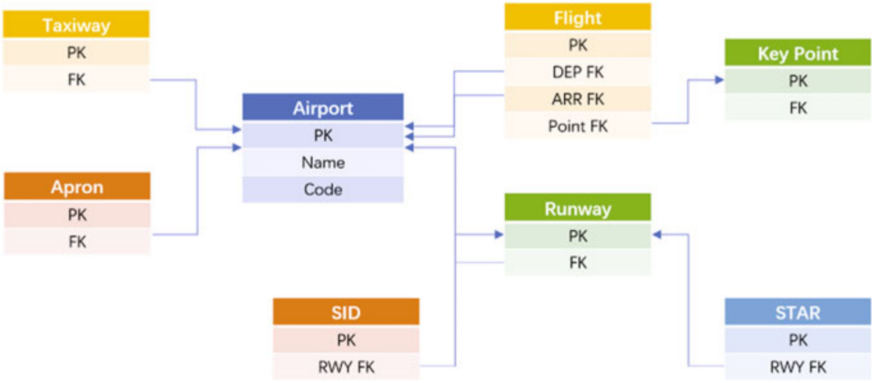


Fig. 1. Different types of ATM features are constructed as data table and connected with each other by PK and FK.

Nowadays, the commonly used method for ATM information system to construct data association and fusion is searching-related feature from multiple data tables through the form of primary key (PK) and foreign key (FK). Figure 1 shows an example of how different feature types are connected by PK and FK.

When the breadth-first search is needed to traverse all relevant data, such as "Pudong Airport in Shanghai as the center, search for all relevant information features," the following questions are asked:

(1) If FK is known, thus FK → PK, the correlation can be obtained. For example, Runway [FK] → Airport [PK], Taxiway [FK] → Airport [PK], Apron [FK] → Airport [PK].
(2) Conversely, if PK is known, then PK → ?, forward associations cannot be performed to know the FK tables who are associated directly unless all data tables are traversed.

When the depth-first search is needed to traverse the association path of two known elements, such as "Query the relationship between flight CCA4228 and Shanghai Pudong Airport," the following questions are asked:

(1) Existing processing capacity, if the take-off and landing airport of flight CCA4228 and the standby airport are not Shanghai Pudong Airport, there is no relationship between them.
(2) The real situation may have the following correlation: Shanghai Pudong Airport → Runway → SID → Route Point → Flight CCA4228, which can be interpreted as that the waypoint of flight CCA4228 at a certain time is the relevant waypoint of departure procedure at 17R_35L runway of Shanghai Pudong Airport.

Through the analysis of the two above-mentioned cases, it can be found that there existing the following shortcomings in nowadays' information fusion association methods:

(1) The traversal mode of breadth-first search is fixed and cannot be exhausted effectively;

(2) The structure of entity relationship is difficult to change dynamically;
(3) The depth-first search traversal ability is insufficient to carry out association reasoning.

3 ATM Knowledge Graph-Based Fusion and Correlation Analysis Method

In order to get all related feature objects on breadth and get relative path on depth, an ATM knowledge graph-based fusion method is proposed to reorganize the information, and after that, a breadth-first and depth-first search-based correlation analysis method is designed to achieve the required goal.

3.1 ATM Knowledge Graph-Based Fusion Method

In this work, we first build the ATM knowledge graph to reorganize all the important ATM features, such as Airport, Route, Airspace, Flight Plan, Airline, Land Mark, Control Unit, and so on. The graph data structure is chosen to represent and save the relationship between all the ATM feature instances, which is different from the E-R divided tables storage method.

The ATM knowledge graph G consists of a collection of nodes N and edges E, which is $G = (N, E)$. The nodes represent all the ATM feature instances and the edges represent the relationship between different feature instances. Figure 2 gives a partial fragment of how the ATM knowledge graph looks like.

In the graph, all the ATM features are organized together and all the relationships between each feature instance are saved as edges. With the graph, we

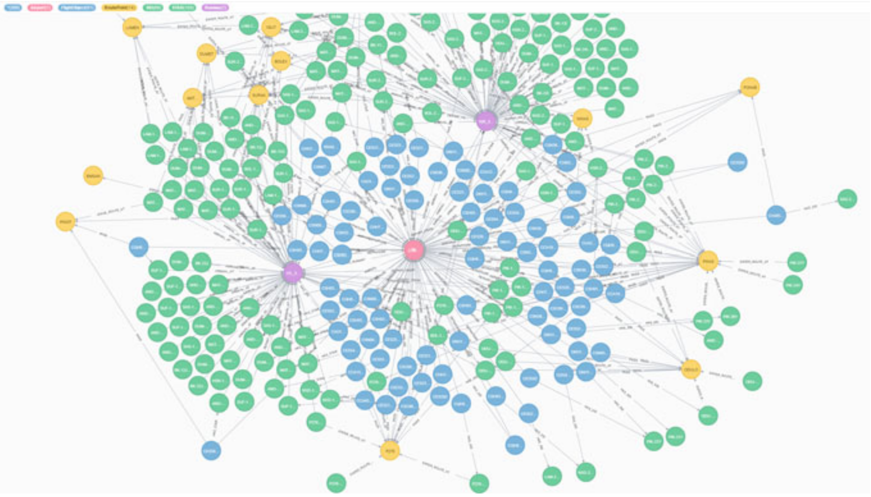


Fig. 2. A partial fragment of the ATM knowledge graph. Different colors of the nodes represent different feature type and the edge between each two nodes has its semantic meaning.

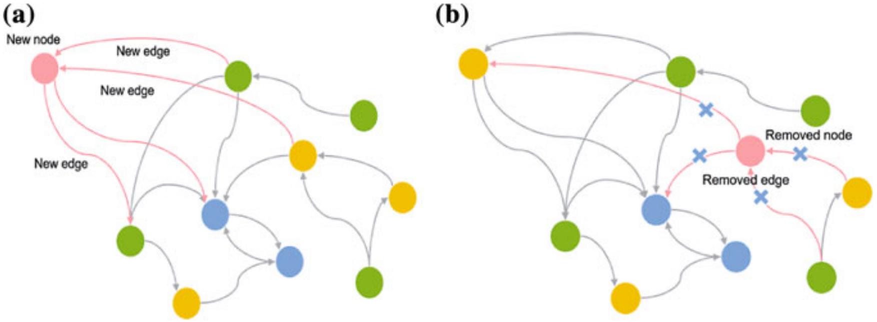


Fig. 3. The graph can be changed to fuse various kinds of data dynamically according to the need. **a** Add new node and edges. **b** Remove old node and edges.

can use degree distribution, average path length, and clustering coefficient index to describe the characteristic of the graph in the future.

Besides, we consider the graph as a dynamic change graph, as shown in Fig. 3, new nodes and edges can be added into the graph at any time. Also, old nodes can be removed from the graph along with the related edges.

3.2 Breadth-First and Depth-First Search-Based Correlation Analysis Method

Based on the ATM knowledge graph, we try to do some correlation analysis with breadth-first and depth-first search in the graph.

For the breadth-first search, we can get the relevant features of the searched feature. Algorithm 1 shows the overall process of the breadth-first search.

Algorithm 1 Breadth-first search method

```
Function BFS (center_node, distance):
Initialize a queue $q$ and a list result_list;
q.push_back(center_node);
center_node.distance = 0;
while $q$ is not empty  do
    first_node = q[0];
    if first_node.distance > distance then
        break;
    end if
    first_node.visited_flag = true;
    result_list.push_back(first_node)
    q.pop_front();
    Get all the adjacent nodes of first_node and put them in a temporary list adjacent_list;
    for each adj_node in adjacent_list  do
        if adj_node.visited_flag is not true  then
            adj_node.distance = first_node.distance + 1;
            q.push_back(adj_node);
        end if
    end for
end while
return result_list;
```

The BFS function needs two parameters, one is the searched feature node and another one is the distance to constrain the search range. Each circulation between line 5 and line 20, we take out the first node of the queue and push back its adjacent nodes at the end of the queue unless the adjacent node has not been visited. In line 7, we check the nodes distance to decide whether the search should be end. At last, we get all the features whose distance from the searched feature node is less than the given distance parameter.

For the depth-first search, we can get all the relative paths of two features. Algorithm 2 shows the overall process of the depth-first search. The search starts from the start node. The recursion function between line 4 and line 19 compares the current node with the end node recursively to find the path. At last, we get all the relative paths of the start and end features by the DFS method.

Algorithm 2 Depth-first search method

```
 1: With the input parameters start_node, end_node;
 2: Initialize a list result_list;
 3: DFS(start_node);
 4:  Function DFS (current_node):
 5: current_node.visited_flag = true;
 6: if current_node == end_node  then
 7:     result_list.push_back(current_node);
 8:     return true;
 9: end if
10: Get all the adjacent nodes of current_node and put them in a temporary list adja-
    cent_list;
11: for each adj_node in adjacent_list  do
12:     if adj_node.visited_flag is not true  then
13:         path_is_right = DFS(adj_node);
14:         if path_is_right == true  then
15:             result_list.push_back(adj_node);
16:         end if
17:     end if
18: end for
19: return false;
```

4 Experimental Results

The proposed method has been implemented in Java language on a Windows 64-bit workstation (Intel 2.2 GHz, 64 GB RAM).

In this work, we build an ATM knowledge graph which has about 20,000 feature nodes and 100,000 relationship edges.

Table 1. The breadth-first search results of different feature type.

Searched feature instance	Feature type	Distance from searched feature instance					
		1 (direct connect)		2		3	
		Feature numbers	Elapsed time (ms)	Feature numbers	Elapsed time (ms)	Feature numbers	Elapsed time (ms)
Shanghai Pudong	Airport	146	58	905	113	3873	258
CSN3173	Flight	8	33	1020	126	2959	283
A593	Route	27	4	55	6	296	36
DOGAR	Route point	84	8	413	39	3150	198

Table 2. The depth-first search results of relative path.

First feature instance	Second feature instance	Relative path numbers	Shortest length	Longest length	Search elapsed time (ms)
Shanghai Pudong airport	CSN3173	85	3	3	702
CCA4228	CES2471	15	2	6	453
Shanghai Pudong airport	Beijing Capital airport	124	2	8	1560

Table 1 shows the case of BFS results of different feature type. We test the distance (1, 2, 3) from searched feature instance to verify the efficiency of the BFS method. Distance 1 represents those feature nodes connect with searched feature node directly, distance 2 represents those nodes connect through two edges to the searched feature node, and so on. The result shows that the proposed method can find the same amount of direct-connected feature nodes more efficiently and flexibly, besides can find more deeper relative feature nodes which is hard to achieve by the traditional method.

Figure 4 shows an example of a breadth-first search result of route A593 with the distance (1, 2, 3).

Table 2 shows the case of depth-first search results of relative path. We choose any two feature instances to get their relative paths which cannot be achieved by the traditional method. The result shows that there may be more than one relative path between two feature instances, some of the paths are useful but some others are not. Further research work is needed to describe the path from semantic view.

Figure 5 shows the example of a depth-first search result of relative path between Shanghai Pudong international airport and flight CSN3173. We can

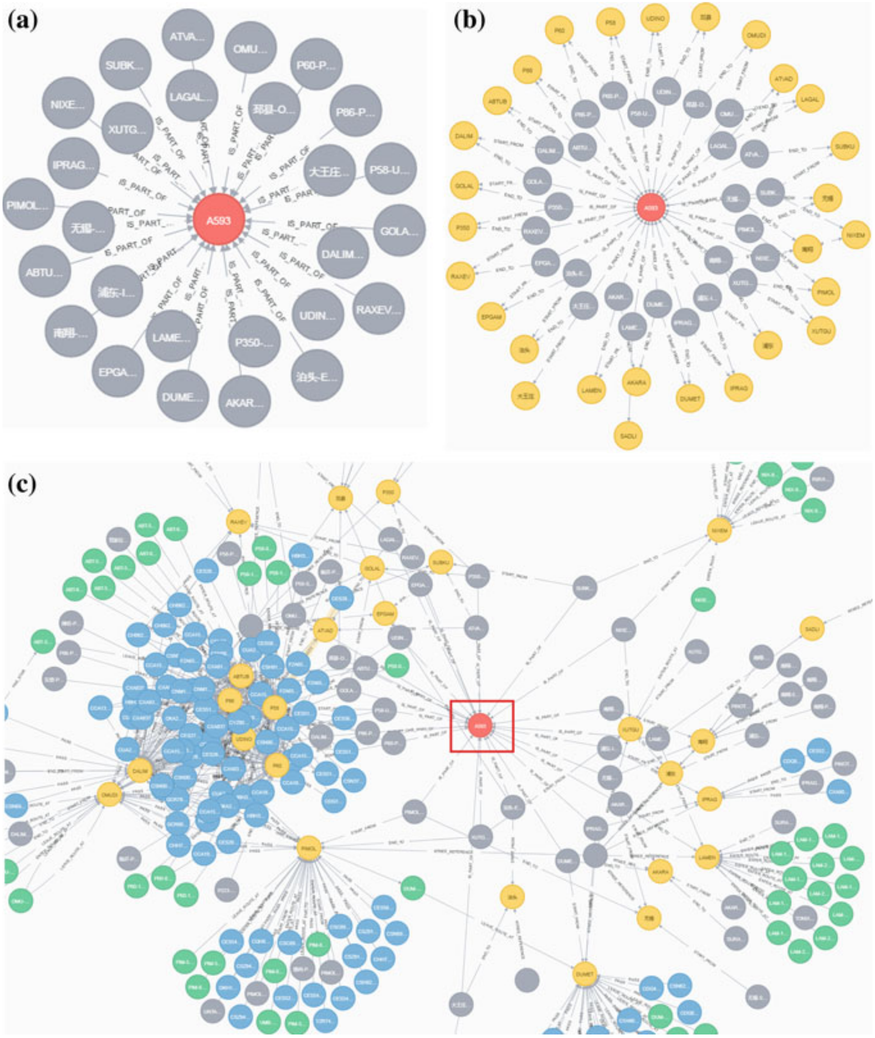


Fig. 4. Example of breadth-first search result of all features with **a** all features distance 1; **b** all features distance 2; **c** all features distance 3.

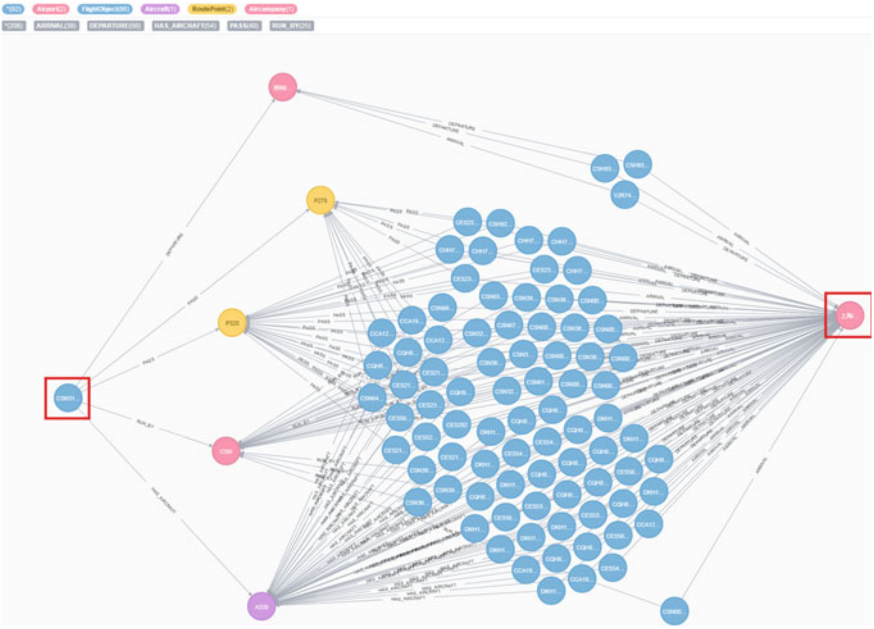


Fig. 5. A depth-first search result of relative path between Shanghai Pudong international airport and flight CSN3173.

find many different paths between them and know how they are related which is difficult to figure out by traditional methods.

5 Conclusion

This paper proposes an ATM knowledge graph-based method to reorganize the information flexibly and manage the fusion process dynamically. After that, a BFS- and DFS-based correlation analysis method is designed to find deeper correlation and improve the searching efficiency. Experimental results show that the proposed method can get all related feature instances on breadth and get relative path on depth with high efficiency and flexibility.

References

1. Zhao W (2017) The opportunities, challenges and obligations in internationalization of China civil aviation. Civil Aviat Manag 09:6–11
2. Yan Y, Cao G (2018) Operational concepts and key technologies of next generation air traffic management system. Command Inf Syst Technol 9(3):8–17
3. Dong N (2013) Current situation, cause of flight delays and governance path of China's civil aviation. J Beijing Univ Aeronaut Astronaut (Soc Sci Ed) 26(06):25–32

4. Netjasov F, Crnogorac D, Pavlovic G (2019) Potential safety occurrences as indicators of air traffic management safety performance: a network-based simulation model. Transp Res Part C: Emerg Technol 102:490–508
5. Yuan L (2108) Correlation among air traffic controllers' fatigue, workload and error. J Civil Aviat Univ China 36(06):31–33
6. Lin J (2014) Construction and development of air traffic control automation system for civil aviation. Electron Technol Softw Eng 08:255
7. Menon PK, Park SG (2016) Dynamics and control technologies in air traffic management. Annu Rev Control 42:271–284
8. Ai B (2012) New-generation ATM information system: SWIM. Chinas Civil Aviat 144(12):20–21
9. Li Q, Huang H, Chen Q (2010) SWIM—the new technology of air traffic management. Digit Commun World 05:50–53
10. Medina M, Sherry L, Feary M (2010) Automation for task analysis of next generation air traffic management systems. Transp Res Part C: Emerg Technol 18(6):921–929
11. Yang H, Yang B, Wu X et al (2018) Research and prospect of intellectualized air traffic management technology. Adv Eng Sci 04:12–21
12. Qi G, Gao H, Wu T (2017) The research advances of knowledge graph. Technol Intell Eng 3(1):4–25

Research of Lung Cancer-Assisted Diagnosis Algorithm Based on Multi-scale Convolution Kernel Network

Yunfei Li[1,2], Chuangang Li[1], Yijia Cao[1], Yue Zhao[1],
and Zhiqiong Wang[1,3,4](✉)

[1] Sino-Dutch Biomedical and Information Engineer School, Northeastern University, Shenyang, China
wangzq@bmie.neu.edu.cn
[2] School of Computer Science and Engineering, Key Laboratory of Big Data Management and Analytics (Liaoning), Northeastern University, Shenyang, China
[3] Neusoft Research of Intelligent Healthcare Technology, Co. Ltd., Shenyang, China
[4] Acoustics Science and Technology Laboratory, Harbin Engineering University, Harbin, China

Abstract. In recent years, the number of patients with lung cancer has risen steadily, becoming the first malignant tumor in men and the second malignant tumor in women. Researchers at home and abroad have found that pulmonary nodule-assisted diagnosis can detect pulmonary nodules early and effectively reduce the risk of lung cancer. Therefore, deep learning has become a new hotspot in the diagnosis of pulmonary nodules. The research content of this paper is as follows: In this paper, we extract features of lung nodules with geometric features, gray value features, texture features and use support vector machine (SVM) and extreme learning machine (ELM) to train and classify the lung nodules. The convolutional neural network (CNN) deep learning method was used to extract the features of CT images of lung nodules, to establish a characteristic model of CT images of pulmonary nodules, and to classify the benign and malignant lung nodules. This paper presents a method for computer-aided diagnosis of pulmonary nodules based on improved CNN. This method uses the convolutional neural network (CNN) to extract the features of CT images of lung nodules and establishes the feature model of CT images of pulmonary nodules. The multi-scale convolution kernel depth learning is used to prove the advancement of improved algorithms.

Keywords: CT · Lung cancer-aided diagnosis · Image processing · Feature parameters · Machine learning

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 497–505, 2020.
https://doi.org/10.1007/978-981-15-0187-6_59

1 Introduction

Cancer has a high morbidity and mortality and is one of the health concerns of widespread concern worldwide. In recent years, several cancers with high incidence rates are lung cancer, breast cancer, colon cancer, and stomach cancer. Among them, the incidence of lung cancer in the population is greater than 10%, the mortality rate is higher than the incidence rate, and the mortality rate exceeds 18%. According to the statistics of lung cancer, the incidence of lung cancer in males ranks first, and the mortality rate also ranks first. The incidence of lung cancer in women ranks second, and mortality and morbidity rank second. Therefore, how to prevent and diagnose lung cancer is one of the issues of widespread concern worldwide [1]. In recent years, computer-aided diagnosis (CAD) can make good use of these unused CT images to reduce the probability of missed diagnosis and misdiagnosis. The CAD system can use the massive CT image of the lungs in the hospital to extract the features of the image and then train the model. The CAD system can use the advantages of computers to detect and diagnose CT images of lung nodules [2].

2 Materials and Methods

2.1 *LIDC-IDRI*

The source of lung CT data used in this article is from the National Cancer Institute (NCI) LIDC-IDRI (The Lung Image Database Consortium) database. LIDC-IDRI has a total of 1018 research examples. For each of the images in the example, there are four experienced doctors who diagnose and label the two phases. In the first stage, four doctors independently diagnosed and marked the lesions. The labels are divided into three categories: nodules larger than or equal to 3 mm in diameter, nodules smaller than 3 mm, and pulmonary nodules greater than or equal to 3 mm. In the second phase, each doctor needs to review the annotations given by several other doctors and give their own diagnosis. The diagnosis of these two stages can minimize the misdiagnosis caused by irresistible factors [3].

The image file format in the database is an image file. DICOM is one of the most common format for medical images, which is shown in Fig.1.

Before using the algorithm to extract the features of the lung CT image, it is necessary to analyze the lung CT image, determine the location of the pulmonary nodule, segment the lung nodule image, and perform corresponding preprocessing on the lung nodule image. Since the focus of this study is on computer-aided diagnosis of pulmonary nodules rather than computerized detection of pulmonary nodules, this paper uses the lung nodule contour coordinates given by the LIDC-IDRI database to segment the lung nodule data [4] which is shown in Fig. 2.

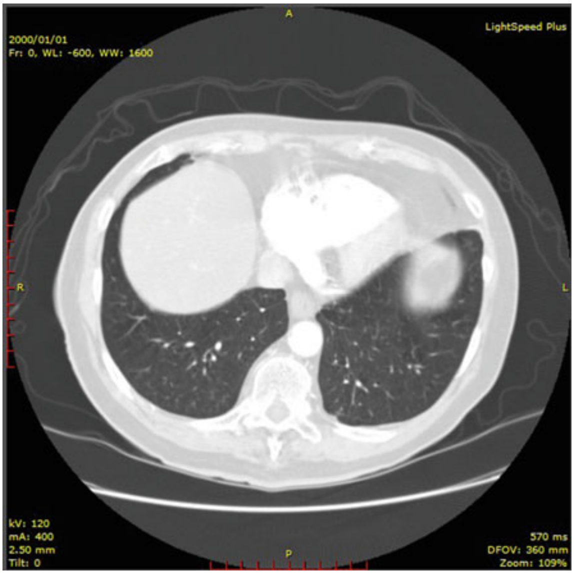


Fig. 1. Breast CT image

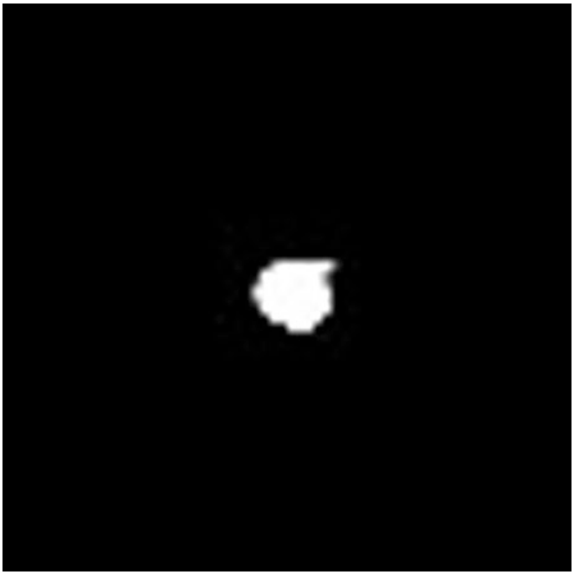


Fig. 2. Lung nodule image

2.2 *Machine Learning in CAD*

In the field of computer-aided diagnosis on lung nodule detection, there are many good methods on extraction of features and classification for benign and malignant nodules. In this chapter, we propose to select representative eigenvalues to classify pulmonary nodules based on the common characteristic parameters of pulmonary nodules. This process directly determines the data integrity and accuracy of benign and malignant discrimination [5].

The support vector machine (SVM) is based on linear separability and is gradually extended to a linearly inseparable case. SVM is essentially a nonlinear mapping, which transforms the nonlinear problem in the original sample space into a linear separable problem in the feature space [6]. The SVM applies the kernel function expansion theorem without having to know the explicit expression of the nonlinear mapping. With the help of the kernel function, the linear learning machine in the high-dimensional feature space does not increase the computational complexity.

The extreme learning machine (ELM) is a fast single hidden layer neural network (SLFN) training algorithm. When using the ELM algorithm, only the number of hidden layer neurons needs to be set, and the output weight needs to solve the Moore-Penrose generalized inverse problem of a matrix. According to this feature, no iteration is required to determine the parameters [7], thus reducing the overall tuning time. Therefore, compared with the traditional training methods of machine learning, ELM is fast and generalized.

2.3 *Deep Learning in CAD*

Convolutional neural network (CNN) is a neural network with a multilayer structure, typically model shown as Fig. 3. Each level of CNN contains one or more neurons, and the relationships between these neurons are independent of each other. The performance of the CNN algorithm depends on two factors, one is the performance of the computer hardware, and the other is a large number of image data sets. When using the traditional method for the diagnosis of pulmonary nodules [8], the chest CT object needs to be analyzed to find the CT image of the chest corresponding to the pulmonary nodules, and the lung nodule area of the chest CT image is marked; then, the characteristics of the lung nodule area are extracted, and the extracted features have the degree of solidity, and the center distance, texture, and density are used to model the features. Finally, a large number of feature models are used to train the classifier to obtain the final classification result. The use of CNN in these situations typically produces a more accurate classification than the handcrafted features of a standard classifier, and CNN learns the most meaningful features of a given task itself. The convolutional layer in the CNN may also be referred to as a feature extraction layer, that is, a C layer [9]. The pooling layer in the CNN may also be referred to as a downsampling layer, an S layer. CNN network has strong earning ability, mainly reflected in its multiple hidden layers. The inclusion of a large number of neurons in multiple hidden layers enables the CNN to process a large number of data sets, and the downsampling operation can assist the CNN in reducing the complexity of training. The structure of the traditional neural network is different from the structure of the

CNN. All nodes of each input layer and all nodes of the hidden layer are connected to each other. All nodes of the hidden layer and all nodes of the output layer are also connected to each other [10].

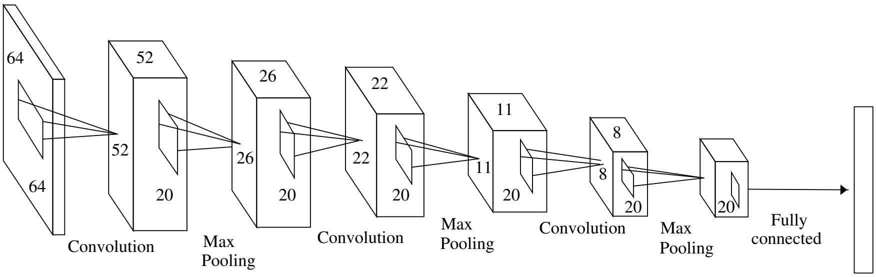


Fig. 3. Architecture of lung nodules diagnosis based on CNN

Moreover, when calculating the output of the neural network, the scale of the matrix to be calculated is relatively large. When the scale of the matrix is large to a certain extent, the convergence time is too long, and it is easy to fall into the local extreme value. In response to these shortcomings, the convolutional neural network (CNN) uses local perceptual fields and weight sharing to shorten training time and improve training accuracy.

Improved convolution neural network method. Local perception field is one of the methods to effectively reduce the number of weights and the number of parameters without reducing the accuracy [11]. It is generally believed that the human brain's perception of the real world is a process from the local to the global. For the discussion of images, you can follow this process. The condition in the image that best describes the relationship of pixel points is spatial location of the image.

In this research, we proposed an improved method in convolutional neural network, for extending the local perception field by using multi-scale convolution kernels as shown in Fig. 4.

The improved kernels are based on shapes of 1 × 1, 3 × 3, 5 × 5. Later two are similar to those working in CNN, and the 1 × 1 kernel works most importantly. The 1 × 1 convolution kernel mainly performs the function of full-join calculation through the convolution operation, thereby extracting features at a higher scale. When a convolutional layer enters a large number of features, convolution of this input will generate a huge amount of computation; if the input is first performed, the number of convolutions will be significantly reduced after reducing the number of features. Then, we need to solve the matter of information combination. Different characters are produced from different scales kernel, these branches are convolved or merged with convolution kernels of different sizes and finally stitched together in the feature dimension. More feature richness also means more accurate judgment in the final classification [12] which is shown in Fig. 5.

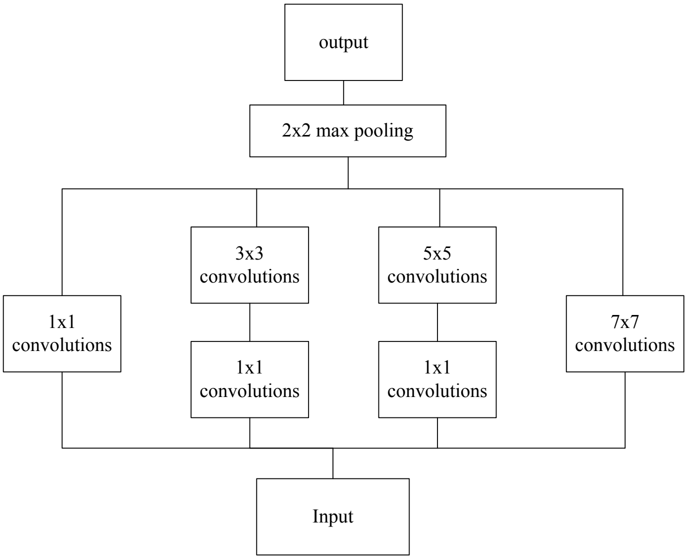


Fig. 4. Modified multi-scale convolution kernel modular diagram

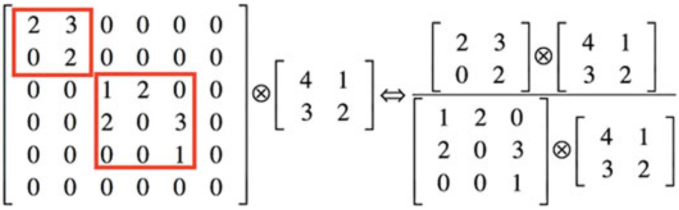


Fig. 5. Schematic diagram of sparse matrix operation

The core component of the BP algorithm is the backpropagation algorithm. The BP algorithm is based on a stochastic gradient descent strategy and adjusts the parameters according to the negative direction of the gradient of the error [13]. The derivative of layer between hidden layer and output layer matrix is shown as (1)

$$\Delta w_{hj} = -\eta \frac{\partial E_k}{\partial w_{hj}} \tag{1}$$

By observing Formula (1), it can be found that the input value corresponding to the j output neuron is affected $\hat{y}_j^k$ first and then responding to its output value E_k, and finally affecting the error, according to the chain derivation rule [14]:

$$\frac{\partial E_k}{\partial w_{hj}} = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} \cdot \frac{\partial \beta_j}{\partial w_{hj}} \tag{2}$$

The special property of activation function sigmoid, $f'(\mathrm{x}) = f(\mathrm{x})(1 * f(\mathrm{x}))$, the derivative of the output neuron is:

$$g_j = \frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} = \hat{y}_j^k \left(1 - \hat{y}_j^k\right)\left(y_j^k - \hat{y}_j^k\right) \tag{3}$$

So, we can get the derivative of the output neuron.

3 Results and Discussion

This chapter focuses on the experimental design of the auxiliary diagnosis function of benign and malignant pulmonary nodules and verifies its performance through experiments. First, this paper uses two deep learning methods for lung nodule CT data. Seven methods were used to compare CNN-based pulmonary nodule-assisted diagnosis with improved lung nodule-assisted diagnosis of CNN. Then, the SVM and ELM-based nodule-assisted diagnosis method were compared to classify the CT data of lung nodules. Finally, the advantages of improved CNN algorithm with higher accuracy and lower algorithm difficulty are obtained.

3.1 *Results*

The experimental data used was the LIDC-IDRI database, which contained a CTG image of the 124G lung. It contains 1018 cases; each case is stored in a separate folder, and each folder has about 100–500 pictures, and the folder records the results of the doctor's diagnosis, the benign and malignant lung nodules, contour coordinates, and other information. Based on the relevant file information, 1067 sets of data can be segmented from the LIDC-IDRI database. We used seven types of evaluation index to show the capabilities of four methods, which are shown in Table 1 and Fig. 6.

Table 1. Experimental evaluation value

	ELM	SVM	CNN	OURS
trainTm (ms)	41.2	62.3	81,057	31,300
testTm (ms)	5.2	5.9	23,945	2410
trainAcc (%)	71.24	74.12	76.35	85.75
testAcc (%)	69.91	72.32	74.15	79.25
Sn (%)	62.35	61.28	78.54	79.28
Sp (%)	67.32	70.13	71.34	77.36
AUC	0.6834	0.7119	0.7412	0.8451

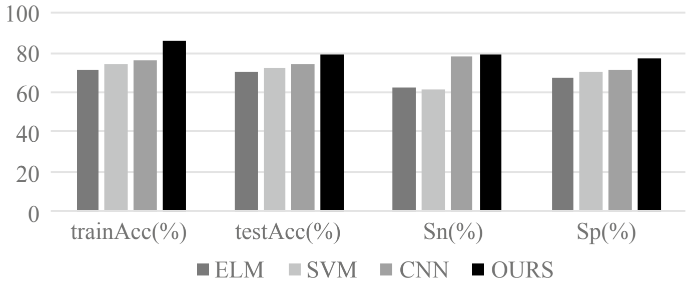


Fig. 6. Performance of ELM, SVM, CNN, and OURS CAD accuracy

3.2 *Discussion*

First, this paper uses two deep learning methods for lung nodule CT data. Seven methods were used to compare CNN-based pulmonary nodule-assisted diagnosis with improved lung nodule-assisted diagnosis of CNN. Then, the classification of lung nodules CT data based on ELM and SVM-based pulmonary nodule-assisted diagnosis was compared, and four experimental indicators were used to evaluate the two methods. Finally, the advantages of improved CNN algorithm with higher accuracy and lower algorithm difficulty are obtained, which proves that the classification accuracy of diagnosis is better.

Acknowledgements. This research was partially supported by the National Natural Science Foundation of China under Grant Nos. 61472069, 61402089 and U1401256, the China Postdoctoral Science Foundation under Grant Nos. 2019T120216 and 2018M641705, the Fundamental Research Funds for the Central Universities under Grant Nos. N161602003, N180408019 and N180101028, the Open Program of Neusoft Institute of Intelligent Healthcare Technology, Co. Ltd. under Grant No. NIMRIOP1802, and the fund of Acoustics Science and Technology Laboratory.

References

1. Gurcan MN, Sahiner B, Petrick N et al (2002) Lung nodule detection on thoracic computed tomography images: preliminary evaluation of a computer-aided diagnosis system. Med Phys 29(11):2552–2558
2. Zhang X, Hoffman EA, Sonka M (2005) A complete CAD system for pulmonary nodule detection in high resolution CT images. Proc SPIE- Int Soc Opt Eng 5747:85–96
3. Ye X, Lin X, Dehmeshki J et al (2009) Shape-based computer-aided detection of lung nodules in thoracic CT images. IEEE Trans Bio-Med Eng 56(7):1810–1820
4. Qiao G, Hong Y, Liu T et al (2013) Automated pulmonary nodule detection system in computed tomography images: a hierarchical block classification approach. Entropy 15 (2):507–523
5. Setio AA, Ciompi F, Litjens G et al (2016) Pulmonary nodule detection in CT images: false positive reduction using multi-view convolutional networks. IEEE Trans Med Imaging 35 (5):1160–1169

6. Liu X, Hou F, Qin H et al (2017) A CADe system for nodule detection in thoracic CT images based on artificial neural network. Sci China Inf Sci 60(7):177–194
7. Huang GB, Ding X, Zhou H (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74(1–3):155–163
8. Serre D (2002) Matrices: theory and applications. Mathematics, 32, xvi, 221
9. Huang GB, Song S, You K et al (2015) Trends in extreme learning machines: a review. Neural Netw Off J Int Neural Netw Soc 61(2015): 32–48
10. Campadelli P, Casiraghi E, Valentini G (2005) Support vector machines for candidate nodules classification. Neurocomputing 68(1):281–288
11. MacMahon H (2000) Improvement in detection of pulmonary nodules: digital image processing and computer-aided diagnosis. Radiographics 20(4):1169–1177
12. Schilham AMR, Ginneken BV, Loog M (2003) Multi-scale nodule detection in chest radiographs. In: Medical image computing & computer-assisted intervention-MICCAI, international conference, Montréal, Canada, pp 602–609
13. Yoshida H (2004) Local contra lateral subtraction based on bilateral symmetry of lung for reduction of false positives in computerized detection of pulmonary nodules. IEEE Trans Biomed Eng 51(5):778–789
14. Campadelli P, Casiraghi E, Columbano S (2006) Lung segmentation and nodule detection in posterior anterior chest radiographs. IEEE Trans Med Imaging 25(12):1588–1603

Design for Interference Suppression Intelligent Controller Based on Minimax Theory

Ling Chang[1,2](✉), Chaoying Shan[1,2], and Huifang Wei[1]

[1] Information & Control Engineering Faculty, Shenyang Urban Construction University, Shenyang 110167, China
changling0321@163.com
[2] Information & Control Engineering Faculty, Northeastern University, Shenyang 110004, China

Abstract. This paper focuses on modifying generalized Hamilton system-based minimax theory for solving the interference suppression of multi-machine power systems with coupling performance. The system determines the interference suppression controller and reduces the conservatism of the traditional method by calculating the worst interference degree. The results of simulation show that this method and control strategy can make the system state converge to the initial equilibrium point rapidly under the influence of large disturbance to effectively improve the transient stability performance of power system.

Keywords: Hamilton · Minimax theory · Multi-machine power systems

1 Introduction

With the development trend of the power system with large scale, it is very important to study the stability and anti-interference ability of multi-machine power system. With the further study of power system by scholars, the robust control of power system by Hamilton theory is well applied and realized. Literature [1] gives the sufficient conditions which can generalize dissipative Hamilton to the simple power system. Then, the Lyapunov function is given based on the energy function which obtains the local dissipative Hamilton implementation of dual-machine system. Literature [2] proposed an improved Hamilton system for the Hamilton implementation problem of NDAS. The energy shaping of H function was carried out by reconstructing the system structure matrix, and the design method of stabilization controller was presented. Literature [3, 4] combines the generalized Hamilton energy theory to design the power system controller. Literature [5] establishes the generalized Hamilton model of multi-machine power system with two cases; the first case does not consider the transfer conductance while the other case considers both self-admittance and mutual admittance. But it is without considering the mutual conductance, which is considering that G_{ij} is equal to 0. Literature [6] provides the improved Hamilton system structure considering mutual conductance, which is called pseudo-generalized Hamilton system. Many scholars have

The National Natural Science Foundation of China (Grant No. 61773108).

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 506–513, 2020.
https://doi.org/10.1007/978-981-15-0187-6_60

achieved good results in interference suppression and adaptive control of multi-machine power system. However, in dealing with interference problems, it increases the conservatism of the system by reducing inequality.

The research findings of the author show the minimax theoretical control method has a good effect on suppressing sudden large interference [7, 8], and it is mature for the research on interference suppression of single-machine power system. Literature [9] for TCSC single-machine infinite power system adopts the method of combining with minimax and backstepping, puts forward a kind of effective output feedback control method, and studies the robust stabilization problem of grid systems; literature [10] imports minimax theory into the dissipative Hamilton system and designs the interference suppression controller for the TCSC single-machine infinite power system.

This paper focuses on modifying generalized Hamilton system-based minimax theory for solving the interference suppression of multi-machine power systems. Minimax method is used to deal with the interference item of the system, and it estimates accurately the inflection of maximum interference on the system. Through the construction of the index function and performance metrics, the conservatism of traditional interference processing methods has been improved.

2 Preliminary Knowledge

2.1 Pseudo-generalized Hamilton

According to reference [11, 12], the general affine nonlinear dynamic system is as follows:

$$\dot{x} = f(x) + g(x)u \tag{1}$$

Definition 1 If there is a Hamilton function $H(x)$, system (1) can be presented as follows:

$$\dot{x} = M(x)\partial_x H(x) + p(x) + g(x)u \tag{2}$$

It is called that system (2) possesses the implementation of pseudo-generalized Hamilton, where $H(x)$ is the generalized Hamilton function of system (2).

In which $\partial_x H(x) = [\partial_{x_1} H, \partial_{x_2} H \ldots \partial_{x_n} H]^{\mathrm{T}}$, $p(x) = (p_1, p_{2,} \ldots p_{n,})^{\mathrm{T}} \in R^n$, and $M(x) = (M_{ij}(x))_{n\times n}$ are the structure matrix. If $M(x)$ can be described as $M(x) = J(x) - R(x)$, and $J(x)$ is the antisymmetric matrix, $R(x)$ is the positive semi-definite matrix, then function (2) is the implementation of strict pseudo-generalized dissipative Hamilton.

If $M(x)^{-1}$ is the non-singular matrix and it has the inverse matrix, we can obtain the expression as follows:

$$\bar{p}(x) = M(x)^{-1} p(x) \tag{3}$$

Lemma 1 If

$$\dot{x} = M(x)(\partial_x H(x) + \bar{p}(x)) + g(x)u \tag{4}$$

There is an energy function as follows:

$$W(x) = H(x) + \sum_{k=1}^{n} \int_{x_{k0}}^{x_k} \xi_k(x_k)dx_k - H(x_0) \tag{5}$$

In which $\xi_k(x_k) = \bar{p}_k(x_{10}, x_{20}, \ldots, x_{k-1}, x_k, x_{k+1,0} \ldots, x_{n0})$

Then, function (2) can be rewritten as follows:

$$\dot{x} = M(x)\partial_x W(x) + \eta(x) + g(x)u \tag{6}$$

where $\eta(x) = p(x) - M(x)(\xi_1(x_1), \xi_2(x_2), \ldots, \xi_n(x_n))^{\mathrm{T}}$.

Then, we have

$$\partial_x W(x_0) = \partial_x H(x_0)\bar{p}(x_0) = 0 \tag{7}$$

Hence, x_0 is one stagnation point of the energy function $W(x)$.

2.2 Interference Suppression Based on Minimax Theory

This section calculates the maximum critical interference degree that system can be withstood by the constructed test function before adopting interference suppression controller with minimax theory. The interference degree depends on the state and changes of system input that is not a simple estimate. Then it focuses on the maximum disruption interference of the system can be withstand to design the interference suppression controller, in order to insure that the system has semi-positive energy storage function $W(x)$ and for any given interference suppression constant, which make the dissipation inequality hold.

$$\dot{W} + Q(x) \leq \frac{1}{2}\left\{\gamma^2\|\omega\|^2 - \|Z\|^2\right\} \tag{8}$$

Lemma 2 For the pseudo-generalized Hamilton system

$$\begin{cases} \dot{x} = M(x)\frac{\partial W(x)}{\partial x} + g(x)u + G\omega \\ z = h(x)g^T\frac{\partial W(x)}{\partial x} \end{cases} \tag{9}$$

According to select the test function as follows:

$$\Psi = \dot{W} + \frac{1}{2}\left(\|z\|^2 - \gamma^2\|\omega\|^2\right) \tag{10}$$

The maximum disruption interference can be calculated as follows:

$$\omega^{*} = \frac{1}{\gamma^{2}} G^{\mathrm{T}} (\nabla W) \tag{11}$$

3 The Design of Multi-machine Minimax Interference Suppression Controller

$$\begin{aligned}
&\dot{\delta}_i = \omega_s \omega_i \\
&\dot{\omega}_i = \frac{1}{M_i}\left\{ P_{mi} - D_i\omega_i - E'_{qi}\sum_{j=1}^{n} E'_{qj}(B_{ij}\sin\delta_{ij} + G_{ij}\cos\delta_{ij}) \right\} + \varepsilon_{i1} \\
&\dot{E}'_{qi} = \frac{1}{T'_{\mathrm{doi}}}\left\{ -b_i E'_{qi} - (X_{di} - X'_{di})\sum_{j=1}^{n} E'_{qj}(G_{ij}\sin\delta_{ij} - B_{ij}\cos\delta_{ij}) + E_{\mathrm{fdsi}} + u_{fi} \right\} + \varepsilon_{i2}
\end{aligned} \tag{12}$$

In the function (12), $b_i = 1 - (X_{di} - X'_{di})B_{ii}$; $c_i = X_{di} - X'_{di}$, where ω_i is the generator angular velocity; ω_s is the synchronous angular velocity; X_{di} is the d-axis synchronous reactance of generator; X'_{di} is the d-axis transient state reactance; T'_{doi} is the d-axis open-circuit time constant; M_i is the generator inertia time constant; D_i is the damping coefficient; P_{mi} is the mechanical power; E'_{qi} is the p-axis transient state potential; E_{fdsi} is the exciting voltage for system steady state operation; u_{fi} is the exciting input voltage of generator; ε_{i1} and ε_{i2} are bounded disturbances, and they represent the mechanical power disturbance and the field winding parameter disturbance, respectively.

We select Hamilton function as follows:

$$\begin{aligned}
H(\delta, \omega, E'_q) = {} & \frac{1}{2}\sum_{i=1}^{n} M_i \omega_s \omega_i^2 + \sum_{i=1}^{n} P'_{mi}(\delta_{is} - \delta_i) + \sum_{i=1}^{n} \frac{b_i}{2c_i}\left(E'_{qi} - \frac{1}{b_i}E'_{\mathrm{fdsi}}\right)^2 \\
& + \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1, j\neq i}^{n} \left(E'_{qis}E'_{qjs}B_{ij}\cos\delta_{ijs} - E'_{qi}E'_{qj}B_{ij}\cos\delta_{ij}\right)
\end{aligned} \tag{13}$$

According to Lemma 1, we select energy function as follows:

$$W(x) = H(x) + \sum_{i=1}^{n}\sum_{k=1}^{3}\int_{x_{k0}}^{x_k} \xi_{ik}(x_k)dx_k - H(x_0) \tag{14}$$

The form of constructed pseudo-generalized Hamiltonian is

$$\dot{x}_i = [J_i - R_i]\frac{\partial W(x)}{\partial x_i} + g_i \hat{u}_{fi} + \eta_i + \tilde{g}_i \varepsilon_i \tag{15}$$

where $J_i = \begin{bmatrix} 0 & \frac{1}{M_i} & 0 \\ -\frac{1}{M_i} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $R_i = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{D_i}{M_i^2 \omega_s} & 0 \\ 0 & 0 & \frac{c_i}{T'_{\text{doi}}} \end{bmatrix}$, $g_i = \begin{bmatrix} 0 & 0 & \frac{1}{T'_{\text{doi}}} \end{bmatrix}^{\text{T}}$, $\tilde{g}_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\text{T}}$, $\varepsilon_i = \begin{bmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \end{bmatrix}$, η_i is the bounded parameter.

Assumed that

$$G_i = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}^{\text{T}}, \omega_i = \begin{bmatrix} \omega_{i1} \\ \omega_{i2} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} (\eta_i + \tilde{g}_i \varepsilon_i) \tag{16}$$

Then, function (15) can be written as follows:

$$\dot{x}_i = [J_i - R_i]\frac{\partial W(x)}{\partial x_i} + g_i(x_i)\hat{u}_{fi} + G_i \omega_i \tag{17}$$

where $G_i \omega_i = \eta_i(x) + \tilde{\text{g}}_i \varepsilon_i$.

And ω_i includes the influence of the uncertain perturbation term and the bounded vector in the system, and it can be regarded as a new bounded disturbance vector.

According to Lemma 2, take the maximum disruption interference into (17), then the test function can be described as follows:

$$\Psi = -\nabla^{\text{T}} W \cdot R \cdot \nabla W + \nabla^{\text{T}} W \cdot g(x)\widehat{u} + \frac{1}{2}\nabla^{\text{T}} W \cdot g h^{\text{T}} h g^{\text{T}} \nabla W - \frac{1}{2\gamma^2}\nabla^{\text{T}} W \cdot G G^{\text{T}} \nabla W \tag{18}$$

If the system still keeps stable under the effect of the worst interference degree, then the controller can be derived as follows:

$$\hat{u}_{fi} = -\frac{1}{2} h_i^{\text{T}} h_i g_i^{\text{T}} \frac{\partial W(x)}{\partial x_i} \tag{19}$$

where $R + \frac{1}{2\gamma^2} G G^T > 0$, $Q(x) = R + \frac{1}{2\gamma^2} G G^T$.

If the interference of the system becomes 0, the system can be asymptotically stable at equilibrium; if the interference is uncertain, it will meet the performance index (9). Hence, on the basis of any bounded disturbance, the controller can be designed as follows:

$$
\begin{aligned}
u_{fi} = -\frac{1}{2}T'_{\text{doi}}h_i(x)h_i^T(x)\Bigg[\frac{b_i}{c_i}(E'_{qi} - \frac{1}{b_i}E'_{\text{fdsi}}) - \sum_{i \neq j}^{n} E'_{qj}B_{ij}\cos\delta_{ij} \\
+ c_i \sum_{1 \le k \le i} E'_{qk}G_{ik}(\sin\delta_{ik} - \sin\delta_{\text{isks}}) - E''_{\text{fdsi}}\Bigg]
\end{aligned} \tag{20}
$$

Theorem 3 For any given $\gamma > 0$, there is an energy function $W(x)$ and $Q(x)$ for system (12),through the controller (20) effect with the worst interference condition, it meets the dissipative inequality (8). The system will have the characteristic of interference suppression and asymptotically stable at the equilibrium point.

According to Minimax theory, the performance index is established for system (12). We can calculate the maximum interference degree of the system and withstand on the base of the test function. Then, we can solve the conservatism caused by inequality reduction, further accurately design the interference suppression controller.

4 Simulation Results

For verifying the effectiveness of the method, the three-machine power system shown in Fig. 1 will be simulated as below. The generator node 3 in the figure is the reference node, and the system network parameters are detailed in the reference literature [1]. It is assumed that a three-phase short-circuit fault will occur between generator busbar 1 and busbar 2 at 2 s, the relay protection operated will disconnect the fault line at 2.1 s, when the trouble remove at 3 s, and the automatic reclosing will lock. The simulation results are shown in Figs. 2, 3, and 4.

From the above simulation results, it can be seen that the curve has a fluctuation when the system is disturbed at 2 s because of the equilibrium state is greatly different from stable state. When the system is trouble shooting and working normally at 3 s, the response curve can converge to the equilibrium points rapidly, and the effectiveness of the method has been proved.

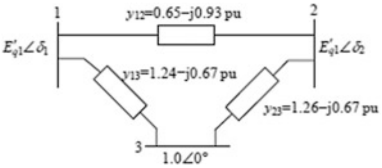


Fig. 1. The simplify structure of three machine power system

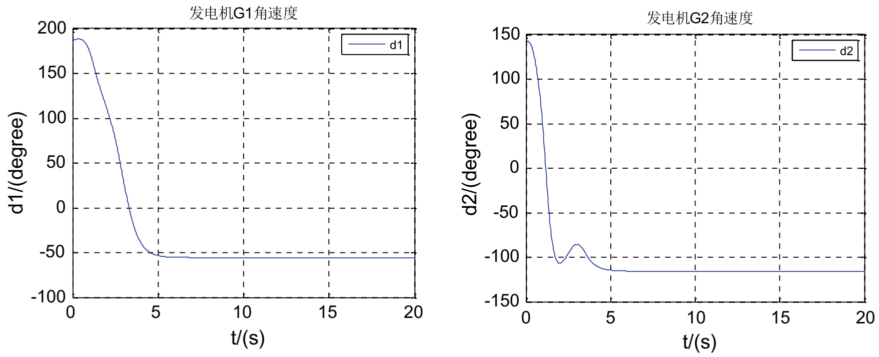


Fig. 2. The response curve of generator power angle

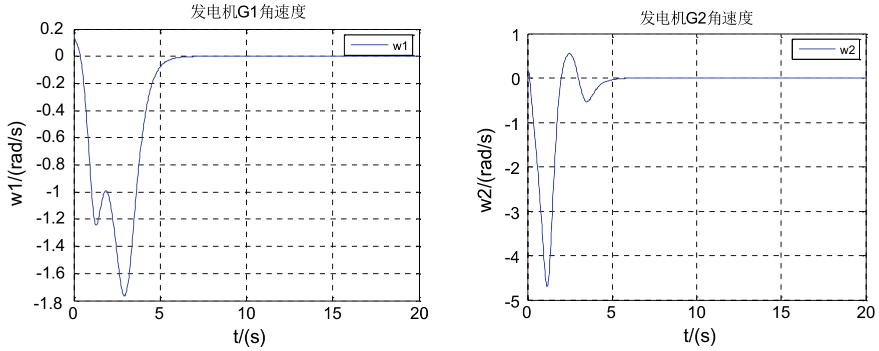


Fig. 3. The response curve of generator angular velocity

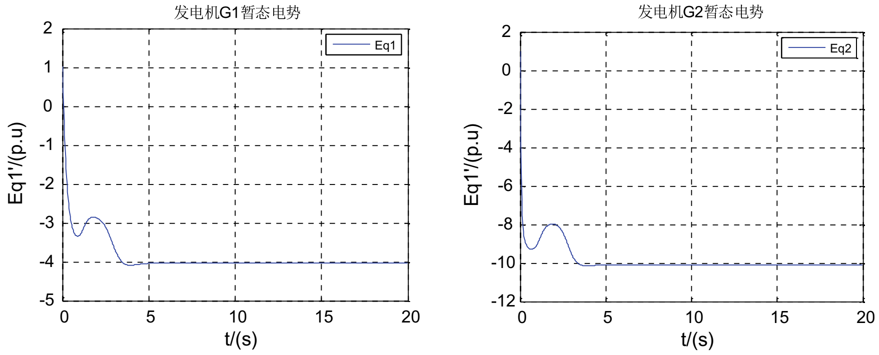


Fig. 4. The response curve of transient electromotive force

5 Conclusion

In this paper, a new technique based on Hamilton theory has been further studied the multi-machine power system interference suppression. Combining minimax theory with Hamilton theory, it calculates the worst disturbance degree when dealing with the interference. The method has solved the problem of the system parameter hypothesis and reduces the conservatism. Simulation experiments show that the control strategy has an effective interference suppression effect.

References

1. Wang YZ (2007) Generalized Hamilton control system theory. implementation, control and application. Science Press 112, 556
2. Lu DI, Zhang XB (2011) Based on the Hamilton energy plastic multi-machine power system excitation control. Power Syst Protec Control 45–50
3. Liu MX, Wang J (2007) Nonlinear control of multi-machine power system with structure preserving models based on Hamiltonian system energy theory. Proc CSEE 28:34
4. Sun YZ, Cao M, Shen TL et al (2002) Decentralized passive control of turbo-generators based on the generalized Hamiltonian system theory. Autom Electr Power Syst
5. Liu MX (2008) Multi-machine power system excitation and the generalized Hamilton realize control method and application. Shanghai Jiao Tong University
6. Shi F, Wang J (2011) Pseudo generalized Hamiltonian theory and its application in multi-machine power system nonlinear excitation control. Proc CSEE 67:74
7. Jiang N, Jing YW (2008) Based on t-s model not fragile minimax control of nonlinear systems. Control Theory Appl 925–928
8. Jiang N, Liu T, Tie T et al (2012) The nonlinear generator excitation system interference suppression controller design. In: Proceedings of the 31st chinese control conference
9. Jiang N, Liu B, Jing YW (2011) The design of nonlinear disturbance attenuation controller for TCSC robust model of power system. Nonlinear Dyn 1863:1870
10. Liu T et al (2013) The TCSC system adaptive minimax interference suppression controller design. Control Decis Mak 1894:1895
11. Sun YZ et al (2007) The nonlinear robust control in power systems. Tsinghua University Press
12. Chang L et al (2015) Excitation controller design for nonlinear multi-machine power system based on Hamilton theory. In: 27th chinese control and decision conference, pp 1362–1366

Research on System Architecture Based on Deep Learning Convolutional Neural Network

Caijuan Chen and Ying Tong(✉)

Tianjin Key Laboratory of Wireless Mobile Communications and Power Transmission, Tianjin Normal University, Tianjin, China
tongying2334@163.com

Abstract. In recent years, because deep learning technology can effectively learn features from data, it has become a powerful technical means in the field of image recognition. Research on image recognition can better promote the development of artificial intelligence and computer vision. This paper has conducted research and review of this field, introduced its important development and application, and made an attempt to promote further development in this field. Firstly, this paper introduces the structure of the network, and then introduces the common structural model of deep learning with CNN. The technical methods of reducing overfitting method, neural network visualization technology, inception structure, and transforming input images are discussed. Finally, the problems that still need to be solved in this field and the future of deep learning are introduced. It is pointed out that distributed computing, bit number reduction, migration learning, image style transformation, image generation, etc., are further research directions in the field of image recognition.

Keywords: Deep learning · Image recognition · Convolutional neural network

1 Introduction

The wave of deep learning is surging, and related literature books are emerging one after another. The world of science fiction movies has become a reality. Artificial intelligence has defeated the Go champion, and mobile phones can not only understand people's speech (such as Microsoft's Xiao Bing, Apple's Siri), but also real-time translation in video calls, the generalization of autonomous driving technology is just around the corner. Artificial intelligence is changing our lives and gradually making our world a new world. Such an amazing speed of development is inseparable from deep learning technology. Deep learning has shown excellent performance in various fields such as graphics, voice, and natural language. Image recognition technology based on deep learning has important applications in weather forecasting, medical imaging, navigation, and environmental monitoring. This paper has conducted research and review of this field, introduced its important development and application, and made an attempt to promote further development in this field. The paper first briefly introduces the structure of the network and then outlines the structural models

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 514–522, 2020.
https://doi.org/10.1007/978-981-15-0187-6_61

commonly used in deep learning. It mainly introduces Alex Net, ZF Net, VGG Net, Google Net, Res Net, spatial transformer module, capsule network, and a comparative analysis of the various improved models for deep learning. Finally, the problems that still need to be solved in this field and the future of deep learning are introduced.

2 The Composition and Principle of Neural Network

The basic neural network consists of input layer, hidden layer, and output layer. The structure is shown in Fig. 1. For the classification problem, the main task of the input layer is to read the information of the image, the number of neurons is the number of pixels of the input image, and the number of neurons in the output layer is the number of categories. For example, if the MNIST handwritten digital image set is a 28-pixel × 28-pixel grayscale image, the number of neurons in the input layer of the neural network is set to 28 × 28 = 784 neurons, and the 10 neurons in the output layer are derived from 10 categories. (Numbers 0 to 9 for a total of 10 categories) The activation function converts the sum of the input signals into an output signal. The activation function mostly uses Relu, and the activation function of the output layer uses the Softmax function. The convolutional neural network also includes a convolutional layer, a pooled layer, and a fully connected layer. The convolutional layer is used for feature extraction, and the convolution operation is equivalent to the filter operation in image processing. Each time a feature is perceived locally in the image and then synthesized at a higher level to obtain information about the entire image. And in order to improve efficiency, parameter sharing and local sensing are introduced [1]. The pooling layer is an operation for reducing the space in the high and long directions and is disposed behind the convolution layer. This operation can reduce the amount of data while retaining valid information. Max pooling is generally used in identifying problems. After using multiple convolutional layers and pooling layers, the fully connected layer is fully connected to all neurons in the previous layer, integrating local information, and linking all features. Convolutional neural networks as the hierarchy deepen, the extracted information is increasingly complex and abstract, from the initial layer to the subsequent layer in turn respond to image edges, textures, and more complex object components. In the improved CNN model, the activation function main uses the ReLu function. Its expression is as follows $h(x) = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$ When the input is greater than 0, the value is directly output; when the input is less or equal to zero, 0 is output. Most use dropout technology, and use GPU parallel training technology, the classifier uses Softmax regression technology. And because training requires large data sets, data enhancements are used to enhance data size in improved CNN. When data is input, the data is usually packaged and input. This operation is called batch processing. Batch processing can greatly shorten the processing time of each image and improve the computing speed of the computer.

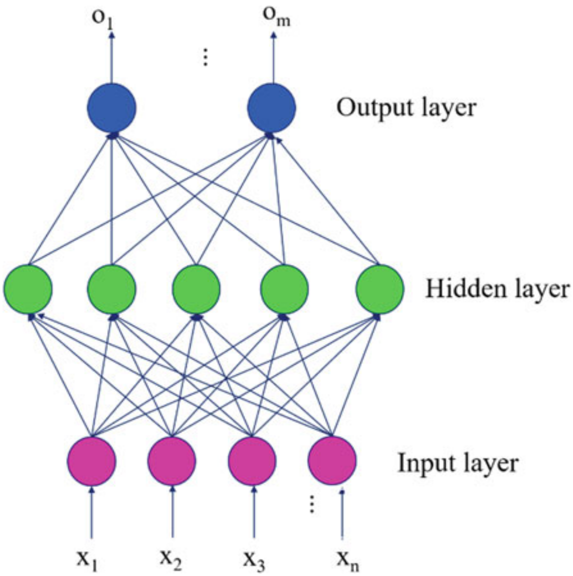


Fig. 1. Simple neural network structure

3 CNN Model

ALexNet [2] is considered to be one of the most far-reaching models in the industry, achieving a 15.4% success in the top five test error rate. It is much better than the 26.2% of the second place, indicating that CNN has a surprising advantage over other methods. Its layout structure is relatively simple compared to some current architectures. It mainly includes five convolutional layers, dropout layer, maximum pooling layer, and three fully connected layers. The data set uses the Imagenet data set, which has a data set of 15 million high-resolution images with labels, with approximately 22,000 categories. The activation function uses the ReLu function, which is six times faster than the activation function using tanh neurons. Excessive training sets make it unsuitable for training on one GPU, and parallel computing on two GPUs can achieve good results. The network has 60 million parameters, although there are many categories in our data set, these categories are not enough for so many parameters. In order to reduce the overfitting of the network, the model uses data-enhanced technology, that is, artificially expand the data set by image transformation, horizontal reflection, patch extractions, etc. In addition, the model also uses the dropout technique to reduce overfitting. Its approach is to set the output of each neuron to zero with a probability of

0.5. In the test, the output of all neurons was multiplied by only 0.5, So every time the neural network is trying different structures, but the weights are shared between these structures. Dropout greatly reduces overfitting and roughly doubles the number of iterations required for convergence. Architecture of ALexNet runs in parallel on two GPUs as shown in Fig. 2.

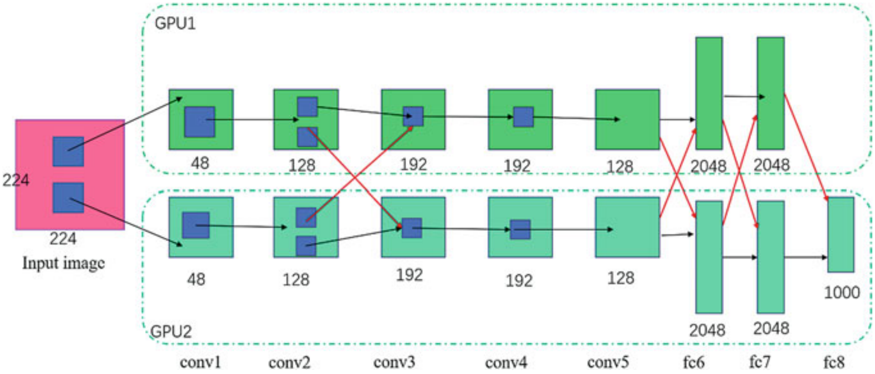


Fig. 2. Illustration of the architecture of ALexNet run in parallel on two GPUs

The neural network is considered to be a black-box operation. It is difficult to understand its internal operating mechanism. If it cannot be solved, the improvement of its network architecture can only rely on trial. ZF Net invented a convolutional network visualization technology called deconvolutional network that allows researchers to know what the convolutional layer is observing and what it is doing in the CNN. The basic principle is that after each layer of the trained CNN network, a deconvnet layer with the same filter settings as the original CNN network is attached, and the basic working principle of deconvolution is shown in Fig. 3. If we want to check the activation value of a feature in a layer, save the activation value of the corresponding feature layer of this layer, and set the other activation values in this layer to zero, then the feature layer is used as the input of the deconvolution network [3]. One of the most interesting places in deep learning is that the first layer responds to simple edges, the next layer responds to textures, and the later layers respond to more complex object parts, if stacked multiple layers, as the layers are deepened, the extracted information becomes more complex and abstract. ZF Net provides CNN with a more intuitive display capability, enabling researchers to better understand the internal workings of CNN and provide guidance for further improving the CNN model. In addition, compared to AlexNet's 11 $*$ 11 filter, ZF Net uses a smaller 7 $*$ 7 filter with smaller steps size, which helps preserve the raw pixel information of the input data.

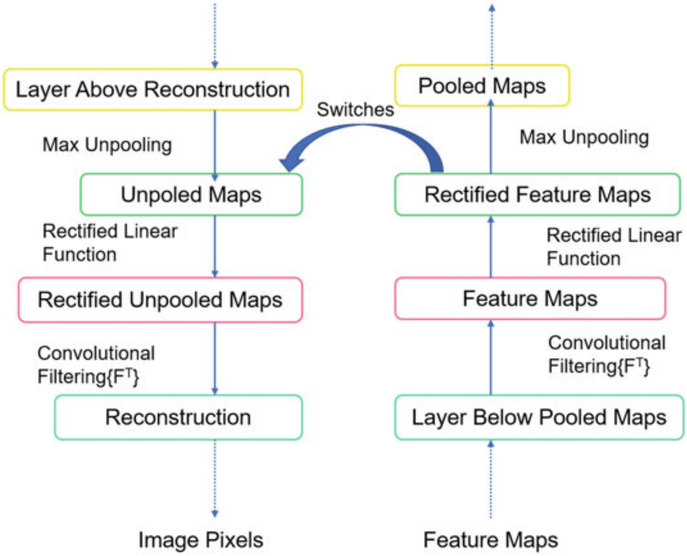


Fig. 3. Basic working principle of deconvolution

Unlike the first two networks, VGG Net uses only 3 ∗ 3 filters. The authors believe that two 3 ∗ 3 filters can be combined to generate a 5 ∗ 5 sensing zone [4]. In this way, it can maintain the same function as the large size and has the advantage of small size. For example, the small size can reduce the parameter, and the multi-purpose linear rectification layer can reduce the performance of the system. Another feature of the network is that it emphasizes depth, that is, the layer with the weights is superimposed on the 16th or 19th layer, and the network structure deep enough to reflect the hierarchy of processing visual data. VGG Net maintains depth and keeps it simple. Its structure is shown in Fig. 4.

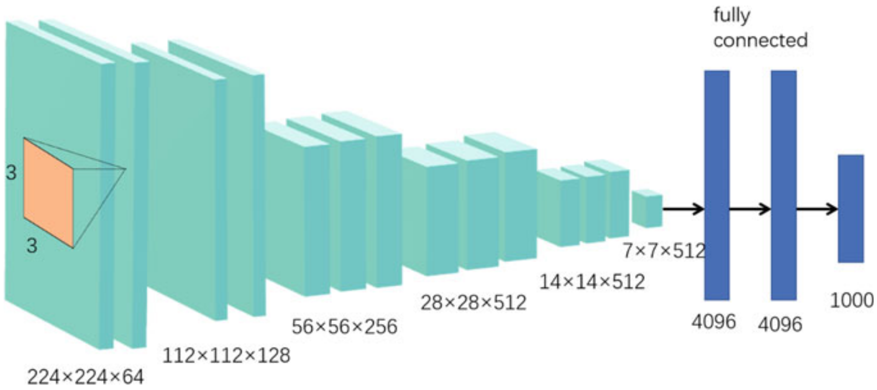


Fig. 4. Architecture of VGG Net use only 3 ∗ 3 filter size

GoogLeNet seems to be very complicated, but it is actually the same as the network structure introduced earlier. However, GoogLeNet is characterized by not only depth in the vertical direction but also depth in the horizontal direction. This is called the inception structure [5]. This structure uses multiple filters and pools of different sizes, and finally merges their results. The author provides a 1 ∗ 1 convolution operation in many places, which can reduce parameters and achieve high-speed processing. Its full inception module is shown in Fig. 5.

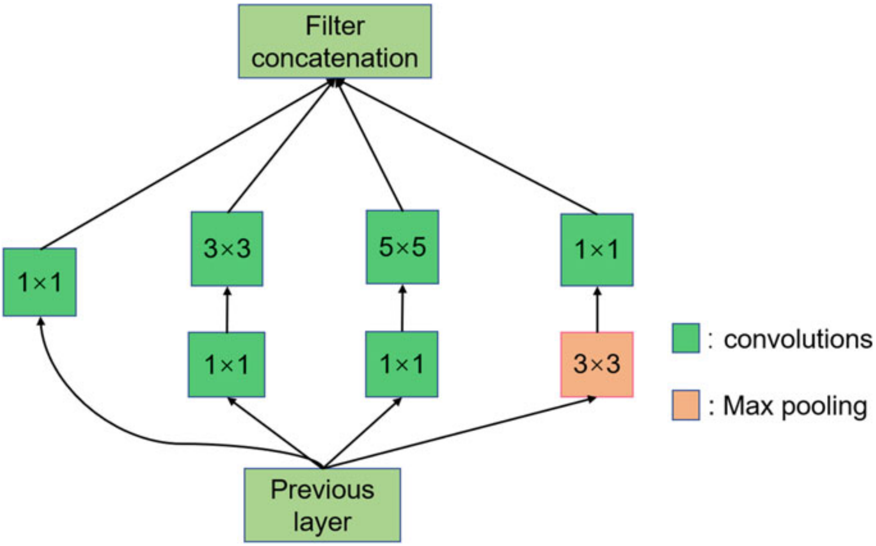


Fig. 5. Full inception module

ResNet is a network with a deeper structure than the previous network. Deeper can improve performance better. However, the excessive deepening level will not work smoothly in many cases resulting in poor performance. ResNet has introduced a shortcut structure that solves this type of problem by merging the input into the output layer across the convolutional layer of the input data [6]. This not only deepens the layer but also enables efficient learning. The main reason is that the signal can be transmitted without attenuation when backpropagating. According to the experimental results, even if the depth is increased to 150 or more, the recognition accuracy will continue to increase.

The network structure introduced above is through large-scale modification of the network architecture, or create another complex model, and a spatial transformer module network is more focused on transforming the input image, and its transformation is mainly aimed at pose normalization and spatial attention, the former mainly refers to whether the object in the scene is tilted or stretched, and the latter refers to how to focus on a specific object in the image of a multi-object [7]. In traditional CNN, spatial invariance mainly depends on the maximum pooling layer, which can extract the relative position of feature information as an important attribute, while the spatial change module performs affine transformation on the input image in a dynamic way. This gives the model a strong deformation, expansion, and rotation invariance.

There is also a network that can improve the recognition ability of an object after rotation and the recognition between objects. It is a capsule network proposed by Hinton et al. in 2017 [8]. Traditional CNN is based on neurons, and a neuron represents an attribute of an object. CapsNet consists of multiple capsules, each consisting of multiple neurons. Therefore, it has more attribute information than the neural network. There is an algorithm called dynamic routing between capsules. The main idea is that the output of the low-level capsules is passed to high-level capsules that can be modified. If the input of the low-level capsule is very similar to the output of the high-level capsule, then the two capsules are considered to be highly correlated. The current capsule network is only three layers shallow, but its performance is better than the traditional CNN network of the same level. In the future, if we increase the level and further study the routing algorithm, it will have great development prospects in image recognition.

In 2019, Zhang et al. proposed a multi-feature weighted network containing two modules, CFR-DenseNet and ILFR-DenseNet, called MFR-DenseNet [9]. CFR-DenseNet adds the SEM module to DenseNet, which is mainly for the channel characteristics of the convolutional layer. ILFR-DenseNet adds DSEM to the front of the $3 * 3$ convolution layer to simulate the interdependence of different features between layers. Unlike other CNN models that are superimposed by multiple convolutional layers, its input is the output of all previous layers. MFR-DenseNet is a network formed by combining CFR-DenseNet and ILFR-DenseNet. This kind of convergence can maximize their respective advantages, so that the network can not only explore the correlation between channels, but also simulate the interdependence of interlayer features. DenseNet is composed of many dense blocks. Its main feature is that each layer has a relationship with other layers. This ensures maximum information flow, suppresses gradient disappearance during network training, enhances feature propagation, and is effective reduce model parameters. MFR-DenseNet is the latest research result in CNN improved model. Due to its special structural features, CNN plays an irreplaceable role in image recognition.

4 Conclusion

Researchers have made significant contributions to deep learning and image recognition. However, at present, deep learning is still not perfect in target detection and image recognition, and there are still many problems that need to be solved.

The convolutional layer is the most time-consuming when learning on the network. Studies have shown that the processing time of all convolutional layers accounts for 89% of the total CPU and 95% of the GPU. Therefore, the main research goal of high-speed learning is how to efficient and high-speed performing a large number of multiplication and accumulation calculations. The CPU is mainly used for continuous and complex operations, while the GPU is better at a large number of parallel operations, and its application in deep learning will achieve an amazingly high speed. Even so, when the level of the network deepens, it still takes days or even weeks to learn. And in order to build a better network, deep learning is always accompanied by various trials and errors, which creates the need to shorten the learning time as much as possible. For this reason, distributed computing can be considered on multiple GPUs, but this is a problem that is difficult to overcome because it involves multiple problems such as communication between machines and synchronization of data. Considering that memory capacity and bus bandwidth may also become bottlenecks, the number of data bits flowing through the network should be reduced as much as possible. In order to achieve high speed, bit number reduction is also a subject of concern. In the training of the network, a large data set is usually needed, and the knowledge of the existing public data set can be transferred to the own network, and a part of the trained model is directly applied to the similar model, such as a small amount. The data only trains part of the network and can achieve good results on small data sets. This migration learning technology is one of the future research directions. Through deep learning, you can also draw artistic paintings, such as drawing a painting with a Van Gogh style, inputting two images, one content image, another style image, and a new image will be generated. In the process of learning, the intermediate data of the network is similar to the intermediate data of the content image, and an input image such as a similar style of the content image can be output [10]. Different from the above research, there is also a network model that can generate new images without any input image. For example, based on deep learning, it is possible to generate images of the living room from zero. This technology is mainly developed by letting the two neural networks of generator and discriminator learn from each other. The generator that will eventually grow up will have the ability to draw images that are enough to confuse the real [11]. Deep learning has shown excellent performance in many aspects and has become a research hotspot. However, there are still many works to be done in the theoretical research of the model. More research work on deep learning requires researchers to carry out.

References

1. Shi C, Pun CM (2019) Adaptive multi-scale deep neural networks with perceptual loss for panchromatic and multispectral images classification. Inf Sci
2. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
3. Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, Cham, pp 818–833
4. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409–1556
5. Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9
6. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
7. Jaderberg M, Simonyan K, Zisserman A (2015) Spatial transformer networks. In: Advances in neural information processing systems, pp 2017–2025

8. Sabour S, Frosst N, Hinton GE (2017) Dynamic routing between capsules. In: Advances in neural information processing systems, pp 3856–3866
9. Zhang K, Guo Y, Wang X et al (2019) Multiple feature reweight densenet for image classification. IEEE Access 7:9872–9880
10. Gatys LA, Ecker AS, Bethge M (2015) A neural algorithm of artistic style. arXiv preprint arXiv:1508–06576
11. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511–06434

Research on the Development and Application of Unmanned Aerial Vehicles

Kai Qiu(✉), Chifei Zhou, Yijing Liu, Lanlan Gao, Xianzhou Dong, and Jiang Cao

Academy of Military Science of the PLA, Beijing 100091, China
q1987k@163.com

Abstract. The new military revolution in the new era is spurring the development of combat means and combat forces to unmanned, intelligent, and clustered direction and gradually forming new methods of warfare and winning mechanisms. Unmanned combat platforms represented by unmanned aerial vehicle (UAV), unmanned combat vehicle, unmanned surface vessel, and unmanned underwater vehicle have gradually become new research areas in which various military powers are committing to. Among the various unmanned combat platforms, UAVs achieve the fastest development and have the most operational applications. This paper gives a brief overview of UAVs and their applications and focuses on the advantages and problems of UAV and UAV cluster.

Keywords: Unmanned combat · Unmanned aerial vehicle · UAV · Drone

1 Introduction

Unmanned aerial vehicle (UAV) is a non-manned aircraft operated by radio remote control equipment and self-contained program control devices or operated autonomously by the onboard computer completely or intermittently. In the Vietnam War of the 1960s, UAVs were first used in wars. The American “Fire Bee” unmanned aircraft carried out reconnaissance missions over Vietnam and obtained accounted for 80% of the total intelligence, which made people realize UAV’s value. In 1991, during the Gulf War, more than 200 drones from the USA, Britain, France, Canada, and Israel all appeared on the battlefield, which provided an important basis for the multinational forces to control the battlefield situation in real time and evaluate the effects of air strikes. In 2003, the US military mobilized more than 10 kinds of drones to participate in the Iraqi battlefield. The Global Hawk carried out several combat missions over Baghdad and collected more than 3700 images [1].

With the rapid development of military revolution, unmanned combat platforms represented by drone, unmanned combat vehicle, unmanned surface vessel, and unmanned underwater vehicle have gradually emerged in military applications, emitting enormous development potential and broad application prospects. The main functions of unmanned equipment have evolved from past target drones, reconnaissance, etc., to the current functions of reconnaissance, surveillance, communication, and attack. With the development of artificial intelligence, network, coordination, and

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 523–530, 2020.
https://doi.org/10.1007/978-981-15-0187-6_62

control technology, all kinds of unmanned cluster like “wolves”, “fish groups”, and “bee colonies” may appear in various combat spaces such as land, sea, air, and sky. The operational cluster implements a global unmanned cluster attack and defensive operation against the enemy. The National Science Council has predicted: “The core weapon of the twenty-first century is the unmanned combat system” [2].

Due to the diversity of drones, there are different classification methods for different considerations: According to the structure of flight platform, it can be divided into fixed-wing UAVs, rotary-wing UAVs, unmanned airships, wing drones, etc.; classified by purpose, it can be divided into military drones and civilian drones, of which military drones are divided into reconnaissance drones, electronic countermeasure drones, communication relay drones, unmanned fighters, target drones, etc., and civil drones are divided into inspection/monitoring drones, agricultural drones, meteorological drones, exploration drones, mapping drones, etc.; according to scale, it can be divided into micro drones, light drones, small drones, and large drones; classified by activity radius, it can be divided into super short-range drone, short-range drone, medium-range drone, and remote drones; according to mission height, it can be divided into super low-altitude drones, low-altitude drones, medium-altitude drones, high-altitude drones, and super high-altitude drones.

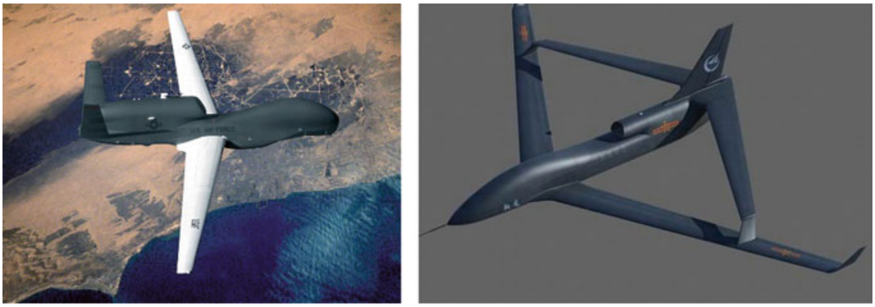
2 Development of UAVs

As early as the beginning of the twentieth century, the US military considered converting a manned aircraft into a specialized unmanned torpedo attack aircraft. During World War II, the US military envisioned converting the B-17 and B-29 bombers into unmanned bombers, but subject to the technical conditions at the time, both programs ended in failure. It was not until the 1950s that the US military successfully used the BQM-147 “Fire Bee” unmanned high-altitude reconnaissance aircraft in the Vietnam War. Afterward, many countries competed to follow suit and developed their own drones, whose functions were mainly based on reconnaissance and surveillance.

Since the beginning of the twenty-first century, the rapid development of electronic information technology has provided an opportunity for the breakthrough of many key technologies of drones. In addition to the traditional reconnaissance and surveillance functions, drones have added new features such as communication and navigation, electronic countermeasures, and firepower. These combat support functions enable military experts from all over the world to see the trend of drones transitioning from performing support tasks to direct participation in the war. According to its operational requirements and operational effectiveness assessment, the foreign military has made a number of UAV development plans that are suitable for different regions and deployment phases. The US military has an unmanned system integrated road map, Russia has formulated the “National Arms Plan 2018–2025”, the British army has a development plan of “watchman” drone, and Europe has a development plan of “neuron” UAV. Among them, the USA has the most complete range of drones, the highest level of technology, the largest number of services, and the most experienced use [3, 4].

The development of US military drones is at the world's leading level. There are unmanned reconnaissance aircrafts at all levels of strategy, campaign, and tactics, as well as reconnaissance/attack UAVs and transport drones. At present, US military drones are deployed in large areas such as the air force, army aviation, and naval air force, including RQ-4 "Global Hawk", RQ-7 "Shadow", MQ-1B "Predator", MQ-9A "Reaper", and BQM-147 "Fire Bee" drones. From low altitude to high altitude, short range to long range, micro to heavy, the US military covers almost all types of military drones, even vertical takeoff drones and transport drones. Compared with foreign troops, the development of China's UAVs started relatively late. However, after entering the twenty-first century, explosive growth began, especially the successful development of "Wing Loong", "Soar Dragon", and "CH" series UAVs, marking that China's level of R&D and design of military drones has entered the international advanced level.

The RQ-4 "Global Hawk" is a high-altitude unmanned reconnaissance aircraft, shown in Fig. 1a. Its total length is 13.5 m, with the wingspan of 35.4 m and the height of 4.6 m. The maximum takeoff weight is 11,610 kg, and the ceiling is 20,000 m. Its maximum flight speed is 740 km/h, and the cruising time is 42 h. With a range of more than 26,000 km, it can fly from the United States to anywhere in the world. Equipped with high-resolution synthetic aperture radar and photoelectric/infrared modules, it can provide long-range and long-time dynamic monitoring. It can also perform espionage work on spectrum analysis, discovering crises, and conflicts around the world in advance and can also help guide the air force's missile bombing to reduce the number of missed attacks.



(a) RQ-4 "Global Hawk" **(b) Soar Dragon**

Fig. 1. Unmanned reconnaissance aircrafts

The "Soar Dragon" drone is also a high-altitude unmanned reconnaissance aircraft, shown in Fig. 1b. The total length is 14.3 m, the wingspan is 24.9 m, and the height is 5.4 m. Its maximum takeoff weight is 6800 kg, the mission load is 600 kg, and the ceiling is 20,000 m. The maximum flight speed is 700 km/h, the battery life is 10 h, and the combat radius is 2500 km. The UAV's electronic equipment is similar to the RQ-4 "Global Hawk," with high-definition digital cameras, digital TVs, and equipped with synthetic aperture radar, inverse synthetic aperture radar, and several parallel communication systems. In addition, the digital communication relay task can be performed by replacing the modular head electronic task compartment, and it plays a very high signal-forwarding tower.

The MQ-9A "Reaper" UAV is a reconnaissance/attack drone developed on the basis of the MQ-1B "Predator", with the flight speed and the amount of bomb load greatly improved, shown in Fig. 2a. Its total length is 11 m, the wingspan is 20 m, and the height is 3.8 m. The maximum takeoff weight is 4760 kg, and the ceiling is 15,000 m. Its maximum flight speed is 460 km/h, and the combat radius is about 5900 km. It can fly continuously for more than 15 h under the status preparing for war. The main airborne weapons include two GBU-12 laser-guided bombs and four AGM-114 "Hellfire" air-to-ground missiles. They can also carry 227 kg of combined direct attack ammunition and 113.5 kg of small-diameter bombs, which can accurately hit the target in inclement weather.

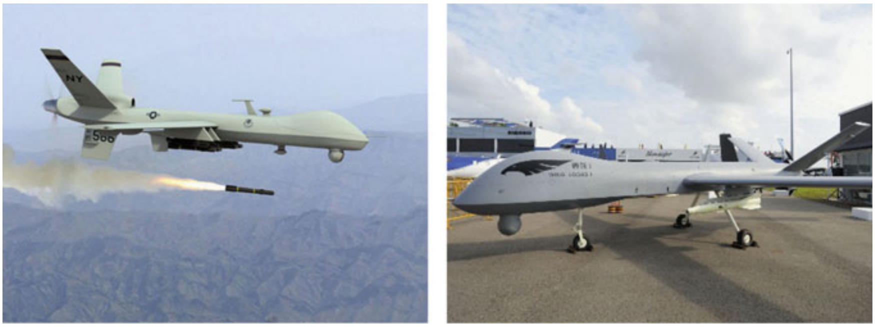

(a) MQ-9A "Reaper" (b) WingLoong-I

Fig. 2. Reconnaissance/attack drones

The "Wing Loong-I" drone is a multi-purpose UAV with medium-altitude, long-range and long-haul, as well as full autonomous takeoff and landing and flight capabilities, shown in Fig. 2b. It looks like the MQ-9A "Reaper", and the size of the fuselage is similar to the MQ-1B "Predator". Its total length is 9 m, with the wingspan of 14 m and the height of 2.7 m. The maximum takeoff weight is 1200 kg, and the ceiling is 5300 m. Its maximum flight speed is 280 km/h, the battery life is 20 h, and the maximum range is 4000 km. The total payload capacity is 200 kg, and the front-view infrared sensor is about 100 kg. Therefore, each wing can also carry 50 kg of ammunition to accurately strike the ground targets.

Since 2001, the USA has released eight versions of the unmanned system integrated road map to assist decision makers in the Department of Defense to plan long-term strategies for research and development and procurement of drones and to guide the industry to carry out research work of drones. In 2009, the "United States Air Force Unmanned Aerial System Flight Plan 2009-2047" was released, which is an actionable plan characterized by Doctrine, Organization, Training, Materiel, Leadership and

Education, Personnel, Facilities, and Policy (DOTMLPF-P) recommendations. It consists of four ideas, five main steps, and several key objectives, whose vision is to harness increasingly automated, modular, globally connected, and sustainable multi-mission unmanned systems resulting in a leaner, more adaptable, and efficient air force.

3 The Advantages of UAVs

Due to its strong adaptability, maneuverability, and no risk of life, drones have received extensive attention in the civil and military fields. Especially in the military field, UAVs have great advantages in performing reconnaissance and surveillance, ground attack, communication relay, setting false targets, and electronic interference. As a substitute for humans to complete the “dull, boring, harsh, dangerous” missions, the military UAVs have strong adaptability, long battery life, and large load capacity, which can provide strong support for military operations. On the one hand, it can realize high-definition coverage reconnaissance and surveillance in the entire airspace for a long time. On the other hand, it can also perform non-lethal attacks or precise destruction on target objects according to operational intentions and provide reliable communication relay and delivery guarantee for combat command [5, 6].

- Low using cost

UAVs do not need to install airborne equipment for pilots such as the cockpit, life support systems, lifesaving systems, human–computer interaction systems, and so on. At the same time, since there is no need to carry pilots, drones do not have a limitation on the size of the body. Therefore, drones can do cheaper than man-machines to achieve the same performance indicators in theory. In addition, drone operators do not need to operate on the aircraft in the sky, because the training on the simulator is not much different from the actual training environment, which can save a lot of training costs.

- Good stealth performance

Currently, the radar scattering cross-sectional area on the side of manned stealth aircraft is usually not ideal. One of the largest short plates is the size limitation of the pilot cockpit, resulting in that the height of the fuselage cannot be lowered too low. However, UAVs are not bothered by this problem. They can be designed with a flatter body and a thinner wing as needed and can also form a wing-body fusion design that is more concealed, thereby greatly reducing the radar scattering cross-sectional area on the side of the fuselage and improving stealth performance.

- Less casualties

For a manned aircraft, its casualties mainly come from two aspects: the usual training accident caused by unskilled operation or aircraft failure and the casualties caused by the aircraft being hit by the enemy in wartime. However, for UAVs, whether in peacetime or in wartime, drone operators can operate the machine remotely from the ground control station, thus avoiding the military personnel directly in danger. There are no pilot casualties even if the plane is shot down. In addition, the long-haul characteristics usually allow the drone to have sufficient time to accurately identify and lock targets in the theater, thereby reducing accidental injuries and collateral damage.

- Strong maneuverability

At present, for a fighter aircraft based on metal material structure, the turning overload that the body structure can withstand is usually within 15G. However, due to the physiological limitations of the pilot, the turning overload will generally not exceed 9G. For a drone, in theory, as long as the engine and maneuverability allow, the drone can make the most overloaded turning maneuver that the body can withstand. In the future, with the improvement of autonomy, drones can greatly shorten the OODA ring. As a result, the agility of the drones will be higher, which can respond to the battlefield situation more quickly.

- Long endurance

For a manned aircraft, its endurance is mainly limited by the pilot's physiology. If the flight time is too long, the pilot will have fatigue and negative emotions, and it is prone to safety accidents. However, the drone operators can operate the aircraft in the ground control station by turns. Meanwhile, automatic flight can be set during the cruise phase, and nearly no human intervention is required. As long as the power and structure of the drone are sustainable, it can be active in the theater for a long time without interruption, so the endurance is far stronger than that of the manned aircraft.

4 UAV Formation

With the continuous development of artificial intelligence technology, the combat style of UAVs has gradually evolved from "single-handedly fighting" to "clustering intelligence", shown in Fig. 3. At present, the world's military powers are all actively developing the UAV cluster combat technology, because the UAV formation can improve the efficiency of task that a single drone cannot complete in reconnaissance

Fig. 3. Schematic diagram of cluster formation

and surveillance, target strike, communication relay, electronic interference, battlefield evaluation, and so on [7, 8].

UAV cluster combat usually refers to hundreds of thousands of small, fast, and powerful UAVs in the battlefield airspace, simulating cluster behaviors such as bee colonies, fish schools, and ant colonies to construct bionics formations. Interactive communication is carried out through various channels such as combat data link systems, tactical radio systems, and communication relay networks. They rely on advanced technologies such as cloud computing, big data, and artificial intelligence to conduct operational coordination and carry out strategic deterrence, campaign confrontation, and tactics action in military operations [9, 10].

UAV cluster combat has multiple advantages [11]. (a) Diversity functions: With different equipment for different drones, a UAV formation can possess multiple functions such as reconnaissance, surveillance, electronic interference, strike, and evaluation; (b) the effectiveness of combat: a UAV formation can attack one target simultaneously from different directions, or strike against multiple targets at the same time; (c) group's invulnerability: When a certain drone in the formation fails or is destroyed by the enemy, the formation can adjust the task division and use another drone to replace it to ensure that the entire formation can continue to perform the task.

There are also some problems in UAV cluster combat that need to be solved [12]: (a) formation flight control technology: When performing certain tasks, the UAV formation needs to change the structure of the formation in real time according to the mission requirements during the flight, which poses a great challenge to the controller; (b) anti-collision and obstacle avoidance technology: When a drone in a formation is disturbed, or in the process of turning maneuver or at high speed, there is the possibility of collision; (c) track planning problem: The UAV formation not only needs offline route planning to avoid known threats and obstacles, but also needs to renew the track in real time during the flight to avoid emerging threats and obstacles.

5 Conclusions

Same as drones, unmanned combat vehicles, unmanned surface vessels, and unmanned underwater vehicles have also achieved great development. With the further breakthrough of key technologies such as modularity, interoperability, intelligence, and autonomy, in the future, unmanned systems can do almost anything that a manned system can do, and they can do it better. There is no doubt that the future battlefield will be dominated by unmanned combat systems and will present cluster, three-dimensional, intelligent, and synergistic features. Therefore, in order to effectively cope with the future intelligent warfare, we need to plan well in advance, rationally arrange, and vigorously develop unmanned combat systems such as drones that are suitable for China's national conditions, and ensure that our army can catch up with Western powers in this new field, and further, walk in the forefront of the world.

References

1. Jiang J, Wang M, Jiang B (2019) Research on UAV combat applications. Aerodyn Missile J 1:41–44
2. Yan J, Wu Y (2017) The impact of intelligent unmanned war on the allocation of defense resources. Natl Def 10:69–72
3. Zhang H, (2017) Research on U.S. UAV collaborative operations. Aerodyna Missile J 8:12–16+21
4. Cheng L, Luo L, Chai J (2018) Review of the development of military UAV equipment technology in 2017. Sci Technol Rev 36(4):69–84
5. Liu Z (2018) Analysis on the operational use and development of American reconnaissance UAVs. Electron World, 5
6. Zhu T, Li L, Ling H (2018) The operation and enlightenment of UAVs in the Syrian war. Aerodyn Missile J 11:31–34
7. Xu B, Zhang Y, Wang C (2018) Colony technology development status and trends analysis of the US military unmanned systems. Aerodyn Missile J 3:36–39
8. Wang H, Deng D (2017) Capability analysis and countermeasures research on cluster UAVs. Aerodyn Missile J 4:15–20
9. Shen C, Li L, Wu Y, Liu D (2018) Research on the capability of the U.S. manned/unmanned autonomous collaborative operations. Tactical Missile Technol 1:16–21
10. Haiyan Gu (2018) Development inspirations of US multiple-UAV cooperative engagement. Telecommun Eng 58(7):865–870
11. Li H (2018) UAV cluster operations under the curtain of informationized war. MilY Dig 5:22–23
12. Zhang C (2018) UAV formation: the operational prospects can be expected and the technology is to be broken. China Aerospace News 5

Recent Development of Commercial Satellite Communications Systems

Jinhui Huang(✉) and Jiang Cao

Academy of Military Sciences, Beijing, China
night098@sina.com

Abstract. Satellites are playing an increasingly important role in communication fields. Many commercial satellite communication systems are designed or constructed now. This paper reviews the recent developments in commercial satellite communications and outlines the performance characteristics of several typical satellite systems, such as Iridium-NEXT, LeoSat, OneWeb, StarLink, and so on. Prospects for the development of satellite communications are expected. The world of satellite communications is hotting-up, and a wide variety of design options will be explored. The future of high-speed "space Internet" is increasingly bright.

Keywords: Satellite communication · Inter-satellite links · Satellite miniaturization · Iridium-NEXT · LeoSat · OneWeb · StarLink · O3b · Kuiper

1 Introduction

In October 1945, British science-fictional writer C. Clarke published an article titled "Extra-Terrestrial Relays—Can Rocket Stations Give Worldwide Radio Coverage?" in the journal Wireless World. In this paper, Clarke put forward the concept of geostationary satellites and calculated the required orbital characteristics for the first time. However, due to the poor performance of radio tubes at that time, almost no one, even including Clarke, anticipated that the concept of satellite would change rapidly from dream to truth in less than 20 years. The invention of transistor and the integrated circuit sped the process of satellite research and development. In 1957, the Soviet Union launched the world's first artificial satellite, Sputnik 1. Eight years later, the world's first practical commercial communications satellite, known as Early Bird, was launched successfully by the USA [1].

In the 1990s, commercial satellite communication experienced low tide due to the fast development of the fiber and terrestrial cellular systems. All three of the dedicated mobile satellite systems, Iridium, ICO, and Globalstar, failed financially and declared bankruptcy one after another [1]. Into the twentieth century, with the advancement of electronic information technology and the further increase in communication requirements, satellite communication has once again received widespread attention [2].

Satellite communication has many advantages, such as wide coverage, large communication capacity, good transmission quality, convenient and rapid networking, and easy global seamless connection [3]. Satellites have been playing an increasingly

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 531–536, 2020.
https://doi.org/10.1007/978-981-15-0187-6_63

important role in communication fields. A large number of satellite communication systems are designed or under construction now [4].

The remainder of this paper is organized as follows. In Sect. 2, several typical communication satellite systems are briefly introduced. Section 3 identifies some development prospects. Finally, Sect. 4 comes the conclusion.

2 Typical Commercial Communication Satellite Systems

2.1 Iridium-NEXT

The Iridium constellation was conceived in the late 1980s by several engineers in Motorola. An early computation revealed that 77 satellites would be needed to achieve full global coverage including the polar regions. Since the number of satellites is the same as the number of electron circling the Iridium nucleus, the project handled by Motorola was named Iridium. The name retained although the number of active satellites composing the constellation reduced to 66 at last.

Iridium satellites began to be launched in 1997. The full constellation was established in May 1998, and the satellite system started offering services in November 1998. Due to the unexpected fast development of the ground mobile communication, it was difficult for the Iridium to attract enough users. As a result, the company went bankrupt in 2000. In 2007, Iridium Satellite LLC (the commercial entity that took over the assets of the original bankrupt Iridium consortium) announced the Iridium-NEXT plan for replenishing the Iridium constellation to provide higher data rates, increase capacity, and introduce new services.

The constellation of the Iridium-NEXT is the same as the original Iridium system with 66 active satellites in six orbital planes. There are nine spare satellites at the same time. The altitude and inclination of the orbits are 780 km and 86.4°. The orbital period is 100.5 min. Except for planes 1 and 6, the adjacent orbital planes are co-rotating and spaced 31.6°. The planes 1 and 6 are counter-rotating at a spacing of 22°.

The Iridium-NEXT satellite establishes L-Band and satellite-ground links to mobile and OpenPort terminals at data speeds of up to 128 kbit/s and 1.5 Mbit/s, respectively. For fixed and transportable terminals, the service speed is higher, up to 8 Mbit/s. Each Iridium-NEXT satellite establishes 23.18–23.38 GHz inter-satellite links to four satellites, two in front and behind within the same orbital plane and the other two in neighboring planes to either side, at data speeds of 12.5 Mbps [5].

2.2 LeoSat

The LeoSat system is developed by LeoSat Enterprises together with Thales Alenia Space. A constellation of up to 108 low earth orbit (LEO) communications satellites is going to be launched to form a mesh network interconnected with laser inter-satellite links. The altitude of the orbits is about 1400 km. A total of 108 satellites will be deployed in six orbital planes.

LeoSat's system adopts optical inter-satellite links, flat panel antennas, optimized Ka-band communications, and semiconductor processing. Each satellite in the LeoSat constellation supports ten Ka-band steerable user antennas, two steerable gateway antennas, and four optical inter-satellite links. Each of the user antennas provides a user terminal data rate within the range of 50–1.6 Gbps. Two steerable gateway antennas transmit at speeds up to 10 Gbps. The data rate of the optical inter-satellite links (ISLs) is up to 1.6 Gbps and even 5.2 Gbps were needed [6].

2.3 OneWeb

OneWeb needs less than 600 satellites working at Ka (20/30 GHz) and Ku (11/14 GHz) bands to provide global coverage Internet connection. These satellites will first be launched into a 450–500 km altitude and then climb to an operational orbit of 1200 km from there. The satellites used in the OneWeb constellation have the following characteristics: small, modular design, high productivity, low cost, and strong industry participation. The weight of a satellite is less than 150 kg.

The satellite production rate is about 40 satellites/month. OneWeb plans to start launching from 2019 and launches every 21 days over two years across multiple sites and brings truly global coverage in 2021. OneWeb was approved by the Federal Communications Commission (FCC), which means that its LEO satellites have successfully entered the US satellite market [7].

2.4 StarLink

Starlink is a large-scale constellation proposed by SpaceX. The FCC has given SpaceX permission to launch two sets of satellites for Starlink: one set of 4409 satellites at the height of 1150 km and another set of 7518 satellites at the height of 335–346 km. SpaceX plans to offer speeds of 10 Gbps to every human on the planet and undercut land-based networks [8]. The limited service will be started by as early as 2020 [9].

As a condition of approval, SpaceX needs to launch half of all those satellites, or about 6000, within the next six years. Those 4409 satellites work at Ka and Ku band, while other 7518 satellites work at V-band. After networking, two sets of satellites will operate together to provide Internet access services for customers across the world.

As part of its efforts to prevent space debris, SpaceX later applied to deploy 1584 of the set of 4409 satellites at 550 km orbit, rather than the previously claimed 1150 km orbit. Space debris at low altitudes can quickly re-enter the atmosphere and disappear automagically, because of low altitude atmospheric drag. SpaceX is designing a satellite that will burn up completely when it re-enters the atmosphere, greatly enhancing the space security [10, 11].

2.5 O3b

Due to economic and geographical constraints, there are still 3 billion people around the world, mainly in Africa, Asia, and South America, who have no access to the Internet now. O3b Networks Ltd., named after “the Other 3 Billion” and founded in 2007, is the operator of the O3b satellite constellation. It aims to provide low-latency, high-bandwidth, and low-cost satellite Internet access to these 3 billion people in remote parts of the world [12].

O3b plans to build a constellation network of medium earth orbit (MEO) communications satellites that will provide satellite communications backbones with fiber-optic transmission speed using Ka-beam antenna technology. The constellation is composed of 20 MEO satellites. The final launch took place in April 2019 [13]. The fleet of satellites orbits the earth in a circular, equatorial orbit at an altitude of 8063 km. It covers all places between the north and south latitudes of 45° and can provide certain services in the range of 45° to 62° in the south and north latitudes. Each satellite is equipped with 12 fully steerable Ka-band antenna. Ten beams are for user terminals, and two beams are for gateway connections. Each antenna provides a data throughput of 1.25 GBit/s, 600 MBit/s for up and downlink resulting in a total capacity of 12.5 GBit/s per satellite [14].

In 2021, seven next-generation MEO satellites will be added to the O3b fleets, forming O3b mPOWER, the new satellite-based system. With the addition of these seven next-generation satellites, the O3b mPOWER system will have more than 30,000 fully shapeable and steerable beam, providing coverage of ±50° latitude for nearly 400 million km^2 [15].

2.6 Kuiper

Amazon confirmed in April 2019 that it has a plan to launch as many as 3236 LEO satellites to cover regions between 56° north and 56° south latitudes where approximately 95% of the world's population lives. The name of the project, "Kuiper", comes from a region of the Solar System, where there are many fragments of protoplanetary disks that surround the Sun, known as the Kuiper Belt.

"Project Kuiper" is made up of LEO satellites rounding the earth at three different altitudes, 784 satellites orbiting at an altitude of 590 km, 1296 satellites orbiting at an altitude of 610 km, and 1156 satellites at an altitude of 630 km. At present, Project Kuiper is still in the initial stage and has just submitted an application to the FCC. The satellite launching will take another few years [16].

3 Development Prospect

1. A wide variety of design options will be explored

Commercial communication satellite systems designed by different companies have different features. Some satellite systems such as Iridium-NEXT and Starlink provide global coverage, while other satellite systems such as O3b mainly cover the middle and low latitudes, thus reducing the satellite number and the system costs. Some satellites have fixed attitude antennas, and others can dynamically adjust the antenna beam according to the user's position. Some satellite networks carry out data transfer through the gateway in the ground, while others using inter-satellite links. Some constellations are composed of LEO, which can reduce the communication delay, while others use MEO in their constellations, which can reduce the number of satellites. Numerous design solutions focus on different needs and have different characteristics. Future communication satellite may absorb the advantages of various design options and continuously improving system performance.

2. The latest communication technology will constantly be adopted

The development of communication technology continues to improve and enhance satellite system performance. Satellite miniaturization will become a trend in the future. The weight of a StarLink satellite is only 227 kg, while the weight of an OneWeb satellite is less than 150 kg. The miniaturization of satellite is convenient for multi-satellites launches, greatly reducing satellite cost and launch cost, and speeding up satellite construction. The adoption of laser communication technology has made it possible to establish a space transmission network which is even faster than the optical cable network. Beam control technology can dynamically adjust satellite beam as needed to meet user needs and minimize the waste of satellite resource. A variety of new technologies will greatly improve the performance of communication satellite systems.

3. The competition for spectrum and orbit resource is becoming increasingly fierce

Spectrum and orbit are the basis and premise for the establishment of a satellite system. They are also the necessary condition for the normal operation of a satellite system. Both spectrum and orbit are non-renewable limited natural resource. As the number of satellites continues to increase, spectrum and orbit resources are becoming increasingly tense. Major communications satellite companies are rushing to declare and acquire spectrum and orbit resources. The competition for spectrum and orbital resources will become increasingly intense in the future.

4 Conclusions

This paper reviews the recent developments in satellite communications and outlines the performance characteristics of several typical commercial satellite systems. Prospects for the development of satellite communications and their applications are expected. The seamless global coverage of satellite communications can yield high practical and social benefits. The world of satellite communications is hotting-up, and the future of high-speed “space Internet” is increasingly bright.

References

1. Pelton JN, Madry S, Camacho-Lara S (2017) Handbook of satellite applications || history of satellite communications. Springer. https://doi.org/10.1007/978-1-4419-7671-0_14
2. Golding LS (1998) Satellite communications systems move into the twenty-first century. Wirel Netw 4:101–107
3. Lutz E, Bischl H, Ernst H et al (2004) Development and future applications of satellite communications. In: IEEE international symposium on personal
4. De Mey S (2016) The future of satellite applications: the end-user. Perspective. https://doi.org/10.1007/978-3-7091-4860-0_6

5. Va M (2019) Iridium completes historic satellite launch campaign: financial and technological transformation nears, with 75 new satellites successfully deployed by SpaceX over the past two years. http://investor.iridium.com/2019-01-11-Iridium-Completes-Historic-Satellite-Launch-Campaign
6. LeoSat (2019) A new type of satellite constellation. http://leosat.com/technology/
7. OneWeb (2019) OneWeb satellite stats. https://www.oneweb.world/technology#keyshot-module
8. Carter J (2019) Everything you need to know about SpaceX's Starlink plans for 'space internet'. https://www.techradar.com/news/everything-you-need-to-know-about-spacexs-starlink-plans-for-space-internet
9. Boyle A (2018) SpaceX launches its first Starlink internet satellites (and Spain's Paz satellite, too) https://www.geekwire.com/2018/spacex-paz-starlink-satellite/
10. Peck AJ (2019) A look at SpaceX's Starlink satellite internet initiative: all the constellation, none of the astrology. https://techreport.com/review/34605/a-look-at-spacex-starlink-satellite-internet-initiative
11. SpaceX (2019) Starlink mission. https://www.spacex.com/webcast
12. Spaceflight101 (2019) O3b satellite overview. http://spaceflight101.com/spacecraft/o3b/
13. SES (2019) O3b MEO low-latency broadband unlocks opportunities. https://www.ses.com/networks/networks-and-platforms/o3b-meo
14. SES (2019) Bold vision. Proven impact. https://www.ses.com/newsroom/bold-vision-proven-impact
15. SES (2017) SES opens new era in global connectivity with O3b mPOWER. https://www.ses.com/press-release/ses-opens-new-era-global-connectivity-o3b-mpower
16. Buncombe A (2019) Project kuiper: amazon set to launch 3,000 satellites to offer broadband to "tens of millions" of people. https://www.expertreviews.co.uk/broadband/1408993/amazon-satellite-broadband-project-kuiper-systems

Feature-Aware Adaptive Denoiser-Selection for Compressed Image Reconstruction

Mengke Ma, Dongqing Li, Shaohua Wu(✉), and Qinyu Zhang

Harbin Institute of Technology, Shenzhen, Guangdong, China
{mamengke, lidongqing}@stu.hit.edu.cn
{hitwush, zqy}@hit.edu.cn

Abstract. Compressed sensing (CS) has been extensively studied in image processing; however, the ill-posed inverse problem in the decoder is complicated, still leaving room for further improvement. Denoising-based approximate message passing (D-AMP) is a fast CS reconstruction algorithm that can transform the complex reconstruction problem into a classic denoising problem. But the existing denoisers are favorable on images with special features, so the D-AMP algorithm which has a fixed denoiser cannot universally achieve the best reconstruction quality for all types of images. In this paper, we propose a CS image reconstruction framework which can adaptively select the proper denoiser, and a feature-aware denoiser-selection method which extracts features by fast Fourier transform (FFT). Results show that the proposed feature-aware adaptive denoiser-selection-based CS image reconstruction method can obtain the best reconstruction quality for different types of images.

Keywords: CS reconstruction · Adaptive denoiser-selection · Feature extraction

1 Introduction

Compressed sensing (CS) [1,2] is a new compression method of signal sampling which breaks through the limitation of Nyquist sampling theorem. CS reduces the sampling complexity of the encoder by increasing the complexity of decoder. Over the last decade, the application of CS in image processing has gained considerable popularity, among which the reconstruction algorithm of decoder is very important, which directly determines the application prospect of CS [3]. Typical CS reconstruction algorithms can be cast as iteratively estimating signals from perturbed observations [4]. Denoising-based approximate message passing (D-AMP) algorithm can integrate various existing denoisers in the iteration and

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 537–545, 2020.
https://doi.org/10.1007/978-981-15-0187-6_64

transform the complex CS reconstruction problem into a classic denoising problem. The denoising performance directly affects the quality of CS reconstruction [5]. At present, there is no denoiser that is universally applicable to all types of images. The mainstream denoisers can be roughly divided into three categories: non-local self-similarity (NSS) prior-based, convolutional neural network (CNN)-based and combine NSS-based and CNN-based methods. The NSS-based methods are favorable on images with regular and repetitive patterns. Since the priors are based on human observation and contain some parameters that have to be tuned by users, it is difficult to find the irregular priors implied in the images [6,7]. The CNN-based methods perform better on irregular structures. However, the patch-based denoising is a local processing and the existing neural network methods do not consider the NSS prior [7,8]. The combine NSS-based and CNN-based methods show advantages for both regular and irregular priors. Since the block matching number of the trained network structure is fixed, the performance of the combine-based method is weaker than the NSS-based method for strong regular feature images. In addition, the network part only simulates the filtering process, so the performance of the combine-based method is weaker than the CNN-based method for strong irregular feature images [7].

In this paper, aiming at the issue that the denoiser is not universal, we propose a CS image reconstruction strategy that can adaptively select the denoiser for different types of images. We note that regular texture and irregular smoothing are two features that universally exist in natural images [9]. Based on the CS measurement image, we design a method that first uses fast Fourier transform (FFT) to extract image features and then applies an adaptive denoiser-selection reconstruction algorithm to different types of images. The main contributions of this paper are summarized as follows:

- We propose a CS image reconstruction framework which can select denoisers adaptively based on the feature extraction from CS measurement.
- We propose a feature-aware adaptively denoiser-selection method which is based on the features extracted by FFT.

The rest of the paper is organized as follows. Section 2 introduces the compressive sensing theory. In Sect. 3, we introduce the D-AMP framework and the typical denoisers. Section 4 introduces the proposed feature-aware adaptively denoiser-selection strategy. In Sect 5, we display the simulation results of our research to demonstrate the effectiveness of the compressed image reconstruction method which can select denoiser adaptively by feature-aware. Conclusion is given in the last section.

2 System Model of CS Image Reconstruction

In this section, we briefly introduce the system model of CS image reconstruction.

In the encoder, CS theory reduces the design difficulty of acquisition equipment and storage equipment, because the simple linear measurement is adopted to combine the sampling and compression processes. In the decoder, CS theory needs to solve the complex ill-posed inverse problem to reconstruct the original signal which can be formulated as Eq. (1).

$$\arg\min_{x} \|y - Ax\|_2^2, \tag{1}$$

where x is the target image, A is the measurement matrix, y is the linear measurements of x, we use $\hat{x}$ to denote the estimate of x and $\hat{y}$ to denote the estimate of y.

Therefore, the reconstruction problem has become the focus of the system research. The block diagram of the image CS system is illustrated in Fig. 1.

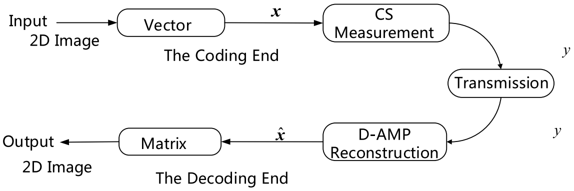


Fig. 1. Block diagram of the image CS system.

3 Denoiser-Based CS Image Reconstruction

In this section, we first describe the framework of D-AMP algorithm briefly and then present the three typical denoisers and their results.

3.1 The D-AMP Framework

Iterative framework is popular in CS theory for its simple iterative process and low computation complexity. A fast iterative framework, D-AMP, cast the CS reconstruction as iteratively estimating signals from perturbed observations. D-AMP employs a denoiser D in the following iteration:

$$\begin{aligned} &x^{t+1} = D_{\hat{\sigma}^t}(x^t + A * z^t), \\ &z^t = y - Ax^t + \tfrac{z^{t-1}\operatorname{div} D_{\hat{\sigma}^{t-1}}(x^{t-1} + A * z^{t-1})}{m}, \\ &(\hat{\sigma}^t)^2 = \tfrac{\|z^t\|_2^2}{m}, \end{aligned} \tag{2}$$

where x^t is the estimate of the true signal at iteration t and z^t is an estimate of the residual. $x^t + A * z^t$ can be considered as i.i.d. Gaussian noise. $\hat{\sigma}^t$ is an estimate of the standard deviation of the noise. $\operatorname{div} D_{\hat{\sigma}^{t-1}}$ denotes the divergence of the denoiser.

Each iteration of the D-AMP algorithm can be regarded as a denoising operation, so the performance of the denoiser directly affects the final reconstruction result.

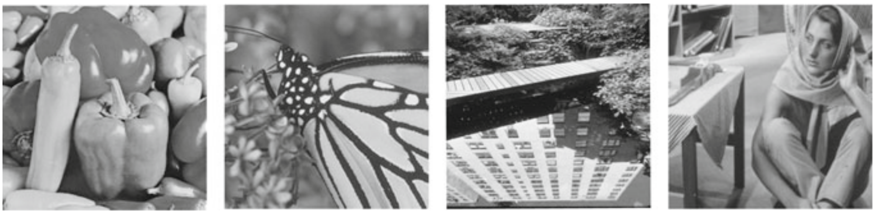
3.2 Typical Denoiser and Their Results

The image denoising model can be expressed as the following optimization problem:

$$\arg\min_{\hat{x}} \|z - \hat{x}\|_2^2, \tag{3}$$

where z is the noise image, x is the true image. The goal of the denoiser is to find the closest estimate $\hat{x}$ to x.

There are three main streams in up-to-date image denoisers, here we introduce one typical denoiser for each.



(a) Peppers (b) Monarch (c) Building (d) Barbara

Fig. 2. Test images.

BM3D is a classical image denoising algorithm based on non-local self-similarity, which can effectively remove noise and maintain regular texture features. As is shown in Eq. (4), BM3D denoises the image by the collaborative filtering of similar patches ($z_{3\mathrm{D}}$), which is realized by the threshold (γ) in 3D transform ($\tau_{3\mathrm{D}}$) domain. Then, we get the denoised image by inverse 3D transform ($\tau_{3\mathrm{D}}^{-1}$).

$$\hat{x}_{3\mathrm{D}} = \tau_{3\mathrm{D}}^{-1}(\gamma(\tau_{3\mathrm{D}}(z_{3\mathrm{D}}))). \tag{4}$$

DnCNN is an image denoising algorithm based on convolutional neural network, which can best use the irregular realistic signal features by learning from the training data. As is shown in Eq. (5), DnCNN separates the noise from the noisy image by the feedforward residual network which uses noisy images (z) as the input and residual images as the output ($\phi(z;\theta)$). θ denotes the weights of the network.

$$\hat{x} = z - \phi(z;\theta). \tag{5}$$

BMCNN is an image denoising algorithm which shows advantages in both regular and irregular features through combining the non-local self-similarity and neural network. As is shown in Eq. (6), BMCNN simulates the 3D collaborative filtering process of similar patches by residual network ($\phi(z_{3\mathrm{D}};\theta)$) to obtain the estimation of the reference patch (z_r).

$$\hat{x} = z_r - \phi(z_{3\mathrm{D}};\theta). \tag{6}$$

Figure 2 shows four images in which the texture features are different. Additive white Gaussian noise with a standard deviation of 50 is added to them, and three denoising algorithms are used to denoise, respectively.

The denoising results are shown in Table 1 which are evaluated by peak signal-to-noise ratio (PSNR). As we can see, BM3D is favorable on images with regular features, DnCNN performs better on irregular features, and BMCNN shows advantages for both regular and irregular features but when the regular or irregular features are strong BMCNN performs worse than BM3D or DnCNN. In a word, there have no denoiser can obtain best performance for all types of images.

Table 1. PSNR and optimal denoiser of standard images denoising.

	Peppers	Monarch	Building	Barbara
DnCNN	**27.34**	**26.64**	22.57	25.24
BMCNN	27.21	26.51	**22.67**	26.90
BM3D	26.77	25.73	22.13	**27.15**
OPTIMAL	DnCNN	DnCNN	BMCNN	BM3D

The bold indicates the best effect of applying three methods to denoise the same image

4 The Proposed Adaptive Denoiser-Selection Strategy

In this section, we proposed a strategy for adaptive-selection of denoisers that can obtain the best reconstruction performance for different types of images.

As is shown in Fig. 3, we select the appropriate denoiser based on the features extracted from the CS measurement and fuse it into D-AMP to obtain the best reconstructed image.

The specific selection method is shown in Fig. 4. Firstly, the noise-robust FFT is applied to the measurement image and the feature extraction is carried out for two main features: regular texture (high-frequency signal) and irregular smoothing (low-frequency signal) which universally exist in natural images. We extract the proportions of the low-frequency components in the central region of the transform domain image (P) and set two thresholds. When $P >$ Thr1, it indicates that irregular smoothing feature is strong, and DnCNN is selected. When $P <$ Thr2, it indicates that regular texture feature is strong, and BM3D is selected. Otherwise, the BMCNN is selected.

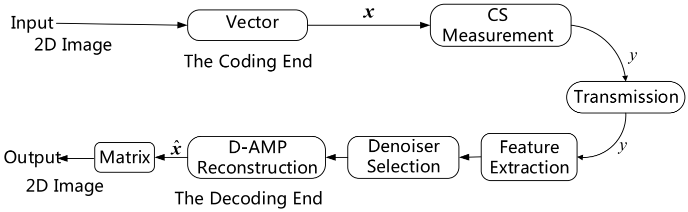


Fig. 3. Framework of feature-aware adaptive denoiser-selection for compressed image reconstruction.

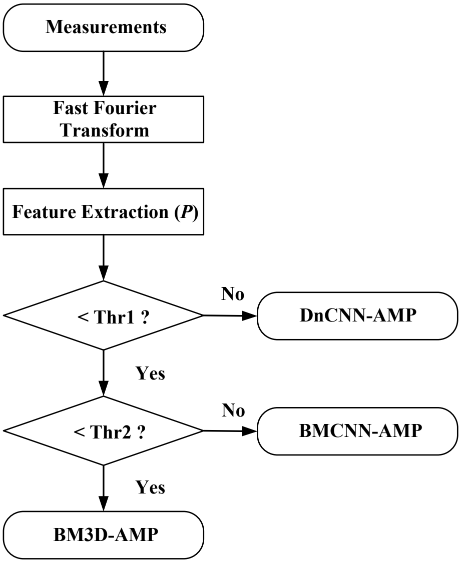
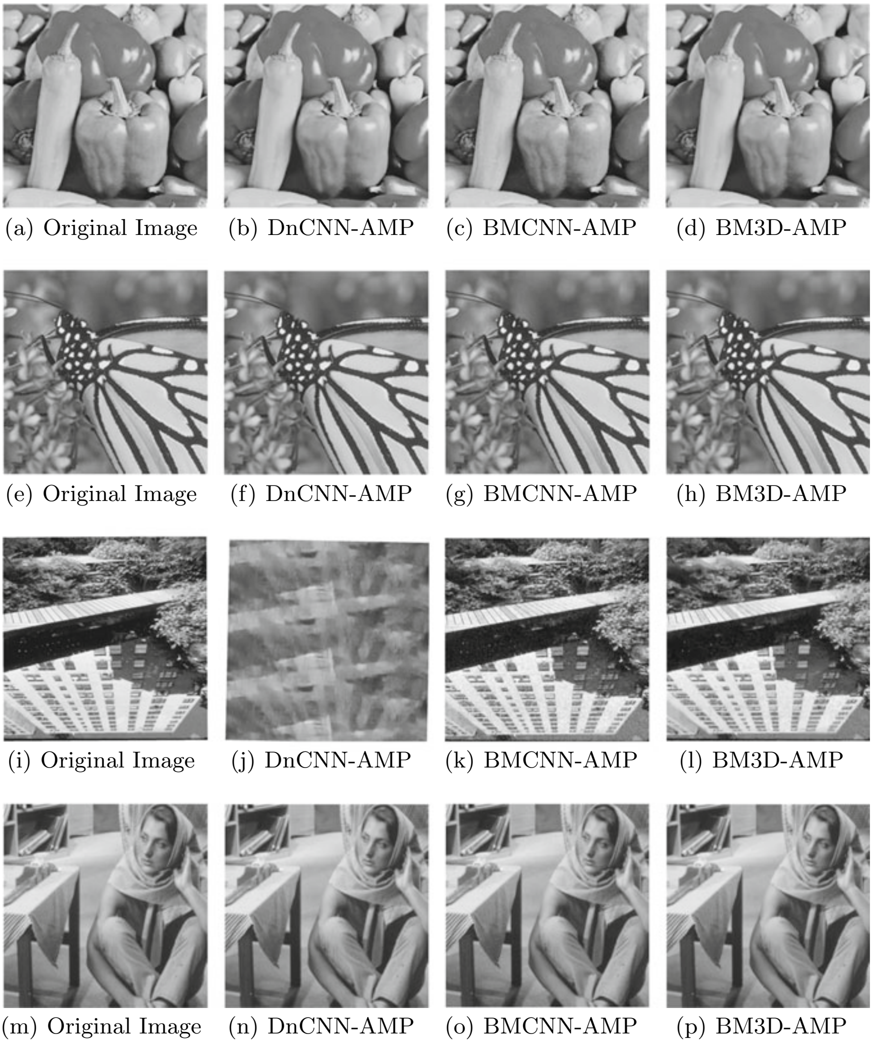


Fig. 4. Selection flowchart of feature-aware adaptive denoiser-selection for compressed image reconstruction.



(a) Original Image (b) DnCNN-AMP (c) BMCNN-AMP (d) BM3D-AMP

(e) Original Image (f) DnCNN-AMP (g) BMCNN-AMP (h) BM3D-AMP

(i) Original Image (j) DnCNN-AMP (k) BMCNN-AMP (l) BM3D-AMP

(m) Original Image (n) DnCNN-AMP (o) BMCNN-AMP (p) BM3D-AMP

Fig. 5. The reconstruction results on a sampling ratio of 0.5.

5 Results and Discussion

To evaluate the performance of the proposed method, we compared the reconstruction quality of D-AMP algorithm combined with different denoisers. We use four images (Peppers, Monarch, Building, Barbara) borrowed from standard test images as in Fig. 2 and set the sampling rate to 0.5. The metric results are shown in Table 2. Figure 5 shows the reconstruction images of D-AMP with different denoisers in visual perception of test images in Fig. 2.

As we can see from Table 2, different types of images are beneficial from different algorithms and can be clearly distinguished according to the features

Table 2. PSNR and optimal method of standard images reconstruction.

50% sampling	Peppers	Monarch	Building	Barbara
DnCNN-AMP	**38.83**	**42.32**	11.40	38.71
BMCNN-AMP	37.37	37.78	**23.58**	36.74
BM3D-AMP	37.55	38.53	22.47	**39.03**
PROPORTION	*2.69*	*2.74*	*1.12*	*0.70*
OPTIMAL	DnCNN-AMP	DnCNN-AMP	BMCNN-AMP	BM3D-AMP

The bold indicates the best effect of applying three methods to reconstruct the same image

Italic represents the proportion of smoothing information calculated after FFT transformation of each image, which is the basis for selecting the best method compared with the two thresholds

The black bold represents the method selected by the selection strategy, which is consistent with the best method, indicating that the selection strategy is effective

extracted by FFT (low-frequency signal ratio). According to the metric (PROPORTION) in Table 2, the Thr1 is set to 2, and the Thr2 is set to 1. Therefore, the CS reconstruction algorithm that selects the appropriate denoiser based on feature-aware can obtain the best performance in different types of images.

Here, we mainly provide a general idea of feature extraction. The selected threshold is an empirical value and can be appropriately adjusted according to different image sets.

6 Conclusion

In this paper, we propose a CS reconstruction framework that can adaptively select denoisers by extract features from CS measurements. Specifically, the selection method is feature-aware where the feature is extracted by FFT. Under this strategy, we can select the appropriate denoiser and combine it with D-AMP algorithm to obtain the best reconstruction results for images with different features.

Acknowledgements. This work has been supported in part by the National Natural Science Foundation of China under Grant nos. 61871147, 61831008, and in part by the Shenzhen Municipal Science and Technology Plan under Grant nos. JCYJ20170811160142808, JCYJ20170811154309920.

References

1. Baraniuk RG (2007) Compressive sensing [lecture notes]. IEEE Signal Process Mag 24(4):118–121
2. Donoho DL (2006) Compressed sensing. IEEE Trans Inf Theory 52(4):1289–1306
3. Huang H (2017) Compressed image sensing reconstruction by jointly leveraging structural and statistical priors. Harbin Institute of Technology, Shenzhen
4. Metzler CA, Maleki A, Baraniuk RG (2014) From denoising to compressed sensing. IEEE Trans Inf Theory 62(9):5117–5144

5. Metzler CA, Mousavi A, Baraniuk R (2017) Learned D-AMP: principled neural network based compressive image recovery. In: 31st conference on neural information processing systems, NIPS, Long Beach, CA, USA
6. Dabov K, Foi A, Katkovnik V, Egiazarian K (2007) Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans Image Process 16(8):2080–2095
7. Ahn B, Cho N (2017) Block-matching convolutional neural network for image denoising. arXiv: 1704.00524
8. Zhang K, Zuo W, Chen Y, Meng D, Zhang L (2017) Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans Image Process 26(7):3142–3154
9. Huang H, Wu SH, Zhang T, Cao B, Zhang Q (2017) Image compressed sensing reconstruction by collaborative use of statistical and structural priors. In: IEEE 85th vehicular technology conference (VTC, spring)

AI-Assisted Complex Wireless Network Evaluation Using Dynamic Ranking Scheme

Yuan Gao[1,2,3], Jiang Cao[1](✉), Weigui Zhou[2], Chao Qing[2], Hong Ao[2],
Su Hu[3], Junsong Yin[1], ShuangShuang Wang[1], Xiangyang Li[1],
and Haosen Yu[4]

[1] Academy of Military Science of the PLA, Beijing 100142, China
caojiang60@foxmail.com
[2] Xichang Satellite Launch Center, Sichuan 615000, China
[3] University of Electronic Science and Technology of China, Sichuan 611731, China
[4] International College, Chongqing University of Posts & Telecommunications, Chongqing 400065, China

Abstract. In recent development of communication systems, increasing amount of mobile terminals and multimedia content will require more and more resources to satisfy the need of users. However, there is rare study about the evaluation of large-scale complex network, from which the shortage of the network could be discovered. In this paper, we discuss the AI-based network evaluation method, we propose a novel discovery and ranking system using deep learning to collect and evaluate the network influence factor, and then, the key factors will be discovered to detect the advantages and shortages of network elements, deployments and scale. The AI-based detection and evaluation system is running along with the LTE-A system-level simulation platform, the accuracy and the effectiveness are evaluated, system shortages are successfully discovered within 100 trainings.

Keywords: AI · Complex · Wireless network · Dynamic ranking

1 Introduction

The breakthrough of fifth-generation wireless communication system has brought significant improvement in our daily life [1–3]. More and more types of terminals are connected to Internet providing varieties of information. One significant different compared to previous wireless communication system is the complexity, caused by the advanced signal processing or information fusion methods, and the huge number of network elements and data [4, 5], where existing methods could not tackle the enormous scale of the evaluation.

Performance evaluation of complex network is still a hard topic in both academic and industry. Existing evaluations are mainly focused on some dedicated points, for

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 546–553, 2020.
https://doi.org/10.1007/978-981-15-0187-6_65

example, in some given scenarios, the scale of the network is limited that means the elements of the network are small, typically less than 100, so the evaluation of the network is simple under the ability of computers and methods. In large-scale complex networks, both academic and industry have launched some related researches to get the precise estimation of the network. In [6], the authors provide the evaluation in WLAN and 3G hybrid networks, to help select proper network and transmit data. In [7], the virtualization of network is given, and the evaluation in core network is performed under the digital modeling. In [8], the topology of the network is modeled complex considering the wireless sensor scenario, a large number of wireless nodes with low network requirements are modeled, but the scale and the complexity are not satisfied. In [9], modeling of data traffic is given in hybrid network is given, and performance evaluation based on mathematical traffic modeling is also presented. In [10], the evaluation of multichannel using back-off method is given, but the scale is not suitable for complex network evaluation. Then, in [11, 12], the authors provide some attempts in evaluation of large-scale wireless networks, but the statistics methods seem complex. In [13], SDN and RPL in wireless networks are evaluated based on the statistical method, in [14], to tackle the problem of scale, the author provides the machine learning way to solve the evaluation, and similarly, authors in [15–17] are trying to evaluate the complex network using deep learning-based intelligent methods.

In this paper, to solve the problem of evaluation in large-scale networks, we propose the artificial intelligent (AI)-based method to simplify the evaluation using parameter detection and ranking. The rest of the paper is organized as follows. In Sect. 2, we provide the system model of complex wireless networks and the difficulties in evaluation; in Sect. 3, we provide the methods of evaluation using deep learning, and then, in Sect. 4, we provide the simulation and analysis. Conclusions are drawn as summary.

2 System Model

The development of the fifth- and the sixth-generation wireless networks provides significant improvement of network scale and the number of mobile terminals. The evaluation of complex networks is quite difficult because the related parameters are quite huge to compute.

In Fig. 1, one example of complex wireless networks is given. As is described above, the scale of the network is one possible difficulty to evaluate the network performance. It is clear in this figure that the topology is scaled into three main parts. In cloud part, there are mainly the server and signal processing units, which will be the content provider and the cloud signal processing. In this part, many types of data traffic will be gathered, due to the requests from terminal users, signal processing in cloud side will decide the optimal resource allocation and transmission paths to reduce the waste of energy and system resource. In edge side, wireless transmission equipment such as eNodeBs, wireless access points (WAP), wireless remote radio head (RRH) is working in this part. In 5G and related advanced wireless networks, number of equipment in edge side has become huge because of the development of MIMO, network slicing, etc. In this part, main signal processing part has been transferred to

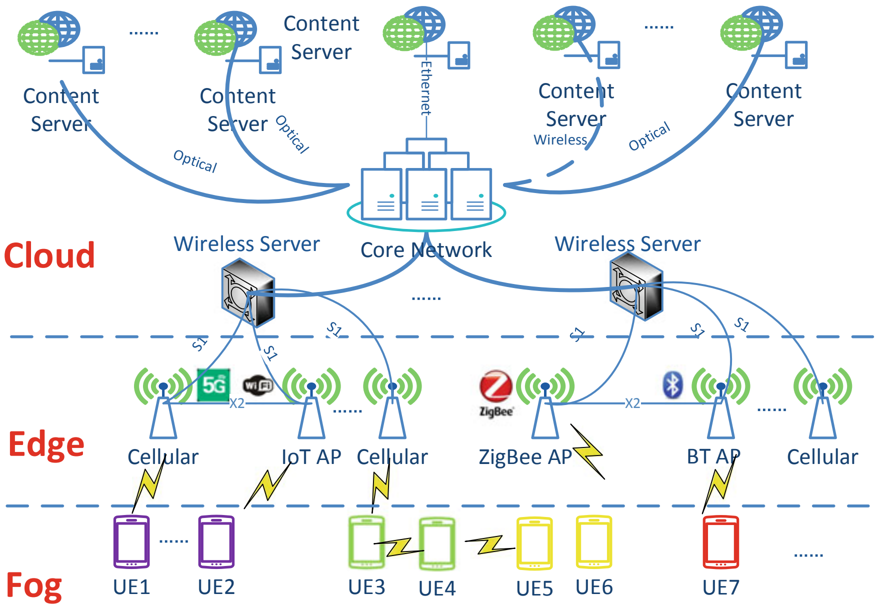


Fig. 1. Structure of complex wireless networks. cloud–edge–fog structure is given according to 5G network.

core network, and only some infrastructure is performing the transmission. In fog side, the number of terminals is many times compared to other two sides, and all IoT terminals, multimedia laptops, Internet of Vehicles (IoV), etc., are connected among each other and the eNodeBs, which will bring significant pressure to the whole network.

From the discussion above, we can infer that the difficulties to evaluate the large-scale complex network could be summarized as follows:

1. Number of nodes are huge, which makes the difficulty of summarizing the parameters and decides which parameters are important [18].
2. Data transmitted through the line is also complex, and it is difficult to discover which data is important.
3. The connection of network consists of multiple standard, and the fusion between different types of communication will also bring delay, fault, etc. How to evaluate the accuracy and the robust of data is still an open hard problem.

3 Generation of Deep Learning Network for Ranking

To tackle the evaluation of complex large-scale networks, we propose a novel deep learning-based network parameter discovery and ranking scheme using the neural network.

The novel design in this work is the five-layer deep neural network given in Fig. 2. We establish a eight-layer with 256 by 256 blank matrix for parameter collection, all the running parameters will be filled without order, then the sorting will be performed, the space is compressed to eight-layer 128 by 128, the parameters and data will be ordered by address, time and relevance, and redundant information will be deleted. Note that the space in this step is half compared to the previous step, and if there is not enough space, information with high relevance will be deleted. The next step is convolution, in this layer, the space is 24-layer 64 by 64 space, for data management, then in frequency layer, the data is absorbed by frequency with 24-layer 16 by 16 size, the last layer is optimization, data is optimized to a vector with the size of 1 by 256, and the last result is a 1 by 16 vector.

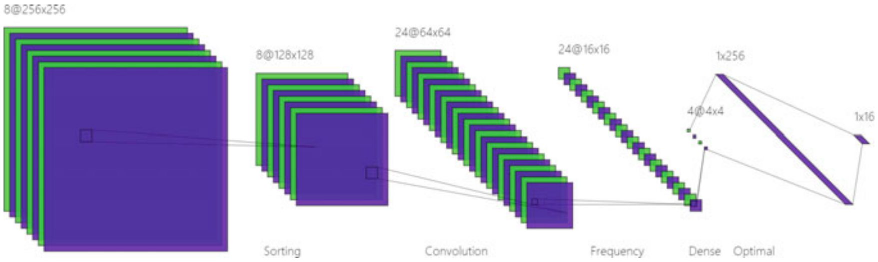


Fig. 2. Five-layer deep learning network for network parameter discovery and ranking.

For any given layer in step 1, element of matrix is defined by the following equation:

$$\mathrm{net}_{hi} = \sum_{i,j} w_{1i} * i_1 + b_\mathrm{j} * 1 \tag{1}$$

For hidden layers located in the middle, network output is defined by Eq. 2, and the output of the given net is

$$\mathrm{net}_{oi} = \sum_{i,j} w_i * \mathrm{out}_{hi} + b_j * 1 \tag{2}$$

$$\mathrm{out}_{oi} = \frac{1}{1 + \mathrm{e}^{-\mathrm{net}_{oi}}} \tag{3}$$

And the update of hidden layer is defined using the following optimization equation:

$$\frac{\partial E_\mathrm{total}}{\partial w_\mathrm{i}} = \left(\sum\nolimits_{o,i} \frac{\partial E_o}{\partial \mathrm{out}_o} * \frac{\partial \mathrm{out}_o}{\partial \mathrm{net}_o} * \frac{\partial \mathrm{net}_o}{\partial \mathrm{out}_{hi}} \right) * \frac{\partial \mathrm{out}_{hi}}{\partial \mathrm{net}_{hi}} * \frac{\partial \mathrm{net}_{hi}}{\partial w_i} \tag{4}$$

4 Simulation Analysis

In this section, we propose a large-scale complex networks containing 1,00,000 terminals, and all of them are freely connected to each other if necessary. Types of the terminals are eNodeBs, IoT devices, signal processing cloud units, etc. Assume the traffic model is Poisson, type of terminals is five, maximum connections for each nodes are ten, and there is no cooperation and information exchange for data fusion. In Fig. 3, we displayed the generated neural network in this scenario.

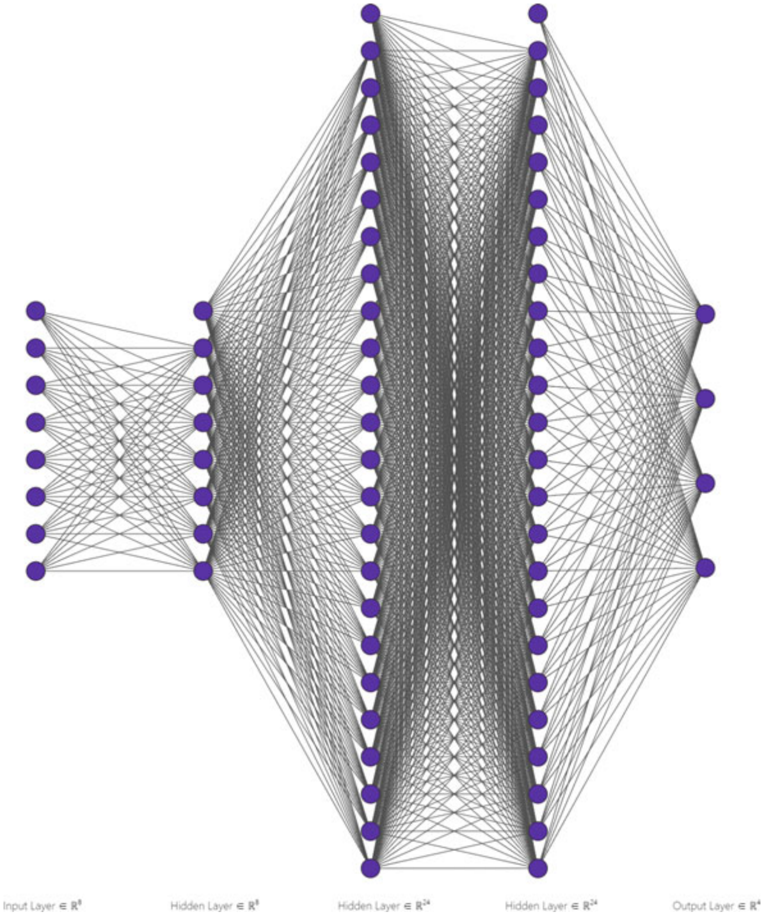

Input Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^8$
Hidden Layer $\in \mathbb{R}^{24}$
Hidden Layer $\in \mathbb{R}^{24}$
Output Layer $\in \mathbb{R}^4$

Fig. 3. Generated neural network with five layers to discover and rank the importance of parameters.

The model of the network is established using 3GPP Rel-13, the simulation platform is 3GPP LTE-R13 system-level simulation platform, and scenario is set to big match. In Fig. 4, we provide the evaluation from four°: the connection, transmission,

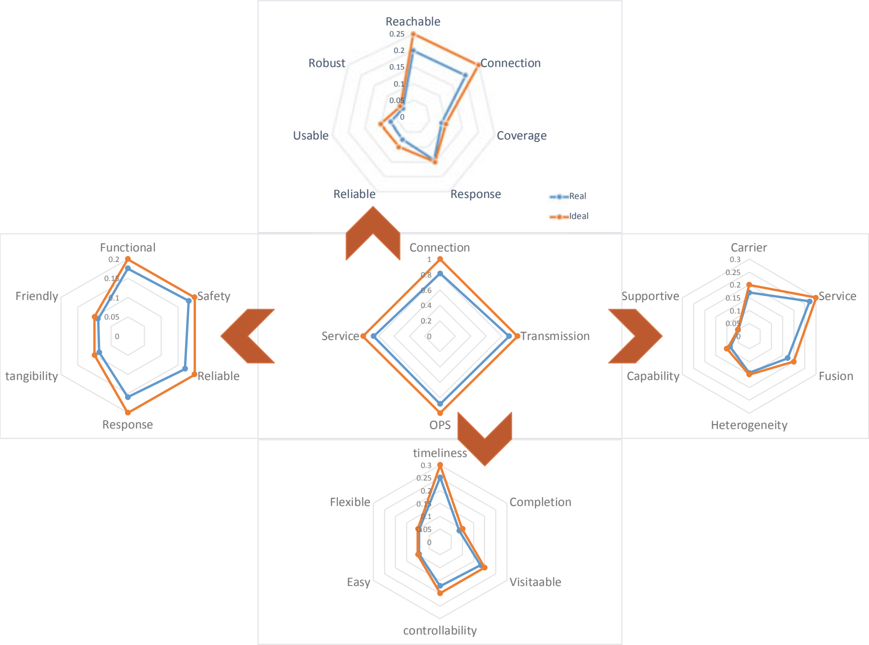


Fig. 4. Evaluation and ranking using deep learning

OPS and service. For each degree, it is expanded to the second-layer parameters with ranking summarized from the deep learning parameter discovery and ranking process. For instance, the connection is divided into seven sub-parameters: reachable, connection, coverage, response, reliable, usable, robust, and the third-level parameters could be discovered during the process of deep learning.

From the process, we can infer that no matter how many the parameter is, the only work we need to do is to establish the space for all candidate parameters, from the deep neural networks, all parameters with frequency and data relations will be generated, and the evaluations could be converged quickly to the target.

5 Conclusion

In this paper, considering the background of large-scale complex networks, we propose one possible solution using deep learning to detect the parameters from network and rank the most important items and provide the accurate evaluation. A layered neural network is generated and successfully evaluated the 5G-based large-scale network under system-level simulation platform. However, this work is a prior work to the hard evaluation method, and there is still some unclear and ideal assumptions during the simulation.

Acknowledgements. This work is funded by National Nature Science Foundation of China under grant of 61701503. The author would also like to thank all the reviewers, and their suggestions help improve my work a lot.

References

1. Zhang P, Yang X, Chen J, Huang Y (2019) A survey of testing for 5G: solutions, opportunities, and challenges. China Commun 16(1):69–85
2. Muirhead D, Imran MA, Arshad K (2016) A survey of the challenges, opportunities and use of multiple antennas in current and future 5G small cell base stations. IEEE Access 4:2952–2964
3. Al-Falahy N, Alani OY (2017) Technologies for 5G networks: challenges and opportunities. IT Prof 19(1):12–20
4. Taufique A, Jaber M, Imran A, Dawy Z, Yacoub E (2017) Planning wireless cellular networks of future: outlook, challenges and opportunities. IEEE Access 5:4821–4845
5. Prasad KNRSV, Hossain E, Bhargava VK (2017) Energy efficiency in massive MIMO-Based 5G networks: opportunities and challenges. IEEE Wirel Commun 24(3):86–94
6. Wang H, Laurenson DI, Hillston J (2013) A general performance evaluation framework for network selection strategies in 3G-WLAN interworking networks. IEEE Trans Mob Comput 12(5):868–884
7. Rofoee BR et al (2015) Hardware virtualized flexible network for wireless data-center optical interconnects [invited]. IEEE/OSA J Opt Commun Netw 7(3):A526–A536
8. Shahzad F, Sheltami TR, Shakshuki EM (2016) Multi-objective optimization for a reliable localization scheme in wireless sensor networks. J Commun Netw 18(5):796–805
9. Wu Y, Min G, Yang LT (2013) Performance analysis of hybrid wireless networks under bursty and correlated traffic. IEEE Trans Veh Technol 62(1):449–454
10. Ng B, Law KLE, Lam CT (2015) Performance evaluation of multi-channel wireless backoff algorithm. IEEE Commun Lett 19(12):2222–2225
11. Guan J, Cao L, Chen W, Cheng X, Xu L, Cao X (2016) A comprehensive method of evaluation for wireless network operation stability. In: 2016 16th international symposium on communications and information technologies (ISCIT), Qingdao, pp 342–346
12. Hussain Z, Karvonen H. Iinatti J (2017) Energy efficiency evaluation of wake-up radio based MAC protocol for wireless body area networks. In: 2017 IEEE 17th international conference on ubiquitous wireless broadband (ICUWB), Salamanca, pp 1–5
13. Tsapardakis E, Ojo M, Chatzimisios P Giordano S (2018) Performance evaluation of SDN and RPL in wireless sensor networks. In: 2018 global information infrastructure and networking symposium (GIIS), Thessaloniki, Greece, pp 1–5
14. Ricky MY, Hendric SWHL, Budiharto W, Abbas BS (2017) Machine learning approach to task ranking. In: 2017 14th international symposium on pervasive systems, algorithms and networks & 2017 11th international conference on frontier of computer science and technology & 2017 third international symposium of creative computing (ISPAN-FCST-ISCC), Exeter, pp 507–513
15. Sakai Y, Oda T, Ikeda M, Barolli L (2016) Performance evaluation of an accessory category recognition system using deep neural network. In: 2016 19th international conference on network-based information systems (NBiS), Ostrava, pp 437–441
16. Han K, Hwang A, Lee JY, Kim BC (2018) Design and performance evaluation of enhanced congestion control algorithm for wireless TCP by using a deep learning. In: 2018

international conference on electronics, information, and communication (ICEIC), Honolulu, HI, pp 1–2

17. Thar K, Oo TZ, Tun YK, Kim DH, Kim KT, Hong CS (2019) A deep learning model generation framework for virtualized multi-access edge cache management. IEEE Access 7:62734–62749
18. Sasaki H, Igarashi H (2019) Topology optimization accelerated by deep learning. IEEE Trans Magn 55(6):1–5, Article No 7401305

Re-sculpturing Semantic Web of Things as a Strategy for Internet of Things' Intrinsic Contradiction

Jingmin An[1](✉), Guanyu Li[2](✉), Bo Ning[2], Wei Jiang[2],
and Yunhao Sun[2]

[1] Faculty of Computer and Software, Dalian Neusoft University of Information,
Dalian, China
870457569@qq.com

[2] Faculty of Information Science and Technology, Dalian Maritime University,
Dalian, China
rabitlee@163.com

Abstract. In this paper, based on annotating objects' information and reasoning semantically with ontologies, the strategy of Semantic Web of Things (SWoT) for Internet of Things' intrinsic contradiction is extended, and machine learning is introduced so that a novel hierarchical structure and the dynamic relationship between entities are proposed on SWoT. Specific contributions: (1) Establish of vertical and hierarchical framework of SWoT; (2) Definite on composite ontologies and (3) construct computational framework of Agent-based dynamic relationship model. And this paper proposes the essence that SWoT should be attributed to the interoperability between its various objects, which constructs inter-objects' certain dynamic relationships.

Keywords: Semantic Web of Things · Interoperability · Composite ontology · Agent · Dynamic relationships

1 Introduction

Since the concept of Semantic Web of Things (SWoT) was put forward, it has been generally believed that SWoT is the result which semantic technology is integrated into the Internet of Things (IoT). In other words, use ontology to annotate the information of objects in IoT, and infer the information in the network based on ontology, in order to achieve global semantic consistency of information.

Ruta and Sciosci [1, 2] point out that SWoT is an organic integration of IoT and Semantic Web (SW). Use pervasive computing to achieve resource sharing in IoT [3], and SW technology can efficiently integrate and reuse resources of IoT. Semantic technology-embedded IoT is utilized to share existing information and generate new information by semantically reasoning. Ontology-based technology of knowledge fusion can be implemented to integrate and share data and information in SWoT [4]. Jia [4] and others reckon that the combination of ontology theory and IoT technology can meet the needs of dynamic and distributed services. Huang and Li [5, 6] deeply analyze

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 554–562, 2020.
https://doi.org/10.1007/978-981-15-0187-6_66

the intrinsic contradictions of IoT, pointing out that the manifestation diversities of things and the inadequate understanding of the users are the key factors, leading to affecting the further development of IoT, and then put forward a strategy for it—SWoT.

Based on these, machine learning is introduced and combined with semantic technology in the paper. Aiming at this research, we design the hierarchical architecture framework of SWoT. And a more specific and rigorous SWoT structural relationship model is constructed; the Agents are extended to the abstract substitutes of all entities, and the interoperability mechanism of the entities is realized equivalently through the dynamic relationships between the Agents. Herein, Agents are divided into two categories: individual-Agent and object-Agent. At the same time, combining with composite ontology, dynamic relationship reasoning models of two kinds of Agents are constructed.

2 Extension of SWoT Model

Construct a corresponding avatar (Agent) for the things (network entities, hereinafter referred to as entities) in SWoT. In this way, SWoT can be considered from two perspectives: (1) at the horizontal level, Agents constitute the network layer, and physical entities constitute the physical layer; (2) at the vertical level, the same category of hierarchy is constituted by each kind of entity and the corresponding Agents. Human-centered construction-based IoT [7] regards serving people as the main purpose, and therefore, in order to be more general, entities are divided into two categories, including the individuals and the objects.

Although layers can be independent each other, they are also interrelated and have loose coupling characteristics. The hierarchies constructed by individuals and corresponding Agents can exist independently. Agent reasoning is used to predict human daily activities and discover interpersonal relationships. Similar mechanisms can also be applied to objects and corresponding agent levels. At the same time, all entities in the real world are connected by events [8, 9], and therefore, the two levels above can influence each other through various events between them.

So, the evolution process of the formal representation model of SWoT is as follows: Entity Set Pattern → Entity Network Pattern → Agent Network Pattern.

Entity set pattern. SWoT= (Ω, N, E, T). Ω = (O, I) represents the entity set in SWoT. Here, O stands for “object,” referring to the entity distinguished from “individual” in SWoT; I stands for “individual;” N stands for “network,” which is the various storage sets of entity information in SWoT; E stands for “event.” T stands for time, which reflects the dynamic signs of SWoT.

Entity Network Pattern. SWoT = (N^{Ω}, E, R, T). $N^{\Omega} = (IoO, IoI) = IoO \oplus IoI = N^{O} \cup N^{I}$ represents the fusion network consisted of entity sets Ω in SWoT. IoO (Internet of Objects) = $O \oplus N = N^{O}$, which is networks of existing objects (e.g., sensor networks, etc.). IoI (Internet of Individuals) = $I \oplus N = N^{I}$, which is social networks (e.g., Wechat, QQ, Blog, etc.). Here, binary operator ⊕ represents fusion; $E = E_{O\otimes O} \cup E_{I\otimes I} \cup E_{O\otimes I \cup I\otimes O}$ is a set of events occurring on $O \cup I$. $E_{O\otimes O}$ is a set of events between objects, $E_{I\otimes I}$ is a set of events between individuals, and $E_{O\otimes I \cup I\otimes O}$ is a set of events between individuals and objects; $R = R_{O\otimes O} \cup R_{I\otimes I} \cup R_{O\otimes I \cup I\otimes O}$

represents relationships of entity in IoO and IoI; Binary operator ⊗ indicates cartesian product, and T represents the time set of all involving events in SWoT.

Agent Network Pattern. SWoT = (N^A, E, R, T). N^A = IoA = IoO_A ∪ IoI_A, and N^A is IoA (Internet of Agents) that includes avatars of IoO and IoI (IoO_A and IoI_A) in SWoT. IoO_A = (O, A_O, E_O, T_O, R_O), O represents the object set in the physical world, and in the network world, A_O is the avatar of O. T_O and R_O indicate time set and event relational set in IoO. IoI_A = (I, A_I, E_I, T_I, R_I), I represents the individual set in the physical world, and in the network world, A_I is the avatar of I. T_I and R_I indicate time set and event relational set in IoI. While R and T, respectively, indicate $R_{AO \otimes AO} \cup R_{AI \otimes AI} \cup R_{AO \otimes AI \cup AI \otimes AO}$ and the time set of all involving events in SWoT.

3 Conceptual Model of SWoT

SWoT has both the essential characteristics of IoT and SW (SWoT= IoT⊕ SW). For solving intrinsic contradiction of IoT, use the characteristic of N^Ω to upload entity ($\Omega = O \cup I$) information to the network and SW = (N^{Onto}, Λ^{Onto}) (ontology network $N^{Onto} = O^{Onto} \cup I^{Onto}$ and use Λ network (Λ^{Onto}) of ontology alignment in N^{Onto}) to achieve semantic annotation and reasoning.

The environment involved in the process includes physical layer and network layer, and the main entities involved are O and I and their information. Different entities exist in different environments, but they depend on each other. As shown in Fig. 1, entities of network layer are abstract representations of ones of physical layer, while instantiation of network layer is physical layer. From the horizontal dimension, SWoT can be divided into network layer and physical layer. Physical layer provides data support for network layer, and network layer provides services for physical layer and realizes the interoperability between physical layer and network layer.

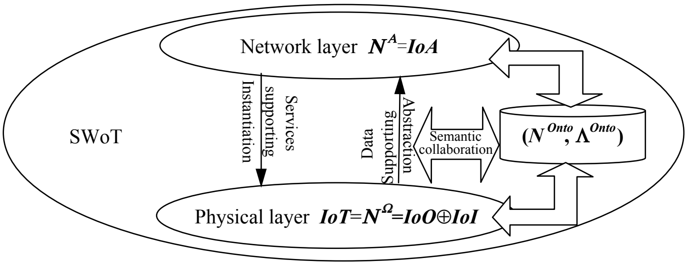


Fig. 1. Hierarchical structure of SWoT horizontal dimension

The Λ^{Onto} is relational representation of interoperability between N^{Onto} ((N^{Onto}, Λ) = Λ^{Onto}) and embodies the core content of SWoT.

At present, in SWoT, the technology of uploading entity information to the network is relatively mature, such as a series of sensor systems or networks, radio frequency identification devices, infrared sensors and global positioning systems. And ontology technology is widely used to process the information in the network. In section above, establish a set of Agents mapping to all entities in the network. Through the interoperability or communication mechanism between Agents, the dynamic relationship between entities can be solved, and the dynamic relationship networks between entities in SWoT can be realized, so as to solve the intrinsic contradiction of IoT.

In summary, in the physical layer of SWoT, each physical entity has a corresponding Agent in the network layer, forming the hierarchical structure of the vertical dimension, shown in Fig. 2. Object set (O) and corresponding Agent set (A_O), as well as the interaction between them, constitute an independent set of IoO, which can independently realize the dynamic relationships between objects and the derivation of deeper relationships. $E_{O \otimes I \cup I \otimes O}$ and $R_{O \otimes I \cup I \otimes O}$ are the carriers of the relationships between IoO and IoI.

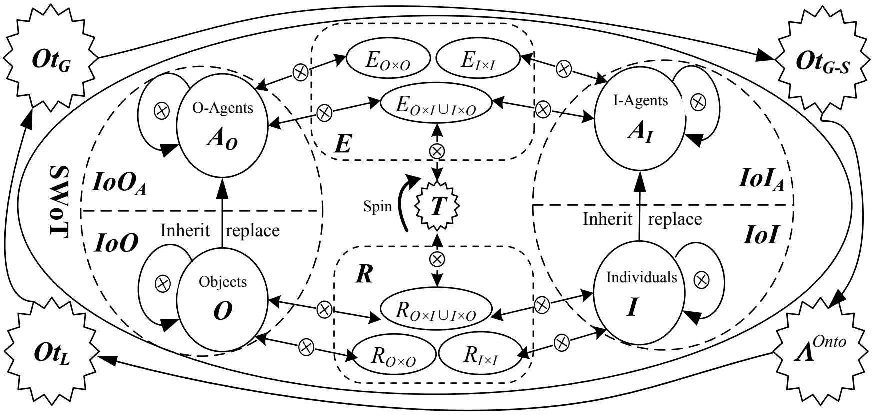


Fig. 2. Hierarchical structure of SWoT vertical dimension

As shown in Fig. 2, the uploaded information of O and I is processed by A_O and A_I, respectively, and then, the result is fed back to the underlying O and I, forming that real-world entities provide data support for the avatar in the network layer. On the contrary, A_O and A_I meet the service needs and finally form a virtuous circle. The explanations of Ot_L, Ot_G, and $Ot_{G\text{-}S}$ are in following section.

4 Reasoning Model of Dynamic Relationships Between Entities

Definition 1 In dictionary, the "relationship" is defined as the state of interaction between things (Thing1 R Thing2), which indicates the relationship R between Thing1 and Thing2. If the state of thing is independent from time, then R is a static relationship.

Definition 2 Dynamic relationship is a relationship that adds time factor to the original one, which is Thing1 (R, T) Thing2. It means that under the influence of time T, Thing1 and Thing2 have relationships, and the relationships will change with the time.

The main works of realizing dynamic relationships are (1) communication and dynamic relationships reasoning between A_I in the same location; (2) communication and dynamic relationships reasoning between A_I and A_O in different locations; (3) reasoning between A_I and A_O in the same region and (4) communication and dynamic relationships reasoning between A_I and A_O in different locations

In order to accomplish the main work, this paper proposes a composite ontology CO (Λ^{Onto}). As shown in Fig. 3, Agent (take A_{O-1}–A_{O-5} and A_{I-1}–A_{I-4} as examples) realizes the above four situations by calling the composite ontology in two independent network structures. The dotted line (---) represents the dynamic relationships between Agents, and the solid line (—) represents the call relationships between Agents and composite ontologies in Fig. 3.

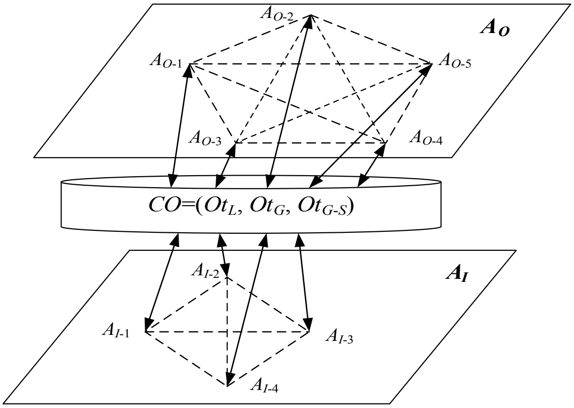


Fig. 3. Framework of agent-to-agent dynamic relationship reasoning

Conveniently, Ot_{O-l} denotes the local ontology corresponding to A_O in region $l \in L$; Ot_{I-l} denotes the local ontology corresponding to A_I in region $l \in L$; Ot_{O-g} and Ot_{I-g} denote the global ontology corresponding to A_O and A_I, respectively.

In Fig. 3, CO is a middleware for providing services for Agent communication and dynamic relationship reasoning; And CO (Λ^{Onto}) is a dynamic ontology (including uncertainty handling) based on event E (illustrated in Fig. 3).

CO is an ontology network, including the local ontology Ot_L ($Ot_{O-l} \cup Ot_{I-l}$) and the global ontology Ot_G ($Ot_{O-g} \cup Ot_{I-g}$), as well as the synthetical global ontology Ot_{G-S}. In fact, it is an interoperability or invocation mechanism model between different regional ontologies and is an ontology structure model to realize the interoperability of Agents (A_I and A_O) in different regions in SWoT, which is the main theoretical means to realize the Agent Network Pattern. At the semantic level, it improves the intrinsic interoperability of SWoT and gives the original static semantic web dynamic nature, resulting in better integration with IoT technology. Ot_L can reason independently to provide semantic technology and data support for Agents in the same region. When in different regions, $\mathrm{Ot_L}$ calls $\mathrm{Ot_G}$ or $\mathrm{Ot_{G\text{-}S}}$ to realize interoperability of the local ontologies and then provides semantic data support for Agent communication in different regions.

In summary, the five-tuple and three-tuple representations of CO are obtained:

$$\mathrm{CO} = \left(\mathrm{Ot_{O-l}}, \mathrm{Ot_{I-l}}, \mathrm{Ot_{O-g}}, \mathrm{Ot_{I-g}}, \mathrm{Ot_{G-S}}\right) \Rightarrow \Lambda^{\mathrm{Onto}} \tag{1}$$

$$\mathrm{CO} = \left(\mathrm{Ot_L}, \mathrm{Ot_G}, \mathrm{Ot_{G-S}}\right) \Rightarrow \Lambda^{\mathrm{Onto}} \tag{2}$$

5 Calling Mechanism of CO

When the Agent obtains the context information of the entity, it calls the local ontology $\mathrm{Ot_L}$ to achieve context semantic analysis and then temporarily stores the parsed semantic information in the local ontology and the global ontology $\mathrm{Ot_G}$ and global synthesis ontology $\mathrm{Ot_{G\text{-}S}}$, respectively, achieving semantic information dissemination finally as shown in Fig. 4.

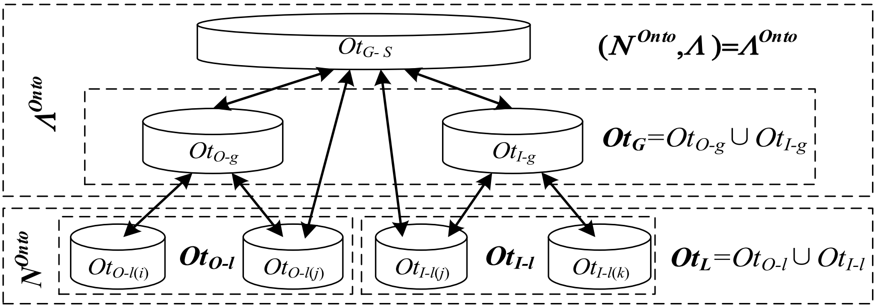


Fig. 4. Principle framework of calling mechanism of CO

The calling mechanism of CO is that the low-level ontology supports the high-level one with (semantic) data, while the high-level ontology feeds back the results of the semantic relationship calculation to the low-level one. Then, $\mathrm{Ot_{O\text{-}l}}$ and $\mathrm{Ot_{I\text{-}l}}$ propagate the semantic information inferred from the contexts to $\mathrm{Ot_{O\text{-}g}}$, $\mathrm{Ot_{I\text{-}g}}$ and $\mathrm{Ot_{G\text{-}S}}$, respectively; Achieve the matching of the semantic information and dynamic relationships between Agents are get.

There are four situations when Agents call CO: (1) $\mathrm{A_O}$ ($\mathrm{A_I}$) in the same region can communicate or interoperate by calling the local ontology $\mathrm{Ot_{O\text{-}l}}$ ($\mathrm{Ot_{I\text{-}l}}$); (2) If $\mathrm{A_I}$ ($\mathrm{A_O}$) comes from different regions, $\mathrm{Ot_{O\text{-}l}}$ ($\mathrm{Ot_{I\text{-}l)}}$ is invoked first, and $\mathrm{Ot_{O\text{-}g}}$ ($\mathrm{Ot_{I\text{-}g}}$) is invoked by the local ontology. And Agents communicate with each other through $\mathrm{Ot_{O\text{-}g}}$ ($\mathrm{Ot_{I\text{-}g}}$); (3) For $\mathrm{A_O}$ and $\mathrm{A_I}$ in the same region, $\mathrm{Ot_{G\text{-}S}}$ is directly invoked to complete the semantic matching calculation and interoperability between them; (4) $\mathrm{A_O}$ and $\mathrm{A_I}$ in different regions call the global ontology through the local ontology first and use the global ontology calls the synthetically global ontology to complete semantic information matching calculation.

In the SWoT, the dynamic relationships between entities are mainly affected by the relative location (L) of entities, the time (T) of interaction between them and the frequency (F) of interaction between them [10–12]. The dynamic relationship $R^k(A_r, A_s) = (D_{rs}^L, D_{rs}^F, D_{rs}^t)$, k = 1, 2, 3, 4 (determination of k is related to L and Agent specific category (Agent_category, denoted as Agc), such as Agc (A_r)::= $(A_O|A_I)$, is based on the dynamic relationships between A_r and A_s (r, s = 1, 2, i, j) at different L and Agc (r, s = 1, 2, i, j); D_{rs}^L, D_{rs}^F and D_{rs}^t are dynamic relationships of L, T and F between Ar and As.

The algorithm for solving the relationship values between Agents is as following:

```
    **Input** *contexts*
    **If**: $L(A_i)=L(A_j)$
    {
**If**: $Match(Ag_c\ (A_i, A_j))==1$
        $A_{O-i}(Ot_{O-l(i)}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij})))\circledcirc A_{O-j}(Ot_{O-l(j)}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij})))\rightarrow R^1$;
**Or**: $A_{I-i}(Ot_{I-l(i)}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij}))\circledcirc A_{I-j}(Ot_{I-l(j)}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij})))\rightarrow R^1$;
**If**: $Match(Ag_c\ (A_i, A_j))==0$
        $A_{O-i}(Ot_{G-S}(D_{ij}^F(F_{ij})), D_{ij}^t(t_{ij})))\circledcirc A_{I-j}(Ot_{G-S}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij})))\rightarrow R^2$;
    }
    **If**: $L(A_i)\neq L(A_j)$
{
 **If**: $Match(Ag_c\ (A_i, A_j))==1$
  $A_{O-i}(Ot_{O-l(i)}(Ot_{O-g}(D_{ij}^F(F_i), D_{ij}^t(t_{ij}), D_{ij}^L(L_{ij}))))\circledcirc A_{O-j}(Ot_{O-l(j)}(Ot_{O-g}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij}), D_{ij}^L(L_{ij}))))$
  $\rightarrow R^3$;
  **Or**: $A_{I-i}(Ot_{I-l(i)}(Ot_{I-g}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij}), D_{ij}^L(L_{ij}))))\circledcirc A_{I-j}(Ot_{I-l(j)}(Ot_{I-g}(D_{ij}^F(F_{ij}), D_{ij}^t(t_{ij}), D_{ij}^L(L_{ij}))))$
  $\rightarrow R^3$;
**If**: $Match(Ag_c\ (A_i, A_j))==0$
  $A_{O-i}(Ot_{O-l(i)}(Ot_{O-g}(Ot_{G-S}(D_{ij}^F), D_{ij}^t(t_{ij}), D_{ij}^L(L_{ij})))))\circledcirc A_{I-j}(Ot_{I-l(j)}(Ot_{I-g}(Ot_{G-S}(D_{ij}^F(F_{ij}),$
  $D_{ij}^t(t_{ij}), D_{ij}^L(L_{ij})))))\rightarrow R^4$;
    }
```

Note: The binary operator ◎ represents the calculation of dynamic relationships between the results of function calling.

6 Calculations of Inter-Agents Dynamic Relationships in SWoT

Combining with the hierarchical relationships of CO interface invocation and the FIPA standard, this paper presents a formula for calculating the weight of dynamic relationship between Agents in SWoT:

$$W_{ij}=\begin{cases} \alpha D_{ij}^{F}+(1-\alpha)D_{ij}^{t}, & (L(A_i)=L(A_j))\wedge((A_i\&A_j)\rhd Ot_L)\ 1 \\ \alpha D_{ij}^{L}+\beta D_{ij}^{F}+(1-\alpha-\beta)D_{ij}^{t}, & (L(A_i)\neq L(A_j))\wedge((A_i\&A_j)\rhd Ot_G)\ 2 \\ \alpha D_{rj}^{F}+(1-\alpha)D_{rj}^{t}, & (L(A_i)=L(A_j))\wedge((A_i\&A_j)\rhd Ot_{G\text{-}S})\ 3 \\ \alpha D_{ij}^{L}+\beta D_{ij}^{F}+(1-\alpha-\beta)D_{ij}^{t}, & (L(A_i)\neq L(A_j))\wedge((A_i\&A_j)\rhd Ot_{G\text{-}S})\ 4 \end{cases}$$

Herein, the operator $\rhd$ representation calls different ontologies depending on the situation.

W_{ij} is the weight between A_i and A_j; α and β are influence factors ($0 < \alpha < 1$, $0 < \beta < 1$), which are the weight of L, F and T. According to the value of the obtained W_{ij}, the network of dynamic relationships between entities can be constructed based on Agents as shown in Fig. 5.

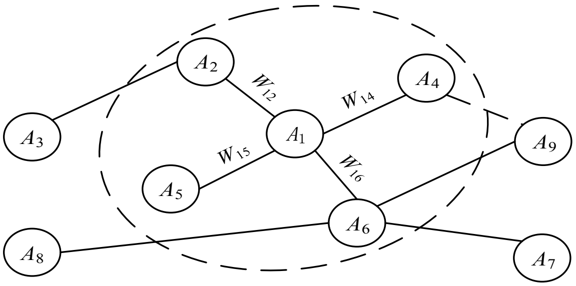


Fig. 5. Diagram of interoperability network between Agents

7 Conclusions

Based on the SWoT view accepted widely by scholars and the solution for the intrinsic contradiction of IoT, this paper extends the research on SWoT. Adopt the technical idea of Agent with CO, and the calling mechanism of CO is designed. In order to realize Agent communication and interoperability instead of entities, the calculation formulas of dynamic relationships between Agents are put forward based on machine learning. Ultimately, solve the intrinsic contradiction of IoT and improve its practicability.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant (No. 61371090). The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper.

References

1. Ruta M, Scioscia F, Pinto A et al (2013) Resource annotation, dissemination and discovery in the semantic web of things: a CoAP-based framework. In: IEEE international conference on green computing and communications and IEEE internet of things and IEEE cyber, physical and social computing. IEEE Computer Society, pp 527–534

2. Ruta M, Scioscia F, Sciascio ED (2012) Enabling the semantic web of things: framework and architecture. In: IEEE sixth international conference on semantic computing. IEEE Computer Society, pp 345–347
3. Wang W, De S, Cassar G et al (2013) Knowledge representation in the internet of things: semantic modelling and its applications. Automatika J Control Meas Electron Comput 54 (4):388–400
4. Yachir A, Djamaa B, Mecheti A et al (2016) A comprehensive semantic model for smart object description and request resolution in the internet of things. Procedia Comput Sci 83:147–154
5. Huang YH, Li GY (2010) Semantic web of things: strategy for internet of things' intrinsic contradiction. Appl Res Comput 27(11):4087–4090
6. Huang YH, Li GY (2011) Internet of things: signature features and model description. Comput Sci 38(b10):4–6
7. Miranda J, Mäkitalo N, Garcia-Alonso J et al (2015) From the internet of things to the internet of people. IEEE Internet Comput 19(2):40–47
8. Brugali D (2015) Model-driven software engineering in robotics: models are designed to use the relevant things, thereby reducing the complexity and cost in the field of robotics. Robot Autom Mag IEEE 22(3):155–166
9. Chen G, Liu HQ (2012) Analysis of the internet of things age from the perspective of the relationship between man and nature. J Econ Res Guide 10:189–190
10. Rusydi MI, Sasaki M, Ito S (2014) calculate target position of object in 3-Dimensional area based on the perceived locations using EOG signals. J Comput Commun 02(11):53–60
11. Wang R, Huang R (2014) Qu B (2014) Network-based analysis of software change propagation. Sci World J 2014:237–243
12. Liu J, Shao Y, Zhu WD (2015) A new model for the relationship between vibration characteristics caused by the time-varying contact stiffness of a deep groove ball bearing and defect sizes. J Tribol 137(3):031–101

The Precise Location Method for Initial Target Based on Fused Information

Yan Junhua(✉), Xu Zhenyu, Zhang Yin, Yang Yong, and Su Kai

College of Astronautics, Nanjing University of Aeronautics and Astronautics,
Nanjing 210016, China
yjh9758@126.com

Abstract. If there is a large error in the initial position of the tracking method, the tracking performance will decrease. This paper proposes an accurate locating algorithm for initial tracking target with fusion of object-like information, saliency information, and priori information to provide accurate initial tracking target for further tracking method. Firstly, the probability distribution of object-like information is calculated by Edge Boxes which is used to detect the object-like regions. Then, the probability distribution of saliency information is calculated by spectral residual method. And the priori probability distribution is calculated according to the target initial position determined by the target detection. Finally, these three probabilities are adaptively fused to build multiple clues fused probability distribution. The precisely located initial tracking target area is the Edge Boxes object-like box with the highest average probability of fusion probability distribution. Experiments show that the accurate location method in this paper can achieve accurate relocation of the target in the case of selection box drift and scale changes and has good real-time performance.

Keywords: Unmanned aerial vehicle · Target tracking · Target relocation · Object-like detection · Pixel saliency

1 Introduction

In UAV target tracking, the initial target position of the tracking module is determined by the target detection module. When the UAV quickly flies approaching the target, the movement state changes violently, and it is difficult to accurately locate the initial tracking target. At this time, there are a large number of errors and background noises in the initial tracking target.

Studies have shown that if background noise and position drift are introduced in the initial tracking target, the performance of most tracking methods will be severely reduced. In [1], the marked target box is scaled, and the range is [0.8, 1.2]. Taking the image in the box area as the initial tracking target, it was found that the performance of the tracking method was significantly reduced. In [2], the initial target tracking frame is shifted, and the offset is 10% of the length and width of the marked target tracking. The performance of the tracking method will also be significantly reduced. Although the tracking method [3] has certain tolerance to the initial tracking target shifting, the tolerance performance is limited, and it can only respond to small error and noise.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 563–575, 2020.
https://doi.org/10.1007/978-981-15-0187-6_67

Therefore, without introducing background information, the initial tracking target of the tracking method is hoped to contain more complete target feature information, such as more complete edge information and color information, so as to construct a more perfect target appearance feature to track the target more accurately.

For the problem, this paper proposes an accurate locating method for initial tracking target with fusion of object-like information, saliency information, and priori information to provide accurate initial tracking target for further tracking methods. The method flowchart is shown in Fig. 1. Firstly, the probability distribution of object-like information is calculated by Edge Boxes [4] which is used to detect the object-like regions. Secondly, the probability distribution of saliency information is calculated by spectral residual method. Thirdly, the priori probability distribution is calculated

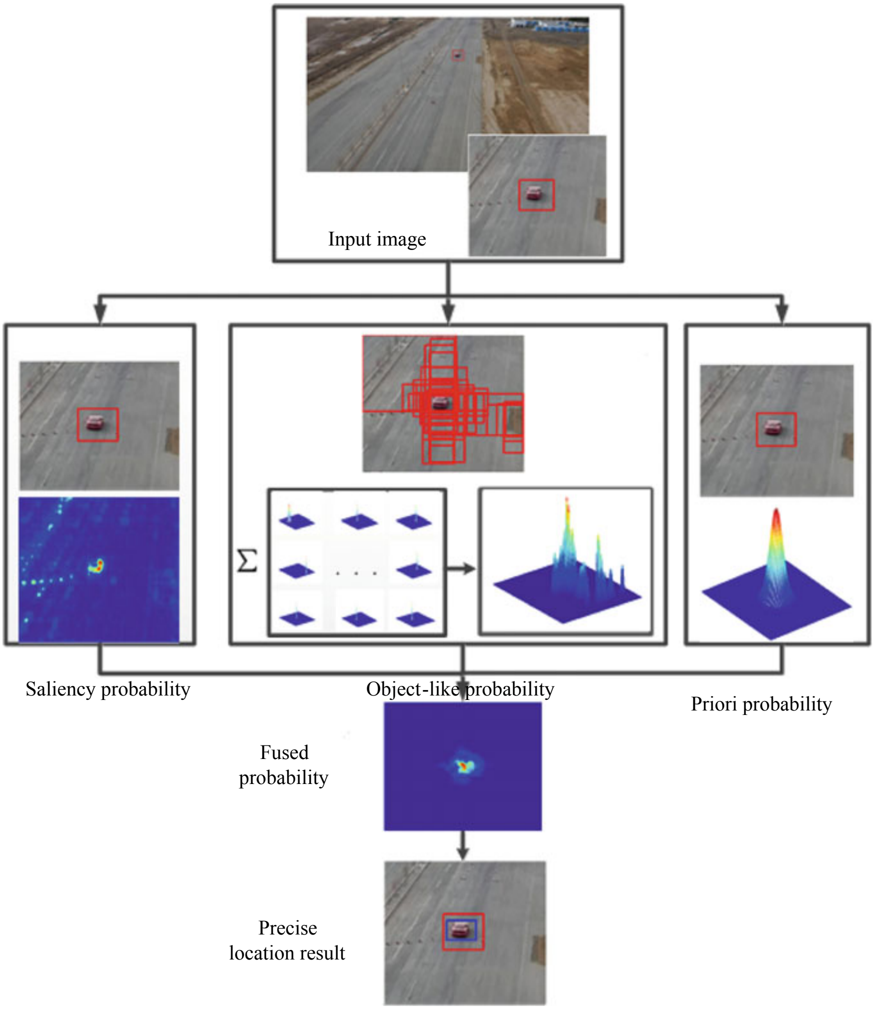


Fig. 1. Flowchart of the precise location method

according to the target initial position determined by the target detection. Finally, these three probabilities are adaptively fused to build multiple clues fused probability distribution. The precisely located initial tracking target area is Edge Boxes object-like box with the highest average probability of fusion probability distribution.

2 Object-like Detection-Based Edge Feature

The Edge Boxes method is designed to get a collection of rectangular regions in the image that are most likely to contain objects. The method first gets edge pixel set E through structured edge detection [5]. Each edge pixel has its corresponding edge intensity m_e and direction θ_e. Eight fields growth is performed on the edge pixels, and the termination condition is that the angle between the edge points beyond $\pi/2$. The set of edge points of growth is called an edge group ES.

Given a set of edge groups $es_i \in ES$, an affinity between each pair of neighboring groups is computed. For a pair of groups es_i and es_j, the affinity is computed based on their mean positions x_i and x_j and mean orientations θ_i and θ_j. Intuitively, the edge groups have high affinity if the angle between the groups' means is similar to the groups' orientations. Specifically, affinity $a\left(es_i, es_j\right)$ is computed using

$$a\left(es_i, es_j\right) = \left|\cos\left(\theta_i - \theta_{ij}\right)\cos\left(\theta_j - \theta_{ij}\right)\right|^{\gamma}. \quad (1)$$

where θ_{ij} is the angle between x_i and x_j.

Given the set of edge groups ES and their affinity a, an object proposal score is computed for any candidate bounding box b. For each edge group, compute a continuous value $w_b(es_i) \in [0, 1]$ that indicates whether es_i is wholly contained in b, $w_b(es_i) = 1$, or not, $w_b(es_i) = 0$. Let ES_b be the set of edge groups that overlap the box b's boundary. For all $es_i \in ES_b$, $w_b(es_i) = 0$ is set to 0. Similarly, $w_b(es_i) = 0$ for all es_i for which $\bar{x} \notin b$, since all of its pixels must be either outside of b or $es_i \in ES_b$. For the remaining edge groups for which $\bar{x} \in b$ and $es_i \notin ES_b$, $w_b(es_i)$ is computed as follows:

$$w_b(es_i) = 1 - \max_{Q} \prod_{j}^{|Q|-1} a\left(q_j, q_{j+1}\right). \quad (2)$$

where Q is an ordered path of edge groups with a length of $|Q|$ that begins with some $q_i \in ES_b$ and ends at $q_{|\Omega|} = es_i$. If no such path exists, we define $w_b(es_i) = 1$. Thus, Eq. (2) finds the path with highest affinity between the edge group es_i and an edge group that overlaps the box's boundary. Since most pairwise affinities are zero, this can be done efficiently.

Using the computed values, we define our score

$$v_b = \frac{\sum_i w_b(es_i)m_i}{2(W_b + H_b)^{\kappa}}. \tag{3}$$

where m_i is the sum of the intensity of all edge pixels in es_i. The W_b and H_b are the box's width and height. A value of $\kappa > 1$ is used to offset the bias of larger windows having more edges on average.

The Edge Boxes can quickly estimate hundreds of object-like boxes b and similar object score v_b. Paper [6] shows that Edge Boxes can better detect various shapes of objects, especially small target objects, in the existing methods for object-like detection. Figure 2 selects the aerial video from [7] and uses the Edge Boxes method to detect 100 object-like boxes. For better visual effect, only the detection box with the overlapped area of the target box remains. The red target in Fig. 2 is the real target position, and the blue color is the test result of the object-like detection. It can be seen from Fig. 2 that the box contains the desired target.

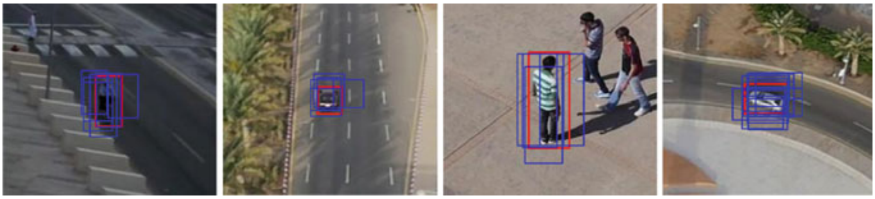


Fig. 2. Object-like detection result based on Edge Boxes

3 Initial Tracking Target Precise Location Method Based on Multiple Information Fusion Probability Distribution

The fusion probability of the object-like information, the saliency information, and the priori information based on the detection module is used to construct the fusion probability. The final precise location result is determined by the fusion probability.

The precise location area *Br* is defined as an Edge Boxes object-like detection box area with the highest area probability score among the fusion probability distributions.

$$B_r = \arg\max_{b_i} p(b_i; c) \quad i = [1, N]. \tag{4}$$

where b_i is represented as object-like box obtained by the Edge Boxes method and c is a target center position determined by the target detection module.

The box probability score is expressed as the average fusion probability of pixels in the area.

$$p(b_i; c) = \frac{1}{H_{b_i} \times W_{b_i}} \sum_{x \in b_i} p(x|c). \tag{5}$$

where x represents the pixel in the area. $p(x \mid c)$ denotes the probability of the fusion probability values of the pixel x belongs to the object-like box b, and the probability value represents the probability of pixel belonging to target, composed of object-like probability $p_o(x|b_i, v_{b_i})$, saliency probability $p_s(x)$, and priori probability $p_m(x|c)$.

$$p(x|c) = p_m(x|c) \times p_s(x) \times \sum_{i=1}^{N} p_o(x|b_i, v_{b_i}). \tag{6}$$

3.1 The Object-Like Probability Distribution Based Edge Feature

The Edge Boxes obtains the object-like box b of the current frame and the object-like score v_b of the object-like box b, where x_b represents the center of object-like box b, and then the probability of pixel x in box b obeys a two-dimensional Gaussian distribution, that is, the farther the pixels are from the center x_b, the less contribution is made to the object-like score of the box b.

$$p_o(x|b, v_b) = \frac{v_b}{2\pi\sigma_o^2} \exp\left(-\frac{(x - x_b)^2}{2\sigma_o^2}\right). \tag{7}$$

where σ_o is proportional to the size of the object-like box b.

3.2 The Saliency Probability Distribution Based on Spectral Residual

If the target position determined by the target detection module drifts too far from the true target position, the peak of the priori probability distribution will be concentrated in the background area. This problem will result in error in the fusion probability. Therefore, it is necessary to reduce the proportion of background information in the fusion probability and highlight the target area, which coincides with the concept of image saliency. Considering the real-time requirements of the scene, the spectral residual [8] theory with good timeliness is used to do saliency detection. The method extracts the image spectral residuals, removes the background information with general characteristics, and finds the parts of the image information which are different. This part of information is defined as saliency information.

The method calculates the Fourier spectrum and the logarithmic spectrum of the image separately. The amplitude of the Fourier spectrum is denoted as $A(f)$, the phase spectrum of the Fourier is denoted as $P(f)$, and the logarithm spectral $L(f)$ is calculated as shown in Eq. (8). The spectrum residual $R(f)$ is calculated as shown in Eq. (9).

$$L(f) = \log(A(f)). \tag{8}$$

$$R(f) = L(f) - h_m(f) * L(f). \tag{9}$$

where h_m represents the mean filtered convolution kernel, which is expressed as

$$h_m = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \tag{10}$$

In order to obtain a better visual effect, the Gaussian kernel $g(x)$ is used to smooth the saliency map, and the obtained image saliency score is shown in Eq. (11), where the inverse Fourier transform is shown.

$$SalS(x) = g(x) * \mathsf{F}^{-1}(\exp(R(f) + P(f)))^2. \tag{11}$$

Figure 3 compares the accuracy of the target before and after the integration of the saliency probability distribution. The first line shows the saliency detection result of the regional image. The response range from black to white is [0, 1]. In the second row, the red box is the simulated detection result with error, blue is the precise location area with no saliency probability distribution, and green is the precise location area that integrates saliency. Table 1 shows the overlap score of the simulated detection area with error, the location area without saliency probability distribution, and the precise location area with saliency probability distribution. The true target represented by the black box, in which the overlap score (OS) is defined as follows:

$$OS = \frac{B_{gt} \cap B_t}{B_{gt} \cup B_t}. \tag{12}$$

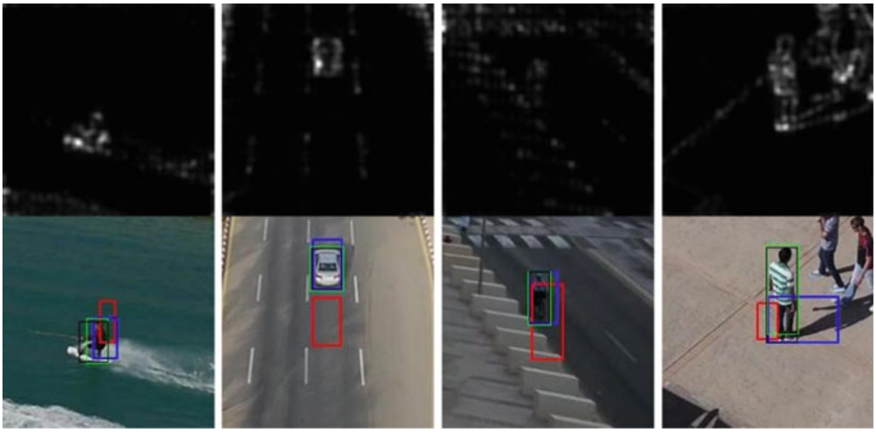


Fig. 3. Comparison of the precise location result with and without image saliency information

Table 1. Comparison of the precise location result overlap score with and without image saliency information

Box type	Detection result with error	Precise location without saliency	Precise location with saliency
OS of first column	0.060	0.307	0.609
OS of second column	0.00	0.686	0.871
OS of third column	0.226	0.676	0.873
OS of fourth column	0.103	0.194	0.811

where B_{gt} represents the target bounding box position in the video frame and B_t represents the position of the exact positioning of the target. Figure 3 and Table 1 show that fused image saliency can make precise location results more accurate.

3.3 The Priori Probability Distribution Based on Detection Result

The object-like boxes include both the true target and some interference targets. This method uses the target position determined by the target detection as priori information to punish the interference target. Although there is a deviation in the target position determined by the target detection, the center position of the target determined by the target detection is generally close to the target. The closer the object-like box is to the target position determined by the target detection, the greater the probability that it is the true initial tracking target. Therefore, it is considered that the priori probability of each pixel x with the target initial position determined by the detection obeys a two-dimensional Gaussian distribution.

$$p_m(x|c) = \frac{1}{2\pi\sigma_m^2}\exp\left(-\frac{(x-c)^2}{2\sigma_m^2}\right). \tag{13}$$

where σ_m is expressed as:

$$\sigma_m = \frac{\lambda_m z}{\sqrt{W_b \times H_b}}. \tag{14}$$

z is the empirical constant value, taking 1/16, c is the initial target center determined by the target detection.

The weight λ_m is used to determine the degree of influence of the target initial position determined by the detection module on the fusion probability. If there are more regions similar to the target in the current image, the object-like scores and saliency scores of these targets will be relatively large, resulting in the fusion probability that cannot distinguish true target from the interference target well. In this case, the priori information provided by the initial position of the target determined by the target detection needs to be used. By decreasing the value of a to make b smaller, the probability peak region of the two-dimensional Gaussian distribution of c is more concentrated on the target position determined by the target detection.

Define λ_m as a function of the average saliency score $SalS_{mean}$ and variance σ_b of all the object-like boxes' score v_b, as shown in Eq. (15). When there are more similar objects, the overall object-like score will be higher and tend to be consistent, and the variance will be smaller. At the same time, if the saliency of the image is stronger and the average saliency score will be higher, β takes the experimental optimal value 2.

$$\lambda_m = \beta * \sqrt{\frac{\sigma_b}{SalS_{mean}}}. \tag{15}$$

$$\sigma_b = \frac{1}{N}\sum_{i=1}^{N}\left(v_{b_i} - \bar{v}_b\right)^2. \tag{16}$$

Figure 4 shows the comparison of the fused probability with and without the priori information probability distribution. It shows that the priori information can make the target more prominent and reduce the influence of the interference target.

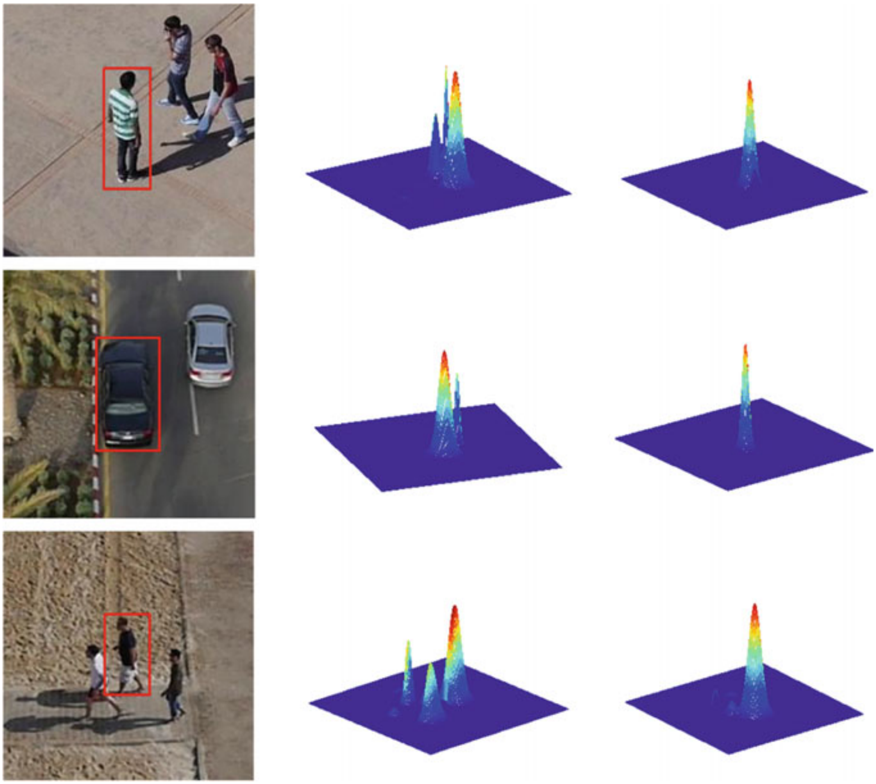
4 Diagram of Precise Location Method for Initial Target Based on Fused Information

Diagram of precise location method for initial target based on fused information is shown as Fig. 5. First, the local search area A is selected based on the detection position and the pseudo-physical property detection results based on the Edge Boxes are performed on the local areas to construct a pixel-like probability distribution. Then the significance of the original image was detected to obtain a pixel probability distribution. Finally, a priori probability distribution is constructed based on the target position obtained by the detection. The fusion of the three constructs multiple clues fused probability distribution. The object-like box with the highest average fused probability is used as the final precise location result.

5 Experimental Results and Analysis

UAV123 UAV aerial video set [7] was selected to carry on target precise positioning experiment. In the case that the tracking area is too large and the tracking position drifts, verify the effectiveness and robustness of the precise location method.

The degree of drift of detection result is measured by using the center location error (CLE), that is, the Euclidean distance between the center position of the detection result B_s and the center of the accurately labeled target B_{gt}. The ratio of the lengths of diagonals B_s and B_{gt} is used to measure the scale of the detection result.



(a) Local image with disturbed objects

(b) Fused probability without priori probability

(c) Fused probability with priori probability

Fig. 4. Comparison of the precise location results with and without the priori information probability distribution

The overlap score is used as the evaluation index of the accuracy of the precise location. The overlap score is expressed as follows, and B_r shows the precise location result.

$$OS = \frac{B_{gt} \cap B_r}{B_{gt} \cup B_r}. \tag{17}$$

Experimental parameters: The number of the object-like box is set to 100, and the local area D is set to 100 × 100. Experimental platform: 3.60 GHz, Intel i7 CPU, 64-bit win7 operating system, 8 GB of memory. Under this experimental condition, the frames per second (fps) of the precise location method can reach 86 frames per second, which meets the requirement of real-time performance.

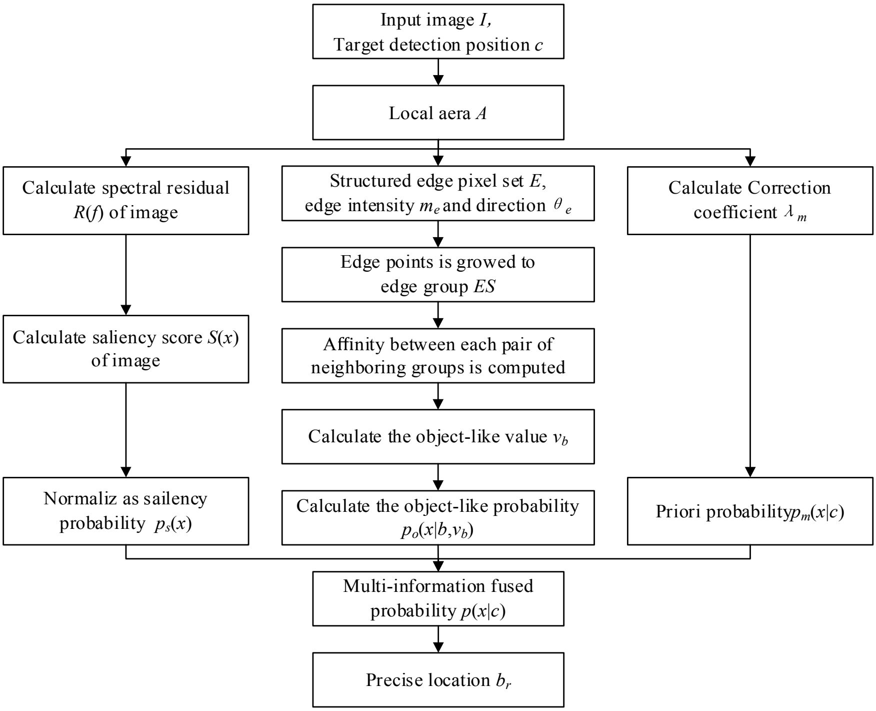

Input image I,
Target detection position c
Local aera A
Calculate spectral residual $R(f)$ of image
Structured edge pixel set E, edge intensity m_e and direction θ_e
Calculate Correction coefficient λ_m
Edge points is growed to edge group ES
Calculate saliency score $S(x)$ of image
Affinity between each pair of neighboring groups is computed
Calculate the object-like value v_b
Normaliz as sailency probability $p_s(x)$
Calculate the object-like probability $p_o(x|b,v_b)$
Priori probability$p_m(x|c)$
Multi-information fused probability $p(x|c)$
Precise location b_r

Fig. 5. Diagram of precise location method for initial target based on fused information

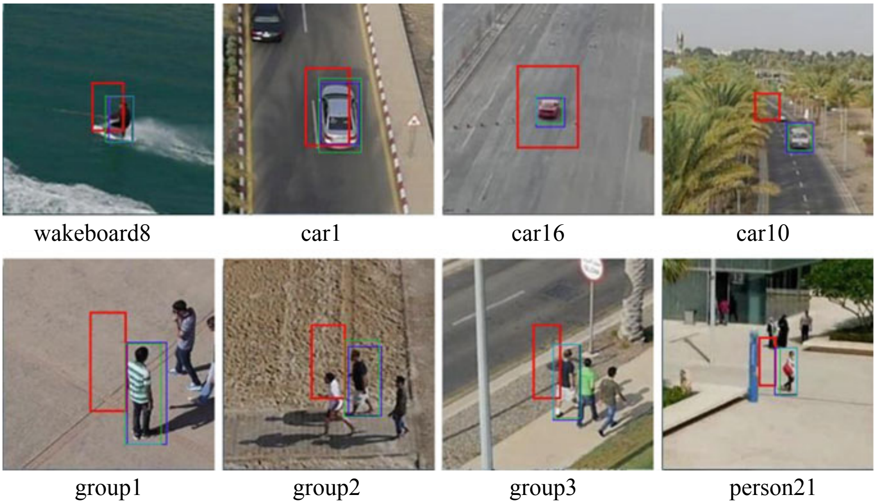
5.1 Validity Experiment of Precise Location Method

Validity experiment is mainly to observe whether the method can location the target accurately when the target detection result is wrong. This paper considers the method location target effectively when overlap score between B_r and B_{gt} exceeds 0.5. In the experiment, 10 frames of image are selected in each video sequence. B_{gt} of each frame of image is subjected to 1000 random translation and scale-up to simulate the detection result with error. The center error range is controlled within [10, 50]. The scale error ratio is within [1, 3] (Table 2).

Table 2. Ration of effective precise location in 1000 experiment results

Video sequence	wakeboard8	car1	car16	car10
Effective ratio	0.846	0.883	0.915	0.854
Video sequence	group1	group3	group3	person21
Effective ratio	0.772	0.693	0.748	0.762

Table 2 and Fig. 6 show the effective ratios and relevant precise location results for 1000 experiments in part of the video sequence in the validity experiment. The red box in Fig. 6 is the simulated detection with error B_s, green is the accurately labeled B_{gt}, and blue is the precise location result B_r.



wakeboard8 car1 car16 car10

group1 group2 group3 person21

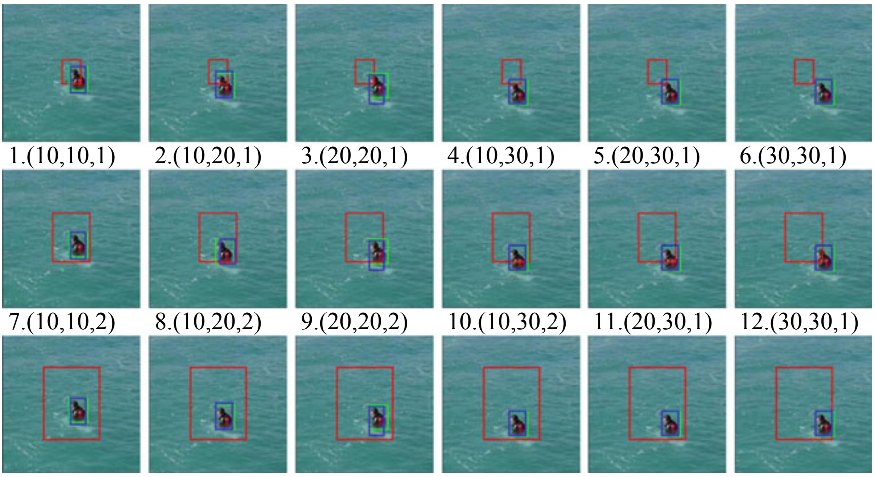
Fig. 6. Example of validity experiment results

The data in the first row of Table 2 and the first row of images in Fig. 6 indicate that in the absence of similar disturbed targets in the surroundings, the precise location method can effectively locate the target with the wrong detection result. The first row in Table 2 and the second row in Fig. 6 indicate that when there are many objects with the same characteristics around the target, the performance of the method will decrease by about 10%. This is because the Edge Box is based on the edge characteristics. These disturbed objects have similar edge characteristics with the target. Therefore, the object-like box with a strong likelihood of a probabilistic object is also detected around the disturbed target. In addition, these disturbed objects have the same saliency relative to the background with target. At this point, the priori information provided by the detection result is used to distinguish. When the detection result drifts to the disturbed object, the location result will drift to other objects.

5.2 Robustness Experiment Precise Location Method

The experimental goal is to explore the effect of the target detection result's drifting pixel size and scale magnification on the accuracy of the overlap score between the precise location result B_r and the accurately labeled target B_{gt}.

Robustness experiment selects single target scenes and interference target scenes respectively for experiments. In the experiment, the labeled target frame B_{gt} is translated along the horizontal and vertical directions and the scaled up to simulate the detection result B_s with errors, such as shown in Fig. 7, the data in parentheses below the picture respectively represent the number of pixels along the horizontal and vertical directions of the target frame and the scale-up factor. The red box is the simulated detection result B_s, green box is the accurate labeled target B_{gt}, and blue box is the precise location result B_r.



1.(10,10,1) 2.(10,20,1) 3.(20,20,1) 4.(10,30,1) 5.(20,30,1) 6.(30,30,1)

7.(10,10,2) 8.(10,20,2) 9.(20,20,2) 10.(10,30,2) 11.(20,30,1) 12.(30,30,1)

13.(10,10,3) 14.(10,20,3) 15.(20,20,3) 16.(10,30,3) 17.(20,30,3) 18.(30,30,3)

Fig. 7. Example of robustness experiment

Figure 7 shows that the proposed method can accurately locate the target under various degrees of displacement and scale error conditions, indicating that the method has good robustness.

6 Conclusion

This paper proposes a precise location method for initial tracking target based on fused information to provide a more accurate initial tracking target for tracking. This method fuses priori information, significant information, and object-like information to generate a fused probability distribution and obtains the most likely target area to achieve accurate positioning of the target. Experimental results show that the proposed method can accurately locate the target in real time under the condition that the initial tracking target has large drift and scale changes. When there are many disturbed objects around the target, there is no priori knowledge and the difference between similar object which is hard to be distinguished. As a result, the accuracy and robustness of the method all decrease.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (61471194, 61705104), Science and Technology on Avionics Integration Laboratory and

Aeronautical Science Foundation of China (20155552050), the Natural Science Foundation of Jiangsu Province (BK20170804), and National Defense Science and Technology Special Innovation Zone Project.

References

1. Wu Y, Lim J, Yang MH (2013) Online object tracking: a benchmark. In: Computer vision and pattern recognition, pp 2411–2418. IEEE, New York
2. Kristan M, Pflugfelder R, Leonardis A et al (2014) The visual object tracking VOT2014 challenge results. In: IEEE international conference on computer vision workshops, pp 98–111. IEEE, New York
3. Yi KM, Jeong H, Heo B et al (2014) Initialization-insensitive visual tracking through voting with salient local features. In: IEEE international conference on computer vision, pp 2912–2919. IEEE, New York
4. Zitnick CL, Dollár P (2014) Edge boxes: locating object proposals from edges. In: Computer vision—ECCV 2014, pp 391–405. Springer International Publishing
5. Zitnick CL (2013) Structured forests for fast edge detection. In: IEEE international conference on computer vision, 99, 1841–1848. IEEE Computer Society
6. Zhu H, Porikli F (2016) Automatic refinement strategies for manual initialization of object trackers. IEEE Trans Image Process 99:1
7. Mueller M, Smith N, Ghanem B (2016) A benchmark and simulator for UAV tracking. In: European conference on computer vision, pp 445–461 Springer International Publishing
8. Hou X, Zhang L (2007) Saliency detection: a spectral residual approach. In: IEEE conference on Computer vision and pattern recognition, 2007. CVPR'07, pp 1–8. IEEE, New York

CMnet: A Compact Model for License Plate Detection

Yao Cheng, Xin Zhou(✉), and Bo Ning

Dalian Maritime University, 1 Linghai Road, Dalian 116026, China
zhouxin314159@dlmu.edu.cn

Abstract. In recent years, with the rapid development of artificial intelligence (AI), intelligent license plate (LP) detection which marks vehicle LP position from the images has become one of the core technologies of intelligent transportation system. The traditional license plate detection algorithm has low accuracy in some scenes, such as rotation, snow or fog, weak light, distortion, and blur. Therefore, this paper proposes a model based on convolutional neural network to realize vehicle LP detection. The model reduces the number of convolution layers in the case of ensuring accuracy, thereby reducing the number of parameters and saving time overhead. Through experiments, our model achieved an accuracy of 98.8%, and the training time and detection time were much less than the compare models.

Keywords: Object detection · Convolutional neural network · License plate detection

1 Introduction

License plate detection [1–5] is an essential part of intelligent transportation systems and is widely used in many traffic monitoring systems. The challenges of LP detection have still existed in many current scenes. There are more or less problems in many actually taken LP images, which makes it difficult to extract license plate features. Such as Fig. 1a the license plate and the body of car are covered with snow in snowy days. Figure 1b the shooting angle is distorted. Figure 1c blurring caused by distance. Figure 1d the lack of clarity caused by low light. Figure 1e uneven illumination caused by strong light reflection. Many LP detections have lower accuracy in such complex scenes, which need us to further explore the license plate detection under complex scenes.

The license plate detection algorithm can be roughly divided into traditional methods and neural network models. Traditional LP detection methods always use rich edge information [6–12] or background color features [13, 14]. In [10], the authors analyzed vertical edge gradients to select real license plate position. According to [14], the author also used a color check module to help find the license plate position. The breakthrough development of convolutional neural networks based on recent years has stimulated the widespread application of object detection models in LP detection problems [3, 4, 15]. Through computer efficient data processing and rich LP image data, CNN improves the performance of LP detection by training the correlation

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 576–583, 2020.
https://doi.org/10.1007/978-981-15-0187-6_68

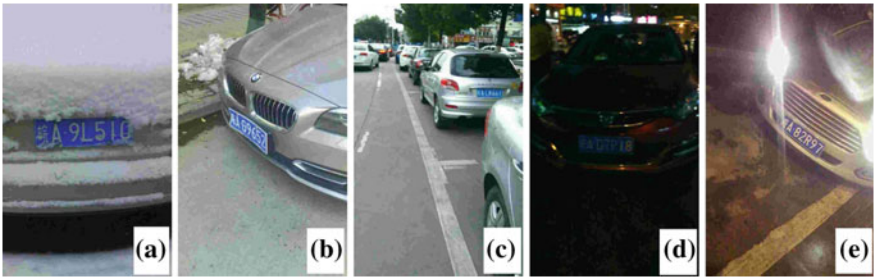


Fig. 1. A sample image that is difficult to detect

features between image data. RPnet [15] uses convolutional neural networks to generate high-quality regions for detection to detect objects more accurately and quickly. However, in numerous challenging and complex scenes, the detection model tends to have a sharp drop in accuracy. Moreover, the complicated of the model and the large number of parameters lead to the long training time and the slow detection speed. We designed a new detection model to use the convolutional neural network to complete the license plate position detection. Based on the simpler model architecture, the detection efficiency is improved under the premise of ensuring the detection accuracy. This paper makes the following contributions:

- We proposed a new detection model named CMnet. The model is consisting of five convolutional layers, and we can accurately detect the license plate at a faster speed.
- During the training, we connect the deconvolution network behind each convolutional layer to visualize the extracted image features. The structure of the network model is adjusted and optimized in this way in order to achieve higher accuracy with a compact model.
- Through experiments, we have proved that our model is superior to other methods in terms of accuracy and speed.

2 A Compact Model for LP Detection

The detection network is shown in Fig. 2. The entire model is a CNN with five convolutional layers [part (a)]. Each convolution layer consists of five parts: Convolution, Batch normalization, Activation, Maxpool, and Dropout [part (b)].

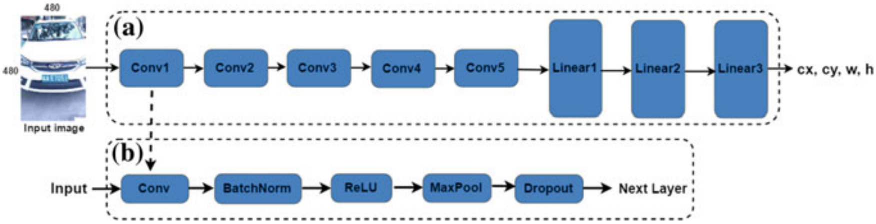


Fig. 2. The structural design of the model CMnet

The convolution layer extracts different levels of features from the input image to obtain feature maps. The model passes the output of the last convolutional layer to the three fully connected layers, ultimately resulting in the position of the predicted bounding box cx, cy, w, h. Where *cx, cy* is the center point *x* coordinate of the bounding box, the center point *y* coordinate, *w, h* are the width and height, respectively.

The detection model input an image and predicts the bounding box of the LP position in the image. The localization loss (see Eq. 2.1) is a Smooth L1 loss between the predicted box (pb) and the ground truth box (gb). Where *N* is the batch size in training, pb^m is one of the four coordinates *cx, cy, w, h* of the predicted bounding box, gb^m is one of the four coordinates *cx, cy, w, h* of the ground truth bounding box. The parameters were updated using a stochastic gradient descent (SGD) method.

$$\text{Loss}(\text{pb}, \text{gb}) = \sum_{N} \sum_{m \in \{cx,cy,w,h\}} \text{smooth}_{L1}(\text{pb}^m - \text{gb}^m) \tag{2.1}$$

3 Visualization with a DeconvNet

The model CMnet uses convolution operations to extract features from the input image. As the number of network layers deepens, the characteristic noise extracted from the low-to-high convolution layer is gradually reduced. It is more conducive to detecting the license plate and predicting its bounding box. We use the deconvolution operation to visualize the feature maps. For example, as shown in Fig. 3.

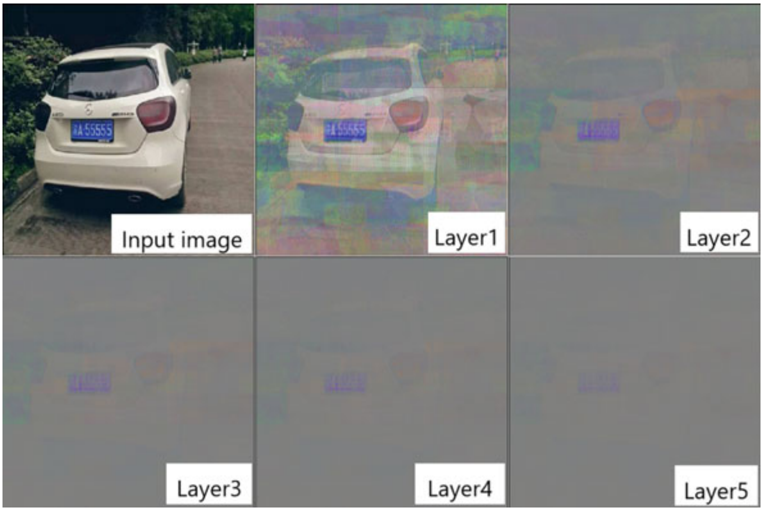


Fig. 3. Deconvolution results. Layer 1-5 is a feature image obtained by deconvolution reconstruction of the feature map obtained by Conv1-5. The final image is color enhanced

In this paper, by using a multi-layer deconvolution network (DeconvNet) [16], the feature image extracted by the convolution layer is mapped back to the pixel space, so that the visualization function enables us to observe the evolution of the feature map and diagnose potential problems during training. DeconvNet can be thought of as an inverse sequence model using the same components (shown in Fig. 4), which does not have any learning ability. First input the image input to the convolution layer to get the feature maps. For this layer's feature map, pass it as input to the DeconvNet deconvolution of this layer to get the feature image.

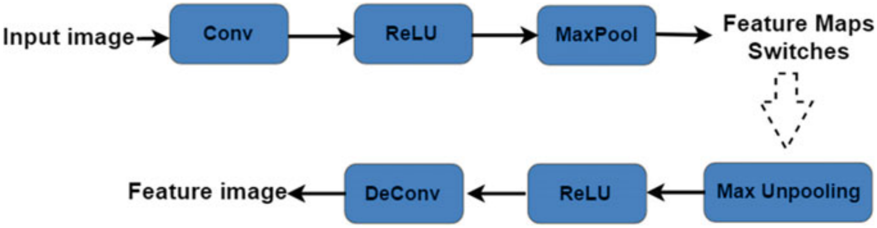


Fig. 4. DeconvNet structure. Switches are the locations corresponding to the maximum value of the merged region obtained by the convolutional layer Maxpool operation

4 Experiments

4.1 Training

Our model uses CCPD dataset [15], which contains 250 k sample images (720 width * 1160 heights * 3 channels). For CCPD-base the only common feature of these images is the inclusion of a license plate. The CCPD-weather is images taken on a rainy day, snow day, or fog day. CCPD-tilt and CCPD-rotate are images with high image horizontal tilt. CCPD-db is the illumination of the LP area is dark, uneven, or very bright. CCPD-fn is a relatively distant or close distance from the LP to the shooting location. The CCPD-challenge is by far the most challenging image. We split the image of the 200 k CCPD-base dataset into two, one as a training dataset and one as a test dataset. The model trained 45 epochs on the CCPD training dataset. We send each image to resize to a size of 480 * 480. The learning rate set to 10-2 and a combined momentum of 0.9. The Dropout rate is 0.2.

4.2 Detection

We follow the standard protocol in object detection Intersection-over-Union (IoU) [17]. And set the IoU value to 0.7. We use accuracy to measure the effect of the model. True detection (TD) indicates the number of samples for effective prediction, and false detection (FD) indicates the number of samples for invalid prediction.

$$\text{Accuracy} = \frac{\text{TD}}{\text{TD} + \text{FD}} \tag{4.1}$$

The results of the tests are shown in Table 1. We tested the accuracy on seven test datasets from CCPD. In addition to the CCPD-Base test dataset, samples of other datasets are challenging samples of some scenarios.

Table 1. The accuracy of the model on the CCPD dataset

Accuracy (%)	CCPD-base	CCPD-weather	CCPD-tilt	CCPD-rotate	CCPD-fn	CCPD-db	CCPD-challenge
VGG16	95.3	94.1	93.8	92.0	91.9	93.5	93.3
RPnet	96.8	96.1	95.0	92.8	93.6	95.7	94.6
CMnet	98.8	98.6	98.5	97.9	96.5	97.3	96.8

4.2.1 Accuracy Evaluation of Comparison Model

We apply the VGG16 [18] neural network model to the license plate detection, the test results as shown in Table 1. For RPnet [15], we experimented with the model of the LP detection part. On the CCPD-Base dataset, our model achieved the highest accuracy of 98.8%. The accuracy on the CCPD-rotate dataset is 97.9%, far exceeding the other two models. Moreover, for several other challenging test datasets, our model accuracy has not dropped significantly.

4.2.2 Speed Evaluation of Comparison Model

We also compared the detection time and training time of the model. The results are shown in Table 2. There are 100 K sample images in the test dataset and training dataset. Our model detection time is 0.6 h, and the average detection time is 0.02 s. The training time is 1.1 h for one epoch. Whether it is the test time or the training time, our model is much faster than other models.

Table 2. The detection efficiency of the model

Time (h)	Train time	Test time
VGG16	18.2	2.1
RPnet	8.0	2.2
CMnet	1.1	0.6

4.2.3 Accuracy Evaluation of CMnet After Model Optimization

In the experiment, we made several adjustments to the model. By visualizing feature extraction using deconvolution techniques, we found that the optimized model is better able to extract LP features (Fig. 5). Also, the accuracy of the optimized model has been improved. The results are shown in Table 3. We reduced the number of layers in the

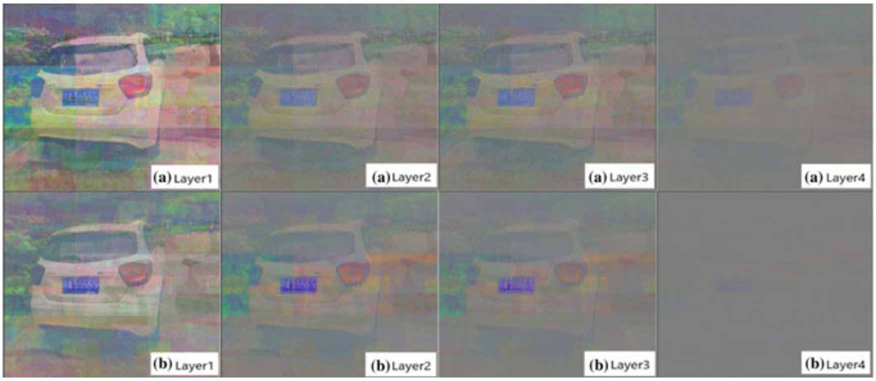


Fig. 5. Feature extraction visualization. **a** The model before optimization. **b** The optimized model

Table 3. Accuracy after model optimization

Model	Accuracy (%)
CMnet-7layer-5size	96.9
CMnet-5layer-3size	97.1
CMnet-7layer-5size	96.7
CMnet-5layer-7size	98.8

network model and increased the size of the convolution kernels of the first two layers, which significantly improved the accuracy of model detection. Samples of detection results produced by evaluating CMnet on CCPD are shown in Fig. 6.

5 Conclusions

Overall, in this paper we propose a new neural network model for single LP detection. Our proposed model is superior to the state-of-the-art method in terms of accuracy and efficiency. In the future work, we will implement the bounding box calibration function for license plate detection and implement license plate number recognition.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (No.61976032).

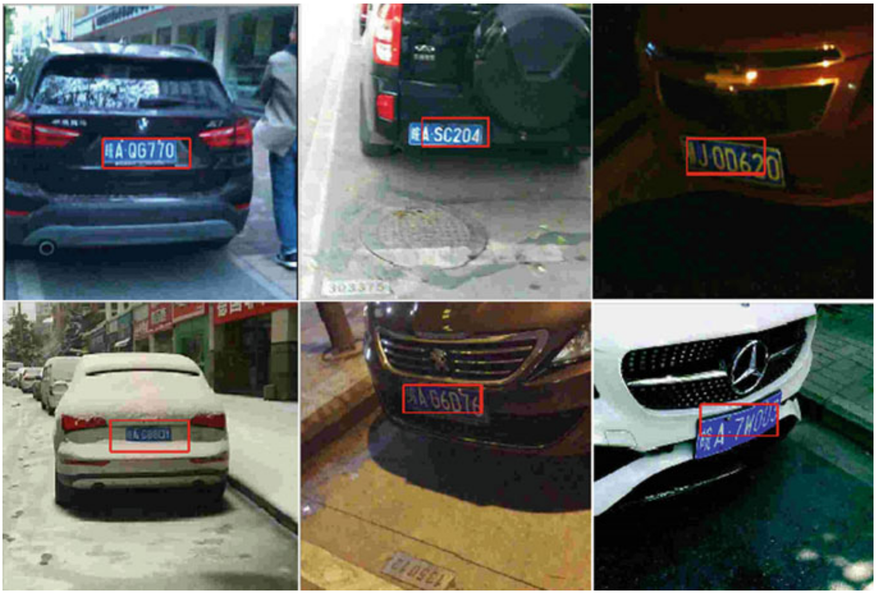


Fig. 6. Detection results on CCPD with our CMnet model

References

1. Xie L, Ahmad T, Jin L, Liu Y, Zhang S (2018) A new cnn-based method for multi-directional car license plate detection. IEEE Trans Intell Transp Syst 19(2):507–517
2. Al-Shemarry MS, Li Y, Abdulla S (2018) Ensemble of adaboost cascades of 3 l-lbps classifiers for license plates detection with low quality images. Expert Syst Appl 92:216–235
3. Laroca R, Severo E, Zanlorensi LA, Oliveira LS, Gonc¸alves GR, Schwartz WR, Menotti D (2018) A robust real-time automatic license plate recognition based on the yolo detector. arXiv preprint arXiv:1802.09567
4. Montazzolli S, Jung C (2017) Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In: 2017 30th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), pp 55–62
5. Selmi Z, Halima MB, Alimi AM (2017) Deep learning system for automatic license plate detection and recognition. In: 2017 14th IAPR international conference on document analysis and recognition (ICDAR), vol 1. IEEE, pp 1132–1138
6. Haralick RM, Sternberg SR, Zhuang X (1987) Image analysis using mathematical morphology. IEEE Trans Pattern Anal Mach Intell PAMI 9(4):532–550
7. Hsieh JW, Yu SH, Chen YS (2002) Morphology-based license plate detection from complex scenes. In: 2002 Proceedings of 16th international conference on pattern recognition, vol 3. IEEE, pp 176–179
8. Wu HHP, Chen HH, Wu RJ, Shen DF (2006) License plate extraction in low resolution video. In: 18th International conference on pattern recognition, 2006. ICPR 2006, vol 1. IEEE, pp 824–827
9. Yu S, Li B, Zhang Q, Liu C, Meng MQH (2015) A novel license plate location method based on wavelet transform and emd analysis. Pattern Recogn 48(1):114–125

10. Saha S, Basu S, Nasipuri M, Basu DK (2009) License plate localization from vehicle images: an edge based multi-stage approach. Int J Recent Trends Eng 1(1):284–288
11. Wang R, Sang N, Huang R, Wang Y (2014) License plate detection using gradient information and cascade detectors. Optik-Int J Light Electron Opt 125(1):186–190
12. Bachchan AK, Gorai A, Gupta P (2017) Automatic license plate recognition using local binary pattern and histogram matching. In: International conference on intelligent computing, Springer, pp 22–34
13. Deb K, Jo KH (2008) Hsi color based vehicle license plate detection. In: International conference on control, automation and systems, 2008. ICCAS 2008. IEEE, pp 687–691
14. Yao Z, Yi W (2014) License plate detection based on multistage information fusion. Inf Fusion 18:78–85
15. Xu Z, Yang W, Meng A, Lu N, Huang H (2018) Towards end-to-end license plate detection and recognition: a large dataset and baseline. In: Proceedings of the European conference on computer vision (ECCV), pp 255–271
16. Zeiler MD, Fergus R (2013) Visualizing and understanding convolutional networks. CoRR, abs/1311.2901
17. Li H, Wang P, Shen C (2017) Towards end-to-end car license plates detection and recognition with deep neural networks. arXiv preprint arXiv:1709.08828
18. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

Parallelization of Preprocessing for Automatic Detection of Pulmonary Nodules in 3D CT Images

Junpeng Weng, Yu Tang(✉), and Di Lin

University of Electronic Science and Technology of China, No. 4, North Jianshe Road, Chengdu, P.R. China
wengjunpeng@qq.com, {yutang,lindi}@uestc.edu.cn

Abstract. Computer-aided diagnosis (CAD) systems have been introduced for therapeutic institutions to help radiologists to diagnose lung cancer at the early stage. This paper works on the parallelization of preprocessing steps in the CAD systems. We first propose a parallelization scheme based on static data grouping. The 3D lung CT images are grouped in the form of 2D image slices, and the grouped data is parallelized. Then, a pipeline-based parallelization scheme is proposed. Mutually independent steps are distributed to the respective execution units, and the preprocessing operations are performed in a parallel pipeline manner. Finally, a bus-based parallelization scheme is proposed. Based on the pipeline, a control unit is added to dynamically assign computing tasks to the various execution units. We test our schemes in an existing CAD system with data from Lung Nodule Analysis 2016 (LUNA16) dataset, and analyze the performance of data testing.

Keywords: Lung nodule detection · Parallelization · Image processing

1 Introduction

Lung cancer is one of the most deadly cancers in the world. According to the Cancer Research Report released by the American Cancer Research Center in 2018, there are more than 2 million new cases of lung cancer in the world in 2018, accounting for about 11.6% of all new cancer cases; more than 1.7 million deaths, accounting for the total about 18.4% of cancer cases, which far exceed other cancer types [1]. Clinical studies show that the five-year survival rate of early lung cancer patients can reach 70%, but the five-year survival rate of patients with advanced lung cancer significantly drops to less than 15% [2]. These studies indicate that the survival rate of lung cancer is closely related to whether they can receive early treatment.

Pulmonary nodule is one of the common lesions in the lungs, and is often seen as early manifestations of lung cancer. The early detection of pulmonary nodules plays a crucial role in the early treatment of lung cancer, but it is hard to identify potential lung cancer patients who have developed pulmonary nodules under the condition of limited medical conditions. To solve the problem, computer-aided detection systems are introduced into the therapeutic facility. By collaborating with the CAD systems, it is

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 584–590, 2020.
https://doi.org/10.1007/978-981-15-0187-6_69

possible to make up for the gap in the doctors' diagnostic level and improve the detection of early symptoms of pulmonary nodules in a wider area.

CAD system is generally composed of five main stages: acquisition of data, preprocessing, lung segmentation, nodule detection and the reduction of false positives [2]. The first stage is responsible for obtaining the images as the input of CAD system. The preprocessing stage includes a series of image processing to improve the quality of the image or to compress the amount of input data to improve the result of later stage. The lung segmentation stage separates the learning area from other organs and tissues in the ray image to reduce the computational cost of the next stage. The nodule detection stage determines the location of the lung nodules in the segmented image as candidates. The reduction of false positives stage use machine-learning algorithms to eliminate false nodules from the candidate nodules.

In this work, our purpose is to parallelize the stages before reduction of false positives, which can accelerate the calculation speed without affecting the output results. To achieve this purpose, we developed three concepts to deal with the existing system [3].

1. Data concept. Because the 3D lung CT image is composed of number of 2D image slices, we treat each 2D image slice as one processing unit. We group the 2D image slices and process these groups at the same time.
2. Instruction concept. Preprocessing, lung segmentation and nodule detection are mutually independent, so we spread these stages across different execution units and perform parallel processing on multiple data in a pipelined manner.
3. Data transmission concept. We use the bus structure to optimize data transmission and simplify the system structure.

2 Parallelization Scheme Design

When designing the parallelization scheme, we considered the differences in various scenarios. For medical institutions with a complete data center and powerful computing capabilities, the CAD systems can be deployed in a local environment. In this case, we designed a parallelization scheme for the local environment. However, for medical institutions without the capabilities, when they need to conduct remote consultation, they have to transmit data to the remote computer cluster. In this case, we consider about how to transmit data efficiently and reliably in computer cluster. We designed three schemes to meet them: static grouping-based scheme, pipeline-based scheme and bus-based parallelization scheme.

2.1 Static Grouping-Based Parallelization Scheme

The static grouping-based scheme refers to the fixed division of slices into the same number of group as the execution unit, and each execution unit only needs to care about the grouped slices assigned to it. There is no data transmission between the execution units, they only need to receive input and send output to control unit. For the control unit, its main job is to group the 2D image slices to be processed. In order to ensure that the order of groups is consistent with the order of execution units, the control unit needs to keep communication with the execution unit, and divide the 3D image into 2D image slice groups that are as uniform as possible, and then send the grouped slices to each execution unit. In the static grouping-based scheme, each execution unit has a complete instruction stream. Each execution unit can be regarded as a separate subsystem, and its workflow is not affected by the outside. The control unit maintains a correspondence between the grouped slices and the execution unit. When the processed slices are received, the processed slices can be rearranged in the order of the original slices by the correspondence. After all the slices have been processed, the 2D image slices can be restored to a 3D image, which is ready to enter the reduction of false positives stage.

2.2 Pipeline-Based Parallelization Scheme

The key problem of the static grouping-based scheme is that some information may not be evenly distributed over the entire picture, such as edge recognition or region of interest, which may result in different load between different execution units during processing. To solve the problem, we introduced pipeline-based scheme. In the pipeline-based scheme, the control unit needs to divide the input as small as possible. In this paper, we take each 2D slice as input to the execution unit. For each execution unit, it does not execute all the stages on each input, but only execute one specified stage. The intermediate result will be transmitted to the next execution unit. In computer cluster, the order of execution units is critical, and it is the main task of the control unit in a pipeline-based scheme. During the startup phase, the control unit assigns the execution unit its stage and its delivery target of intermediate result. During execution, the control unit is only in contact with the execution unit of the first stage and the one of the last stage, for the input and output of pipelines. The output of the pipeline-based parallelization scheme obviously remains in the same order as the input, so the control unit does not need to reorder the processed slices.

2.3 Bus-Based Parallelization Scheme

Pipeline-based scheme lacks flexibility because the stage for execution unit to execute is specified before image enters the system. For better flexibility, we designed a bus-based parallelization scheme, which transmits data with information about stage in a bus. The control unit needs to maintain two queues to store the intermediate data and the processed data. Whenever data is transmitted back to the control unit, the control unit determines which queue to put it into according to the stage of the data, and sends the intermediate data to the idle execution unit. In bus-based scheme, execution units receive slices and the stage of the slices, then performs the corresponding processing. The control unit in computer cluster plays the role of bus, it sends the slices with stage information to execution units and receives processed data.

The bus-based scheme requires that each input be as small as possible. It can make the execution unit spend as little time as possible in executing an input. If the input is too large and takes too long time to be executed, it will cause the overall execution time to become uncontrollable, and affects load balancing. However, in order to achieve load balancing, it also costs a price. The most important price is the time cost of communication. The number of groups that need to be executed is big, and each execution unit needs to report the result to the control unit after the execution is completed, so a lot of communication costs are generated during the period. This communication cost tends to be particularly expensive when communicating through the network. Especially, when the amount of calculation of instructions is relative small to the number of execution units, the overhead caused by such load balancing may even exceed the benefits of parallelization.

3 Experiments

To test the gain of our scheme, we ran a CAD system in our scheme using data from the LUNA16 dataset, and then compared the execution time between in single thread and in our schemes.

In preprocess stage, we use OTSU method to calculate the threshold of whole image [4], and then use the threshold to binary the image. Then, we use morphological operations to segment the lungs [5]. We labeled various connected components and cleared image border, then treated the two components with largest areas as lung parenchyma. There were some other issues like blood vessels remained, so we applied erosion operation and closure operation to remove the issue and filled the hole. With the lung segmentation, we use region growing to generate candidate lung nodules [6] (Fig. 1).

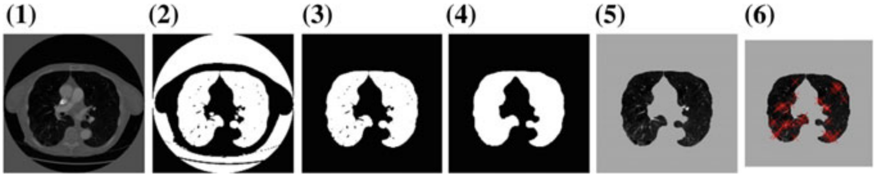


Fig. 1. (1) Origin image; (2) image after threshold; (3) image after clear border; (4) lung mask; (5) lung segmentation; (6) candidate lung nodules

We use multi-thread to test our scheme in local environment, and use docker to simulate computer cluster. Performance of our scheme in local environment and in simulate computer cluster is show in Fig. 2. In local environment, we controlled the number of execution units from 1 to 7, but the pipeline-based parallelization scheme has only 1–4 execution units of data due to its poor scalability. In simulate computer cluster, we controlled the network transmission speed from 1 to 10 MB/s.

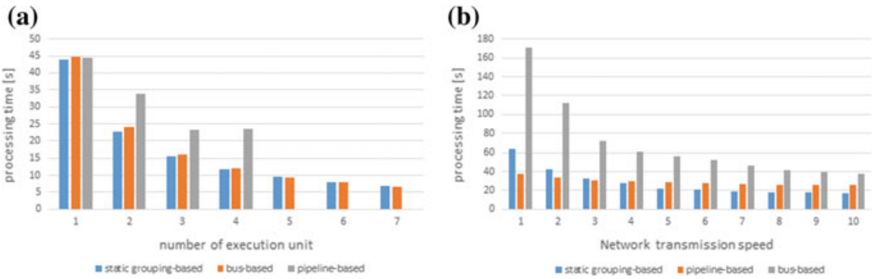


Fig. 2. Data processing time in **a** local environment and **b** simulate computer cluster

The static grouping-based scheme had the best performance, which is slightly inferior to the bus-based parallelization architecture in more than five execution units; the pipeline-based scheme is not only poor scalability but also poor performance in execution, which took about 1.5 times time than the other two schemes; the bus-based parallelization scheme is slightly slower than the static group-based scheme, but as the number of execution units increases, the execution efficiency can be maintained very stably. Therefore, the bus-based parallelization architecture is more adaptive than the static grouping-based scheme for systems with a large number of processing units.

In simulate computer cluster, static grouping-based scheme run fastest; the pipeline-based scheme has bottlenecks in the processing steps, so the overall speed in the network environment is still slow, but the impact of network transmission is gradually covered by the delay in the processing step as the network speed increases, so it can gradually approach the execution speed in the local environment; the bus-based scheme was most affected in simulate computer cluster, and its execution speed has even dropped to a point slower than the pipeline-based scheme. In summary, the three parallelization schemes are analyzed separately:

(1) Static grouping-based scheme: The static grouping-based scheme has excellent performance in the local environment, and the data to be transmitted in the workflow is the least, so it still shows a good speed in computer cluster. At the same time, however, it is also very affected by the speed of the network. Increasing the speed of the network can significantly improve its speed in computer cluster.

(2) Pipeline-based scheme: The pipeline-based parallelization scheme performs poorly in the local environment because it is affected by the bottleneck steps in the pipeline and affects the overall execution speed. Figure 3 shows processing time of each stage, and it is obvious that removing the issue took over half of time of preprocessing. This shortcoming also exists in computer cluster, and at the same time it is affected by the time-consuming transmission. However, in pipeline-based scheme, only the most influential bottleneck steps and transmission time are reflected in the execution speed. Therefore, as the network speed increases, the impact of transmission time is gradually reduced, and the impact of the bottleneck step gradually becomes the main influencing factor of the execution speed, so the execution speed becomes close to the execution speed in the local environment.

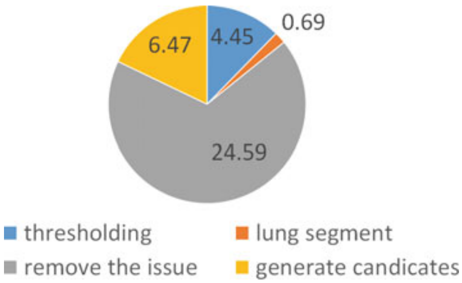


Fig. 3. Processing time of each stage

(3) Bus-based scheme: The bus-based parallelization scheme is the most seriously affected by the transmission time. As Fig. 2b shows, bus-based scheme with four execution units in a 10 MB/s environment is not even as fast as a single process, and overall, it is slower than a pipeline-based solution. The reasons for this include: First, the design of the bus-based solution is based on frequent message transmission for more uniform execution, so it is particularly affected by the transmission speed; second, for being more precise to control the operation time, the execution unit needs to report to the control unit after the execution of the current processing step, and the address of the execution unit of the next step is to be obtained from the execution unit, so that the transmission time cannot be covered by the execution time, thereby causing the effect of transmission time is further amplified.

4 Conclusion

Based on the existing CAD system, this paper conducts a parallelization study on the preprocessing part of the system to find out the time-consuming parallelization that can most effectively reduce the preprocessing time.

We propose three parallelization schemes from the aspects of data and instructions, and tested these three schemes on an existing CAD system at the end of the paper. Firstly, we tested the static grouping-based scheme, and a good result was obtained. The processing time was shortened by making full use of the computing power of the processor. The pipeline-based scheme was then tested, and the process of repairing the lung parenchyma took too much time, causing the entire pipeline to be blocked at this stage, resulting in poor performance. Finally, a bus-based scheme is tested. The performance is similar to that of static data-based scheme. When the execution unit is small, the processing is slightly more, but the execution efficiency can be maintained as the execution unit increases. Compared with the three schemes, the pipeline-based parallelization architecture cannot achieve the desired results due to the uneven time-consuming distribution of the stages before the reduction of false positives stage in the CAD system; the static grouping-based scheme performs best, but with the number of execution units increasing, the execution time is affected by the unevenness of the groups and increases slightly; the bus-based scheme is less efficient than the static grouping-based scheme due to the loss of data transmission, but its advantage lies in its execution efficiency is not affected by data grouping, and execution efficiency is maintained with the increase of execution units. In general, the static grouping-based scheme can achieve good enough results in most cases, but when there are more execution units and the cost of data transmission is not high, using a bus-based parallelization architecture can achieve better results.

References

1. Bray F, Ferlay J, Soerjomataram I, Siegel RL, Torre LA, Jemal A (2018) Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: Cancer J Clin 68(6):394–424
2. Valente IRS, Cortez PC, Neto EC, Soares JM, de Albuquerque VHC, Tavares JMRS (2016) Automatic 3D pulmonary nodule detection in CT images: a survey. Comput Methods Programs Biomed 124:91–107
3. Saxena S, Sharma S, Sharma N (2016) Parallel image processing techniques, benefits and limitations. Res J Appl Sci Eng Technol 12(2):223–238
4. Ying W, Cunxi C, Tong J, Xinhe X (2009) Segmentation of regions of interest in lung CT images based on 2-D OTSU optimized by genetic algorithm. In: 2009 Chinese control and decision conference. IEEE, pp 5185–5189
5. Sasidhar B, Ramesh Babu DR, Bhaskarao N (2013) Automated segmentation of lung regions and detection of lung cancer in CT scan. Int J Eng Adv Technol 2(4):757–758
6. Dehmeshki J, Amin H, Valdivieso M, Ye X (2008) Segmentation of pulmonary nodules in thoracic CT scans: a region growing approach. IEEE Trans Med Imaging 27(4):467–480

Predicting the Trajectory Tracking Control of Unmanned Surface Vehicle Based on Deep Learning

Wenli Sun[1](✉) and Xu Gao[2]

[1] Navigation College, Dalian Maritime University, Dalian 116026, China
itslab@dlmu.edu.cn
[2] Dalian Maritime Navigation Co., LTD., Dalian 116026, China

Abstract. The waypoint behavior is a method of trajectory tracking control for unmanned surface vehicle (USV). We take the waypoint behavior as the research object, choose speed, steering angle, capture radius, slip radius, lead distance, lead damper as features, and establish a prediction model for trajectory tracking control of USV based on deep neural network. The model effectively predicts the navigation effect and provides assistance and reference for the maneuvering decision of the USV. Test results show that the predicted results and the reference samples have same tendency. The proposed model can improve the performance of trajectory tracking control for the USV.

Keywords: USV · Deep learning · Waypoint behavior · Trajectory tracking control

1 Introduction

In recent years, unmanned surface vehicle (USV) has been widely used in military and civilian fields, playing an important role in port defense, maritime safety, marine surveying, and mapping. For USV, autonomous control technology is a core issue. It is the key performance that USV can execute tasks such as trajectory tracking, automatic collision avoidance, and cooperative formation [1]. Among them, the trajectory tracking control is a broad and important research area [2].

Many researches have used neural network in USV's control problem, particularly for trajectory tracking control, instead of traditional control systems which are based on identifying and modeling a nonlinear and complicated system. Arjun Gupta et al. adopt deep reinforcement learning to learn autonomous behaviors without relying on explicit models [3]. Cunhe Li et al. present an adaptive radial basis function neural network controller for nonlinear course control of USV, which can enable the USV to track the designed heading angle and turning rate [4]. A neural network is used to control the USV for course tracking by Yan Wang et al., solving the uncertainty problem of ship motion [5]. Signe Moe et al. train a neural network to mimic a feedback linearizing speed controller for an autonomous ship, which outperforms the traditional controller in terms of modeling errors and measurement noise [6]. A deep neural network is used to learn the automatic maneuver features of the USV based on vision system by Xu et al. [7].

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 591–598, 2020.
https://doi.org/10.1007/978-981-15-0187-6_70

The rapid development of artificial intelligence and deep learning provides new ideas and new methods for unmanned autonomous navigation. Yize Chen et al. propose a data-driven control method based on deep neural networks, which greatly improves the control efficiency and accuracy of complex systems [8]. The NVIDIA AutoPilot team use a deep neural network to map the raw images obtained from the front camera to the instructions for car autopilot, which achieve good results in simulators and road tests [9].

As described above, the artificial neural network is accessed to the control loop, to control the ship motion directly. We propose a prediction model based on deep learning and artificial neural network, which predicts the waypoint behavior effect and provides assistance and reference for the maneuvering decision of the USV, instead of traditional method in which the neural network is a part of ship motion control system. According to the waypoint behavior, the deep neural network model is established, combined with the navigation data samples of different speeds, steering angles, and other parameters of the USV in the simulated environment of Mission-Oriented Operating Suite-Interval Programming (MOOS-IvP). The deep learning method is used to train and solve the problem of multi-dimensional vector space decision-making involved in the trajectory track control process of the vehicle, making decisions for its autonomous navigation.

2 Background

2.1 The MOOS-IvP Architecture

MOOS-IvP is an open-source project aiming to provide a general autonomy architecture for autonomous system research. MOOS is a suite of software modules that coordinate software processes running on an autonomous platform. IvP is a technique for solving multi-objective optimization problems.

The MOOS-IvP software design is based on three architecture philosophies [10], (a) the backseat driver paradigm, (b) publish and subscribe autonomy middleware, and (c) behavior-based autonomy.

2.2 Waypoint Behavior

In the 1990s, Rodney Brooks proposed an inclusive structure that simplified the control of agents into several task-based behaviors. Based on this, the autonomous control of USV can be transformed into waypoint behavior, patrol behavior, fixed-point behavior, and trailing behavior. Among them, the waypoint behavior is the most basic, and many other behaviors are improved on this basis [3].

(1) Definition

The waypoint behavior means that the USV moves along a series of specified points in the *X*-*Y* plane, controlled by trajectory tracking.

As shown in Fig. 1a, the connection between each two adjacent waypoints is the shortest track, also known as the planned route. The USV sails autonomously according to the waypoint behavior, that is, sailing along the planned route. With the waypoint, there are two concentric circles, where the inner circle is called the capture circle and the outer circle is called the slip circle. When the USV sails into the capture circle, it is considered that the waypoint has been reached. Due to the influence of the wind, waves, and current on the sea, the USV may not be able to reach the capture circle. For this reason, a slip circle is set up. In the harsh sea environment, the USV is considered to reach the waypoint if it enters the slip circle.

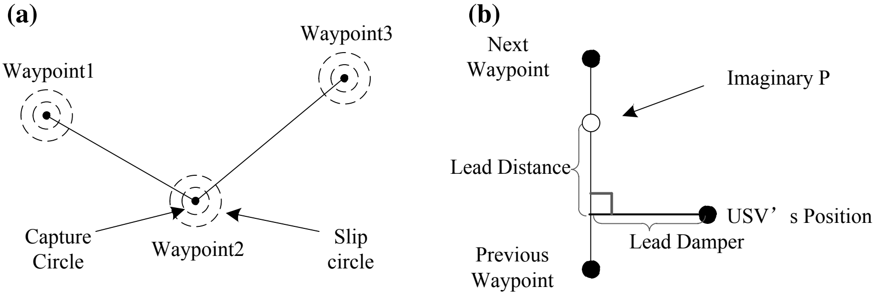


Fig. 1. USV's waypoint behavior. **a** Capture circle and slip circle; **b** guidance principle

(2) Guidance principle

USV's waypoint behavior adopts the principle of forward guidance control, as shown in Fig. 1b. The guidance algorithm involves two parameters: lead distance and lead damper. The lead distance is the distance from the foot to the imaginary point P on the planned route, and the point P is an imaginary point which is set on the planned route in the direction of the next waypoint. Based on this imaginary point, the waypoint behavior outputs the optimal steering angle. If the lead distance is greater than the distance to the next waypoint, point P is replaced by the next waypoint. Lead damper represents the distance from the planned route, i.e., the distance between the foot and the position of the ship. When the USV sails outside the predefined lead damper threshold, it turns toward the planned route to reduce the drift distance and the degree of deviation.

(3) Evaluation criteria

The evaluation criteria for the waypoint behavior is the maximum drift distance of the USV from the waypoint. According to the angle of the planning route, it can be divided into three cases: acute angle, right angle, and obtuse angle. From the view of the actual navigation, except for the special sea area, the acute angle of the planned route is barely appeared, and the acute angle is usually converted into two obtuse angles. Therefore, we assume that the angle of the planned route in the waypoint behavior is the right angle or the obtuse angle.

In Fig. 2, the point W_0 is the previous waypoint, the point W_1 is the current waypoint, and the point W_2 is the next waypoint. W_0W_1 is the route before steering, W_1W_2 is the route after steering, and the angle of $W_0W_1W_2$ is denoted by α, which is the supplementary angle of current steering angle.

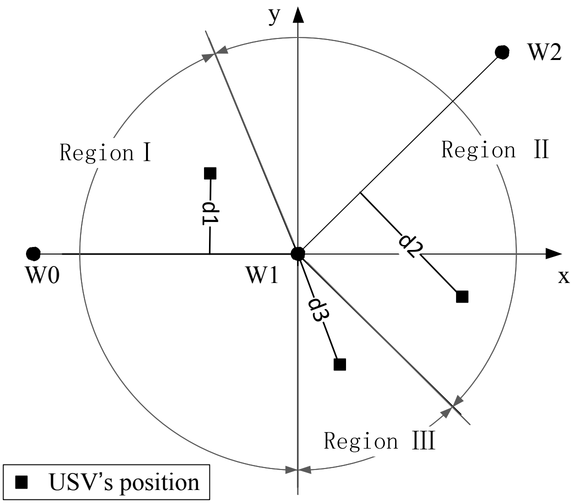


Fig. 2. Drift distance of waypoint behavior in different region of waypoint

When USV sails on the route of $W_0W_1W_2$ and steers to the point W_2, the trajectory tracking will be in one of the three regions. The angle range of different regions and the maximum drift distance are shown in Table 1.

Table 1. Definition of trajectory region and its waypoint drift distance

Region	Range of angle	Maximum drift distance
I	$\left[-90^\circ, \frac{\alpha}{2}\right)$	The maximum length of vertical line between USV's position and W_0W_1 (d_1)
II	$\left[\frac{\alpha}{2}, \alpha + 90^\circ\right)$	The maximum length of vertical line between USV's position and W_1W_2 (d_2)
III	$[\alpha + 90^\circ, 270^\circ)$	The maximum length between USV's position and W_1 (d_3)

In each region, the maximum drift distance of USV is denoted by d_i, $\mathrm{i} = 1, 2, 3$, and the maximum drift distance between the USV and the waypoint is $d_{\max} = \max\{d_i\}, i \in \{1, 2, 3\}$.

(4) Control process

The process that USV sails along the waypoints to follow the track has four steps:

(1) Guidance: The waypoint behavior guidance algorithm outputs the speed and steering commands, guiding the USV toward the next waypoint.
(2) Control: It converts the speed and steering commands into throttle and rudder angle commands.
(3) the USV is affected by wind and waves during the voyage.
(4) the performance is the actual trajectory track.

3 Deep Learning for Waypoint Behavior of USV

The trajectory tracking control of USV is complex. There are two main reasons: Firstly, different engines are installed in different vehicles, making it hard to use linear model for converting speed and steering angle into throttle and rudder angle. And secondly, the actual track of the vehicle is mighty affected by the wind and current, which is a nonlinear effect. Traditional methods such as Lyapunov's methods or passivity-based theorems can only approximate rather than solve it fundamentally.

In addition, there are many factors in the implementation of waypoint behavior, which belongs to the multi-dimensional vector space decision problem, and the influence of various factors cannot be given through the linear or high-dimensional space plane division. Therefore, by using the deep learning method to establish a deep neural network model, the relationship between the trajectory tracking control and the behavior parameters of the waypoint can be better mined. After training, the deep neural network model has the ability to predict the trajectory tracking effect, that is, if multi-dimensional feature is entered, the model can export the effect level for the maneuvering decision of the USV and improve the track following performance.

3.1 Feature Selection

By analyzing the waypoint behavior for USV, there are six features, which can be divided into three categories, as shown in Table 2.

Table 2. Features and categories of waypoint behavior for USV

Number	Features	Categories
1	Capture radius	Relate to the definition of waypoint
2	Slip radius	
3	Lead distance	Relate to guidance algorithm
4	Lead damper	
5	Speed	Relate to navigation
6	Steering angle	

3.2 Dataset Generation

We use MOOS-IvP to generate the dataset of the waypoint behavior. In order to be able to approximate the real ocean environment, we simulate the influence of wind and current by generating periodic random external forces on the vehicle. The USV performs the waypoint behavior under different steering angles and speeds. The specific method is:

(1) Set the steering angle at the intervals of 15° and plan the route.

(2) Set the speed at the interval of 10 knots, and under the same steering angle conditions, do the experiments at 10, 20, and 30 knots speeds, respectively.

(3) Adjust the parameters related to the waypoint behavior (capture radius, slip radius) and the relevant parameters of the guidance algorithm (lead distance, lead damper) at a certain steering angle and a certain speed.

(4) Calculate the maximum drift distance of the waypoint for each experiment.

(5) Set the drift distance evaluation criterion and divide the effect of the waypoint into five levels. The drift range belongs to Level I within 1 times vehicle's length, as shown in Fig. 3a. The drift range belongs to Level V within 5 times vehicle's length, as shown in Fig. 3b, and other levels are in range of them.

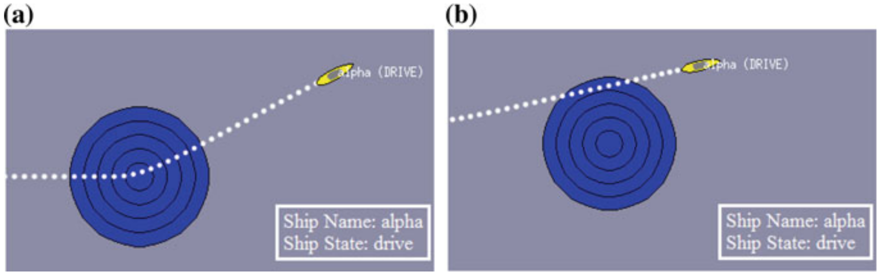


Fig. 3. Effect of waypoint behavior. **a** USV's drift range belongs to Level I; **b** USV's drift range belongs to Level V.

3.3 Fully Connected Network Architecture

The artificial neural network used in this paper is an eight-layer forward fully connected neural network $N_{6\text{-}6\text{-}7\text{-}7\text{-}8\text{-}7\text{-}6\text{-}5}$. The hidden layer uses the Relu function as the activation function. This paper uses cross-entropy as a loss function to determine how close the output vector to the expected vector is. The output layer uses the softmax function. The neural network outputs a five-dimensional array. Each dimension in the array corresponds to one trajectory track level.

3.4 Training

With TensorFlow, we use the optimizer of the gradient descent algorithm to train the neural network. We set the learning rate to 0.001 and batch size to 100.

We train the model for 1800 iterations. A total of 3420 samples are randomly selected from the sample dataset as the training set, and 730 samples are taken as the test set. In order to prevent over-fitting, the training process outputs an accuracy result on the verification set after every 100 iterations and saves the weight parameters of the model. After the training, the deep neural network model with the highest accuracy on the verification set is saved as the USV waypoint behavior model.

4 Results

The neural network structure with different depths and widths was used to observe the prediction accuracy on the verification set and the test set. Experiments show that the deep neural network with 6-6-7-7-8-7-6-5 nodes in each layer has the best performance. It achieves a maximum accuracy of 88.2% on the verification set and 84.9% on the test set.

In the experiment, we use the trained neural network to predict the USV in the MOOS-IvP platform, to track a waypoint route including different speeds and steering angles. The experimental results are as follows.

Figure 4a, b shows the predicted levels with different steering angles when other variables are fixed. Figure 4a is the reference samples from the test set, and Fig. 4b is the predicted value.

Figure 5a, b shows the predicted levels with different speeds when other variables are fixed. Figure 5a is the reference samples from the test set, and Fig. 5b is the predicted value.

We can see that the predicted results and the reference samples have the same tendency.

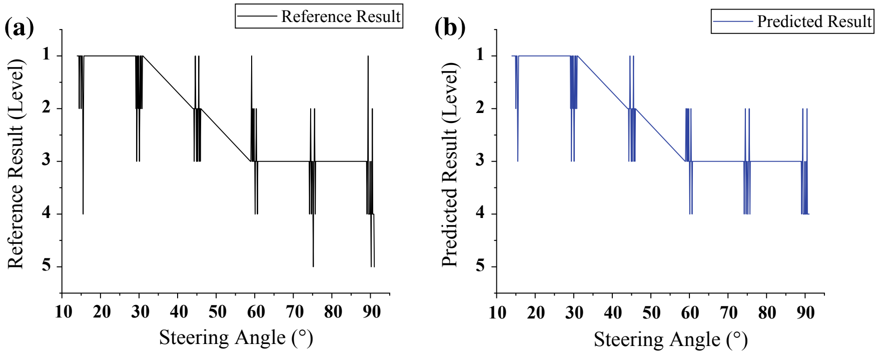


Fig. 4. Drift level of steering angles. **a** Reference level from test set; **b** predicted level from model

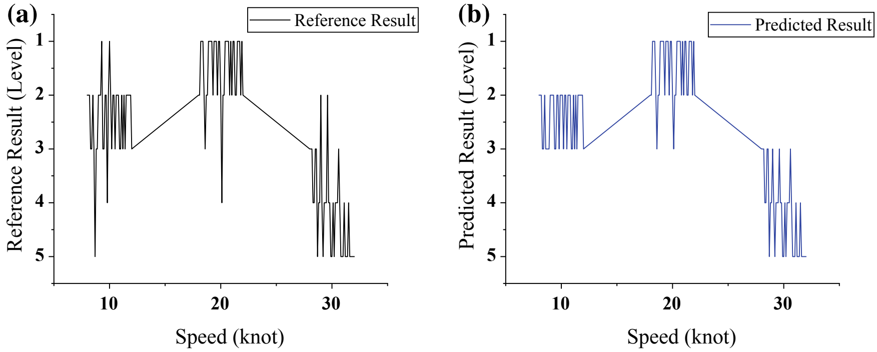


Fig. 5. Drift level of speeds. **a** Reference level from test set; **b** predicted level from model

5 Conclusions

Based on the deep learning, the waypoint behavior of USV has been taken as the research object. The speed, steering angle, capture radius, slip radius, lead distance, and lead damper have been selected as the features to establish the trajectory tracking control model of the waypoint behavior. By using the experimental sample data obtained from MOOS-IvP for training, the following conclusions have been drawn:

(1) Based on the deep learning method, we can effectively solve the complex problems brought by the environment, equipment, and algorithms faced by the USV.
(2) The prediction model can judge the influence of different feature parameters on the waypoint behavior of the USV, and the prediction accuracy can reach 84.9%.

References

1. Švec P, Thakur A (2014) Target following with motion prediction for unmanned surface vehicle operating in cluttered environments. Auton Robot 36:383–405
2. LaValle SM (2006) Planning algorithms. Cambridge University Press, Cambridge. http://planning.cs.uiuc.edu
3. Gupta A, Novitzky M, Benjamin M (2018) Learning autonomous marine behaviors in MOOS-IvP. In: Oceans 2018 MTS/IEEE Charleston
4. Li C (2016) Adaptive RBF neural network control for unmanned surface vessel course tracking. In: Sixth international conference on information science and technology
5. Wang Y, Tong J, Song TY, Wan ZH (2018) Unmanned surface vehicle course tracking control based on neural network and deep deterministic policy gradient algorithm. In: Oceans-MTS/IEEE Kobe Techno-Oceans (OTO)
6. Moe S, Rustad AM, Hanssen KG (2018) Machine learning in control systems: an overview of the state of the art
7. Xu QY, Yang YQ, Zhang CJ, Zhang L (2018) Deep convolutional neural network-based autonomous marine vehicle maneuver. Int J Fuzzy Syst 20(2):687–699
8. Chen YZ, Shi YY, Zhang BS (2019) Optimal control via neural networks: a convex approach. To Appear in International conference on learning representations
9. Marius B (2019) End-to-end deep learning for self-driving cars. http://devblogs.nvidia.com/deep-learning-self-driving-cars
10. Benjamin MR, Newman P, Schmidt H, Leonard JJ (2009) An overview of MOOS-IvP and a brief users guide to the IvP helm autonomy software. MIT Department of Mechanical Engineering, Technical report MIT-CSAIL-TR-2009-028

Coordinated Learning for Lane Changing Based on Coordination Graph and Reinforcement Learning

Hao Zuo(✉), Huaiwei Si, Nan Ding, Xin Wang, and Guozhen Tan

Dalian University of Technology, Dalian, China
1507154320@qq.com

Abstract. The rapid development of autonomous driving has aroused widespread interest in academia and industry. Due to vehicular mobility, it is not feasible to adopt the existing coordinated learning approach based on static topology directly. To solve this issue, we propose a coordinated learning approach based on dynamic graph and reinforcement learning to enable distributed learning of cooperative lane changing. The dynamic graph model is constructed by evaluating the driving risk between each pair of vehicles. The lane change decision making of the vehicles is guided by the global optimal action based on the dynamic coordinated graph. Experiments verify that the proposed approach can achieve the accuracy and safety of lane change decision making, and with the increase in the number of vehicles, the approach has good scalability.

Keywords: Coordination graph · Reinforcement learning · Lane change · Autonomous driving

1 Introduction

Automated driving is one of the most important artificial intelligence applications, which attracts a lot of interest from technology companies and research institutions [2]. Baidu Apollo Project, Google Self-Drive Car and DARPA Urban Challenge et al. show great potential for autonomous driving. In the field of autonomous driving, lane-changing processes have received great attention [1]. Lane change decision is considered to be essential driving behaviors that affect both traffic safety [16] and efficiency [9]. However, due to the variability of road traffic conditions, even with extensive expert experience, manual design rules to deal with all possible lane change situations may lead to overly simple and sub-optimal decision.

The above problem can be solved using the reinforcement learning (RL) method. A large number of studies [5, 15] have applied RL to autonomous driving. However, all of these studies only focus on learning the driving decisions for a single vehicle, which contradicts the fact that autonomous driving is a typical multi-agent system (MAS) [17]. Some scholars [4, 13] have studied how to design control strategies through vehicles communication to coordinate multiple vehicles. However, all of these

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 599–607, 2020.
https://doi.org/10.1007/978-981-15-0187-6_71

studies just apply simple distributed RL into a multi-vehicle environment, in which it will lead to inefficient and uncoordinated driving decisions [6].

This paper aims to solve the coordination lane change problem by using multi-agent RL (MARL) technology. Since the main strategies of autonomous driving are car following and lane changing for overtaking [18], we pay more attention to the study of lane changing and car-follow decision on the road. We use the coordination graph (CG) to achieve the cooperative decision making for multi-vehicle. However, due to the motion characteristic of the vehicles continuously change topology, the direct application of CG to autonomous driving is a thorny issue. In order to solve this problem, we propose a MARL approach based on the combination of dynamic graph and RL, which use driving risk evaluation and graph theory to model the interaction between vehicles, then use the consistency of latent conflicts relations to achieve effective distributed coordination learning. The experiment verifies that the proposed approach allows the vehicles to achieve higher driving safety under cooperative decision making. Compared with other approaches that have no coordination or rely on some mobile models and expert driving rules, the proposed approach is more advantageous.

2 Related Work

RL has been widely used in the field of autonomous driving. Some researchers have used RL to learn driving strategies. For example, Wang et al. [20] use deep reinforcement learning with rule-based constraints to learn driving strategies. Other work uses Deep RL to make driving decisions directly from real-world scenarios, such as virtual to real reinforcement learning for autonomous driving [14]. All of these studies only focus on the driving strategies of a single vehicle, without considering the interactions and coordination between multiple vehicles.

A large number of scholars have studied lane changing before, such as Moriarty and Langley [11] used RL to solve the lane selection problem of multiple vehicles. Although these methods have good results in their respective applications, they either focus on the relatively simple traffic environment or do not simulate the interaction between multiple vehicles. Some studies [12] have proposed methods to simulate vehicle interaction and formation using graph theory, but these methods all focus on formation and communication. Recently, automobile safety accidents happen frequently, so our work is more dedicated to driving safety issues. We realize distributed learning in multi-vehicle through dynamic CG and RL, which can provide more safe and accurate lane-changing decision for autonomous driving.

3 Related Theories and Approaches

Traditional coordination graph can only model closed, static problems, which does not change during the solution process. In order to realize coordinated learning in dynamic environments, we propose an approach to construct dynamic coordination graph and a MARL approach to solve it.

3.1 MDP Formalization for Autonomous Driving

The decision-making process of each vehicle can be modeled as the Markov Decision Process (MDP) by four tuples of (S, A, P, R). The state S is mainly composed of factors that affect the decision of the vehicles, which mainly includes the following parts: the lane l in which the vehicle is currently located; the remaining reaction time t_i of the vehicle and the neighboring vehicles. The state of each vehicle consists of a five tuple of $(l, t_i)\,(i = 1, 2, 3, 4)$ that can be computed from the data illustrated in Fig. 1.

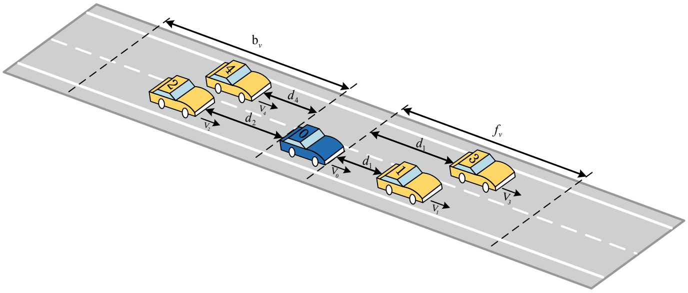


Fig. 1. State of vehicle CAR0

Finally, the most important safety in autonomous driving is used to evaluate the chosen actions. We define the reward function as follows:

$$r_s = \begin{cases} -40 & \text{if collision} \\ \min(t_1, t_2) & \text{if } l = 0 \text{ and } d_1 > 3 \text{ and } d_2 > 3 \\ \min(t_3, t_4) & \text{if } l = 1 \text{ and } d_3 > 3 \text{ and } d_4 > 3 \\ -5 & \text{else} \end{cases} \tag{1}$$

In the experiment, we will use function approximated RL based on neural network to solve the above-mentioned MDP problem and compare it with the famous mobile model Mobil [7] and expert method [10] to verify the validity of the MDP formulation.

3.2 Dynamic Coordination Graph for Multi-vehicle

In the dynamic environment, the dependencies between agents change rapidly as the state changes. In order to apply the CG to the continuous domain and realize the coordination among the agents, we propose an approach of constructing the dynamic graph model based on evaluating the driving risk between each pairs of the vehicles.

A general highway scene is used to illustrate how to construct the coordination relation between the two vehicles. The vehicle 1 is the lag vehicle, the vehicle 2 is the lead vehicle, the speeds of vehicle 1 and vehicle 2 are v_1 and v_2, respectively, and the following distance between the two vehicles is d. The formula [19] for evaluating the driving risk between two vehicles is as follows:

$$\mathrm{DR} = \frac{GR_1R_2M_1M_2}{d^{k_1}}(1 + D_{r1})(1 + D_{r2})\exp[-k_2(v_1 - v_2)] \tag{2}$$

DR is the driving risk evaluation between vehicle 1 and vehicle 2. In order to facilitate comparison of follow-up experiment results, we assume that the vehicles have the same attributes. Here, we chose the following set of parameters: $G = 0.001$, $k_1 = 1$, $k_2 = 0.05$, $M_1 = M_2 = 5000\,\mathrm{kg}$, $D_{r1} = D_{r2} = 0.2$, $R_1 = R_2 = 1$. For constructing an effective dynamic CG (DCG) model without producing too complex graph structure, $\mathrm{DR}_{\mathrm{safe}}$ is set to 180 kN here. When $\mathrm{DR}_1 > \mathrm{DR}_{\mathrm{safe}}$, it is necessary to construct the coordination relation for the two vehicles because of the great driving risk. We define this relation, which is built by the greater risk of driving between the two vehicles, as the latent conflict relation.

The basic analysis approach between the above pair of vehicles is extended to all pairs of vehicles in the graph. According to the DCG we constructed, it can be clearly seen which pairs of vehicles have latent conflict relation in driving safety as the edge of the DCG.

3.3 Pretraining of Local Cooperative Control Model

In order to solve the problem of high complexity and consistency in DCG, we found some aspects of latent consistency based on the above approach. According to the above approach, there are the consistent latent conflict relation between the pairs of vehicle as edges in the graph, that is, the same utility function can be used to calculate the payoff between the pairs of vehicles.

We decompose the large-scale CG problem with multiple agents into multiple homologous sub-problems with the same structure and small scale. By solving the sub-problems and reusing them on other sub-problems, we can solve the large-scale problems. Therefore, we decompose the large-scale global payoff function Q_{global} in the DCG into multiple local payoff functions $Q_{ij}(i, j \in E)$. The local payoff functions Q_{ij} can be solved by the local cooperative control model as the source problem. The local cooperative control model defined on the two vehicles is put into the traffic flow for pretraining. The reuse of the local cooperative control model on other edges can achieve the purpose of solving the payoff functions in the complete DCG.

In each time step, Eq. (2) is used to calculate whether there are the latent conflict relation between all pairs of vehicles, and the pair of vehicle with the greatest driving risk is controlled by the local cooperative control model. The vehicles with the small risk of driving are controlled by the individual neural network. Generate the following update rule:

$$\Delta\omega_i = \alpha\left[r_i + \gamma \max q_i(s'_i, a'_i, \omega_i) - q_i(\mathsf{s}_i, \mathsf{a}_i, \omega_i)\right] \times \nabla_{\omega_i} q_i(s_i, a_i, \omega_i) \tag{3}$$

The vehicles with the greatest driving risk are updated by the following rules:

$$\Delta\omega_{ij} = \alpha\left[r + \gamma \max Q_{ij}(s', a', \omega) - Q_{ij}(\mathsf{s}, \mathsf{a}, \omega)\right] \times \nabla_{\omega_{ij}} Q_{ij}(s_{ij}, a_{ij}, \omega_{ij}) \tag{4}$$

Through pretraining, the local cooperative control model can make coordinated decisions on the basic pair of vehicles in various scenarios and thus can accurately calculate the coordination payoff functions between pairs of vehicles with latent conflicting relations.

3.4 The Coordinated Learning Mechanisms

Autonomous driving can be achieved by using MARL technology. Coordination graph (CG) [8] is an effective way to solve the MARL problems. In order to reduce the computational complexity, the global payoff function can be decomposed into $Q^{\pi}(s,a) = \sum_{i \in V} q_i(s_i, a_i) + \sum_{(i,j) \in E} Q_{ij}(s_{ij}, a_i, a_j)$, where V defines the set of vertices that is equivalent to the set of vehicles, E defines the set of edges, $q_i(s_i, a_i)$ defines the individual payoff function of vehicle i, and $Q_{ij}(s_{ij}, a_i, a_j)$ defines the payoff function of neighboring vehicles (i,j).

Although several coordinated RL methods have been proposed previously [3], it is not feasible to directly apply these methods to the multi-vehicle coordination problem because the dependencies of moving vehicles are continuously changing. Therefore, inspired by action-based methods, we propose to use the above DCG model to guide the driving strategy of the autonomous vehicle and realize distributed coordinated learning between multiple vehicles. Previous work used manual definition of expert rules to guide the action selection of agents. However, it is difficult to define accurate expert rules to guide coordinated decision making in actual collaborative environment. So, we guide the autonomous vehicle to make decisions through the global optimal decision of the DCG. Our autonomous vehicles are controlled using an independent neural network while using the local cooperative control model after pretraining to calculate the local $Q_{ij}(s_{ij}, a_i, a_j)$ value for each edge.

In each time step, the DCG is constructed for vehicles with latent conflicting relations. Each vehicle uses the global payoff function $\arg\max_{ja} Q_{\text{global}}$ when the agent action is selected. The decision of the vehicles is made by maximizing the global payoff function but using its own local payoff function to update the vehicles' q_i. It can be explained as guiding the vehicles to explore toward the global optimal state-action pairs. The neural network of each vehicle can be updated in the following ways:

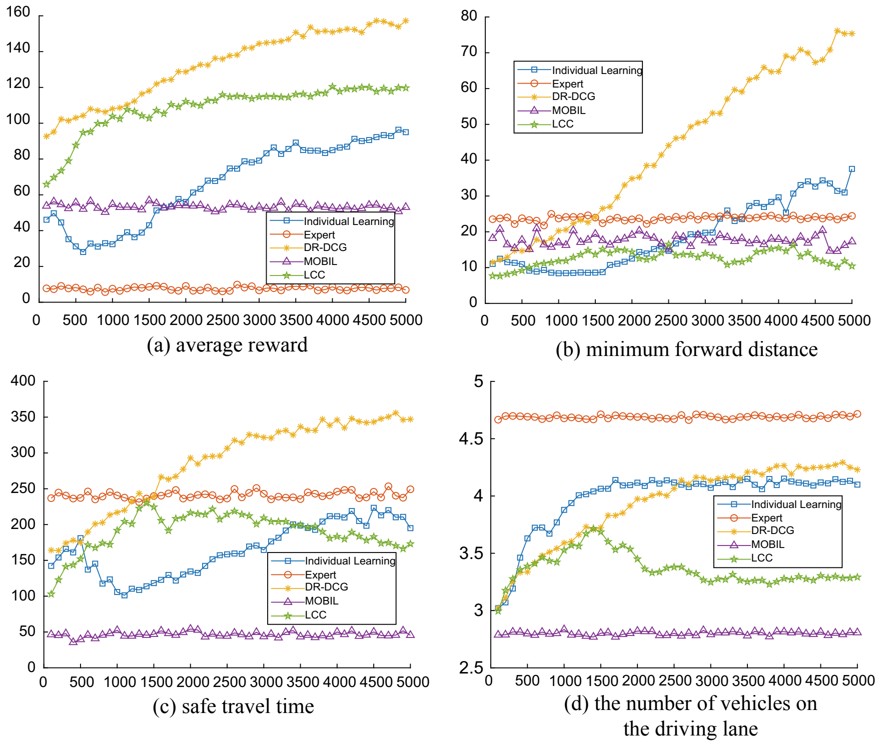
$$\Delta w_i = \alpha[r_i + \gamma \max_{ja} q_i'(s', a_{ja}', w) - q_i(s, a_{ja}, w)] \nabla_{w_i} q_i(s_i, a_i, w_i) \tag{5}$$

4 Experimental Evaluation

In order to verify the effectiveness of the above MDP formulation, local cooperative control model and dynamic coordination graph, we compare the learning result of RL method, the famous mobility model MOBIL [7] and the expert rule-based approach

[10]. DR-DCG learning denotes the coordinated learning approach proposed by us, and LCC denotes the local cooperative control model. We simulate the learning process at two-lane highway.

We averaged more than 10 runs as the final result. Since individual learning of optimal behavior may not guarantee the best group performance, the comparison with individual learning approach shows the effectiveness of the DR-DCG approach in decision making of cooperative lane changing. In the proposed approach, based on the independent MDP, the two-vehicle local cooperative control model is transferred to the edge of the DCG we constructed after pretraining and calculated all the payoff contribution between the pairs of vehicle on the edges. Figure 2a shows the learning of the average reward over five vehicles. It can be seen from the figure that the local cooperative control model can obtain better results than individual learning, indicating that it can accurately calculate the coordination payoff contribution between the pairs of vehicles that have latent conflicts relations. More obvious is that the DR-DCG approach we proposed can achieve a higher average reward than the other approaches, which shows the benefits of coordination between vehicles. Figure 2b–d plots the microscopic mobility of different approaches include the minimum forward distance, safe travel time and number of vehicles on the driving lane.



(a) average reward

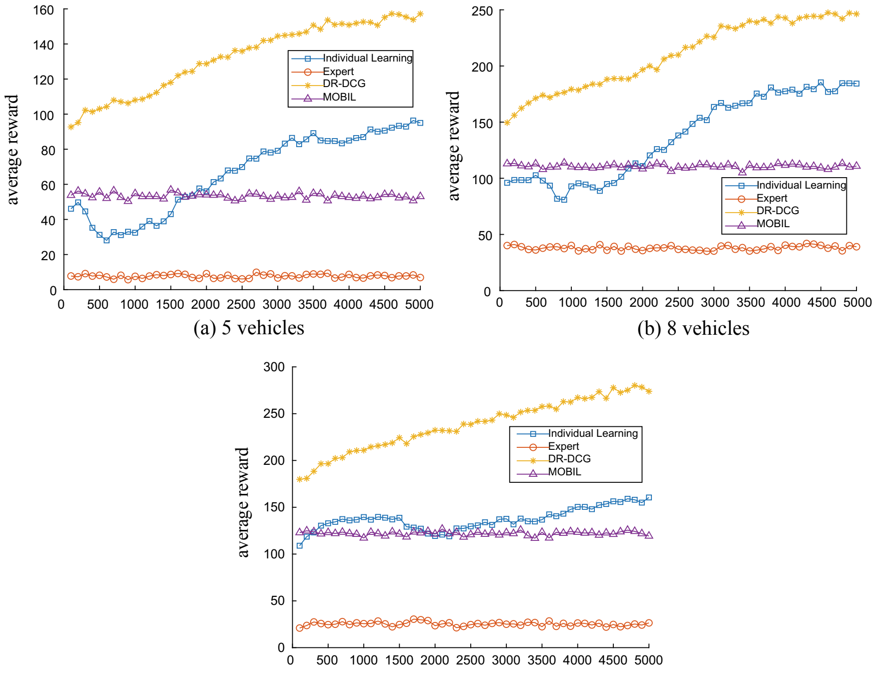
(b) minimum forward distance

(c) safe travel time

(d) the number of vehicles on the driving lane

Fig. 2. Average reward and microscopic mobility of different approaches

In order to verify the scalability of the approaches, we apply the above approach to different vehicle densities environments. The parameter settings are the same as before. We gave the results of five vehicles, eight vehicles and 11 vehicles in turn. Figure 3 shows that as the number of vehicles increases, DR-DCG approach always obtains higher average rewards, indicating that it has better scalability and can be extended to more complex situations. In summary, through the analysis of vehicle motion characteristics and experiments of different vehicle densities, it shows that our approach can effectively solve the latent conflict risk between vehicles by coordination lane changing and further verifies the effectiveness of our approach.



(a) 5 vehicles

(b) 8 vehicles

(c) 11 vehicles

Fig. 3. Average reward of different vehicle densities with regard to learning episode

5 Conclusions

Autonomous driving is a typical multi-agent domain, but the application of RL to multiple vehicles coordination problem has been a challenging task due to the changing topology of vehicles. This paper proposes a MARL approach based on the combination of DCG and RL. This approach contributes to the safety and coordination lane change decision making of autonomous driving which are increasingly concerned by people. Our work has the following specific points: (1) Propose an approach to construct DCG for vehicles with latent conflicts based on driving risk evaluation, (2) Propose a distributed MARL approach for multi-vehicle coordination lane change decision making under dynamic environment. Most of the current research in MARL is still concentrated in static and closed environments, but our work is carried out in an open and dynamic environment, which provides a prospective direction for future research. Our approach is validated by lane change problems, which have better security and scalability than other approaches.

Acknowledgements. This work was supported in part by the National Science Foundation of China under Grant No. 61471084, National Key R&D Program of China under Grant No. 2018YFB1700102 and Fundamental Research Funds for the Central Universities No. DUT19JCC29.

References

1. Cao P, Hu Y, Miwa T et al (2017) An optimal mandatory lane change decision model for autonomous vehicles in urban arterials. J Intell Transp Syst 21(4):271–284
2. Fagnant DJ, Kockelman K (2015) Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transp Res Part A Policy Pract 77:167–181
3. Guestrin C, Lagoudakis M, Parr R (2002) Coordinated reinforcement learning. ICML, vol 2, pp 227–234
4. Hult R, Zanon M, Gros S et al (2018) Optimal coordination of automated vehicles at intersections: theory and experiments. IEEE Trans Control Syst Technol
5. Isele D, Cosgun A, Subramanian K et al (2017) Navigating intersections with autonomous vehicles using deep reinforcement learning, May 2017. http://arxiv.org/abs/1705.01196
6. Kalantari R, Motro M, Ghosh J et al (2016) A distributed, collective intelligence framework for collision-free navigation through busy intersections. In: 2016 IEEE 19th international conference on intelligent transportation systems (ITSC). IEEE, pp 1378–1383
7. Kesting A, Treiber M, Helbing D (2007) General lane-changing model MOBIL for car-following models. Transp Res Rec 1999(1):86–94
8. Kok JR, Vlassis N (2006) Collaborative multiagent reinforcement learning by payoff propagation. J Mach Learn Res 7:1789–1828
9. Li X, Sun JQ (2017) Studies of vehicle lane-changing dynamics and its effect on traffic efficiency, safety and environmental impact. Physica A 467:41–58
10. Li X, Xu X, Zuo L (2015) Reinforcement learning based overtaking decision-making for highway autonomous driving. In: 2015 sixth international conference on intelligent control and information processing (ICICIP). IEEE, pp 336–342
11. Moriarty DE, Langley P (1998) Learning cooperative lane selection strategies for highways. AAAI/IAAI 1998:684–691
12. Olfati-Saber R, Murray RM (2002) Graph rigidity and distributed formation stabilization of multi-vehicle systems
13. Plessen MG, Bernardini D, Esen H et al (2016) Multi-automated vehicle coordination using decoupled prioritized path planning for multi-lane one-and bi-directional traffic flow control. In: 2016 IEEE 55th conference on decision and control (CDC). IEEE, pp 1582–1588
14. Pan X, You Y, Wang Z et al (2017) Virtual to real reinforcement learning for autonomous driving. arXiv preprint arXiv:1704.03952
15. Sallab AEL, Abdou M, Perot E et al (2017) Deep reinforcement learning framework for autonomous driving. Electron Imaging 2017(19):70–76

16. Schorr J, Hamdar SH, Silverstein C (2017) Measuring the safety impact of road infrastructure systems on driver behavior: vehicle instrumentation and real world driving experiment. J Intell Transp Syst 21(5):364–374
17. Shalev-Shwartz S, Shammah S, Shashua A (2016) Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295
18. Toledo T (2007) Driving behaviour: models and challenges. Transp Rev 27(1):65–84
19. Wang J, Wu J, Li Y (2015) The driving safety field based on driver–vehicle–road interactions. IEEE Trans Intell Transp Syst 16(4):2203–2214
20. Wang J, Zhang Q, Zhao D et al (2019) Lane change decision-making through deep reinforcement learning with rule-based constraints. arXiv preprint arXiv:1904.00231

Pretreatment Method of Deep Learning Method for Target Detection in Sea Clutter

Yanan Xu, Ningbo Liu(✉), Hao Ding, Yonghua Xue, and Jian Guan

Naval Aviation University, Erma Road 188, Yantai 264001, Shandong, China
{yt_yananxu, lnb198300}@163.com

Abstract. Sea target detection is widely used in military and civilian fields. Because of the space-time non-stationary characteristics of high resolution radar sea clutter, traditional target detection methods have many limitations and are limited by the use of scenarios. In recent years, with the progress of deep learning in image classification tasks, a series of target detection methods based on deep learning have emerged. By applying these methods to target detection in radar sea clutter, high accuracy and good generalization can be obtained. However, there are many new problems when these target detection methods, mostly based on computer vision, are introduced to target detection in radar sea clutter due to different data forms and detection standards. This paper mainly discusses the effects of different pretreatment modes of target detection in sea clutter using classification target detection framework on training speed and detection accuracy.

Keywords: Deep learning · Target detection · Sea clutter · False alarm rate

1 Introduction

Effective detection of sea targets under sea clutter background is of great significance in both military and civilian fields. In order to meet the needs of target detection, radar needs to have high resolution in range and azimuth; high resolution sea clutter has non-stationary characteristics, and contains a large number of sea peaks, which seriously affects the accuracy of target detection. At the same time, traditional adaptive matched filtering methods need clutter to meet certain model assumptions and have a small application range [1]. Although the method based on fractal theory is simple and efficient, it needs to accurately estimate the interval with fractal characteristics and needs long observation time [2]. In the actual detection of sea targets, only for the specific target type under the background of specific clutter has a good detection accuracy. Since AlexNet [3] put forward in 2012, a large number of target detection algorithms based on deep learning have been produced, including classification-based detection algorithm based on convolutional neural network (CNN) and regression-based target detection framework [4].

Deep learning target detection methods are mostly proposed for object detection in two-dimensional images in computer field, which are often used in face recognition, automatic driving, etc. [4]. Although radar display screen has different visualization forms such as A-display and P-display, they are all obtained by radar echo signal

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 608–614, 2020.
https://doi.org/10.1007/978-981-15-0187-6_72

processing. Radar echo signal is a one-dimensional discrete sequence. A radar data file usually contains data of several distance units for a certain observation time, which determines that there are many input forms under the conditions of translation invariance and locality required for classification using convolutional neural network. In order to better use the deep learning method for target detection in radar sea clutter, several feasible input forms are introduced in this paper.

2 Pretreatment Mode

2.1 One-Dimensional Form

Common radar data sets, such as IPIX data sets [5] and CSRI data sets [6], usually store the values of sampling points in matrix form. In the longitudinal direction, the sampling points are determined by the pulse repetition frequency (PRF) and the observation time, while in the transverse direction, they correspond to each distance unit. Each column of data is a discrete one-dimensional time series, so it can input in the form of one-dimensional time series. In addition, there are many commonly used processing methods for time series, such as time-frequency transform, joint time-frequency analysis, etc. [7]. These methods can be used to preprocess the data before the input network, so as to improve the discrimination between target data and clutter data, and then improve the accuracy.

2.1.1 Direct Input

Each range unit of radar data file usually contains thousands of sampling points (such as CSIR data set) or tens of thousands of sampling points (such as IPIX data set). Due to the limited amount of data, it is unrealistic to input all sampling points of a distance unit directly. A certain echo data can be intercepted on a distance unit with a fixed preset length as a data sample. As shown in Fig. 1, each distance unit is slipped for interception, and after interception, it is transferred to the next distance unit for interception until enough data samples are intercepted. When the amount of data is insufficient, the sliding step can be smaller than the preset length; when the amount of data is sufficient, the sliding step can be larger than the preset length. Compared with the target detection in clutter by input radar P-display images, direct input of radar echo sequence samples can make better use of the original information contained in the echo sequence, but requires a higher signal-to-clutter ratio as support. At the same time, according to the specific data set, we need to determine the value of the preset length through a large number of experiments, which is too short to contain enough data for classification, and too long to ensure that there are enough samples to form a training set to achieve convergence. So it is necessary to find a balance between the amount of information in each data sample and the amount of data in the data set. In addition, in the process of sample interception, it is inevitable that the texture information of radar echo data will be destroyed at some interceptions.

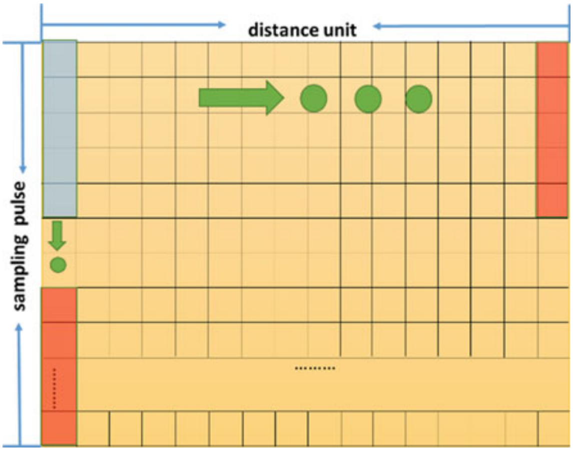


Fig. 1. Direct input in one-dimensional form

In order to achieve better classification effect, besides directly inputting these data samples into CNN for classification, some classical signal transformation methods can be used for preprocessing and inputting.

2.1.2 Input After Pretreatment

In order to visualize the influence of preprocessing on discrimination, the time–magnitude and frequency–magnitude diagrams of IPIX data sets before and after transformation are drawn, respectively, as shown in Fig. 2. Figure 2a is a distance unit without target and Fig. 2b is a distance unit with target. It can be seen that after FFT processing, the clutter has a positive Doppler translation due to the wind direction, and the target is more concentrated than the clutter energy distribution. Compared with the time domain before pretreatment, the frequency domain has better distinguishing features. The intercepted samples can be transformed by FFT of the same number of points, and then input into CNN network for classification. Compared with the direct input intercepted echo sequence, the classification effect has been significantly improved. In high sea conditions, the spectrum of some small floating targets will overlap with the clutter spectrum in varying degrees, as shown in Fig. 3. Even in the clutter spectrum, it is difficult to get high detection accuracy by using FFT method to preprocess. However, the target detection in sea clutter can be better achieved by using in-depth learning method through appropriate preprocessing method, i.e. signal processing method, to improve the quality of data samples.

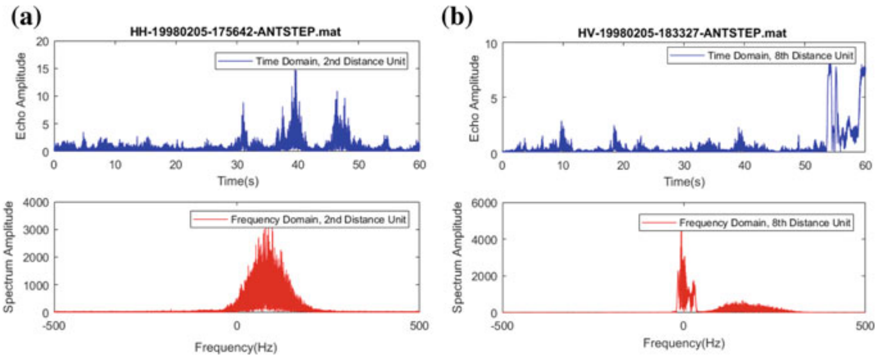


Fig. 2. Comparison of two kinds of echo sequences processed by FFT or not

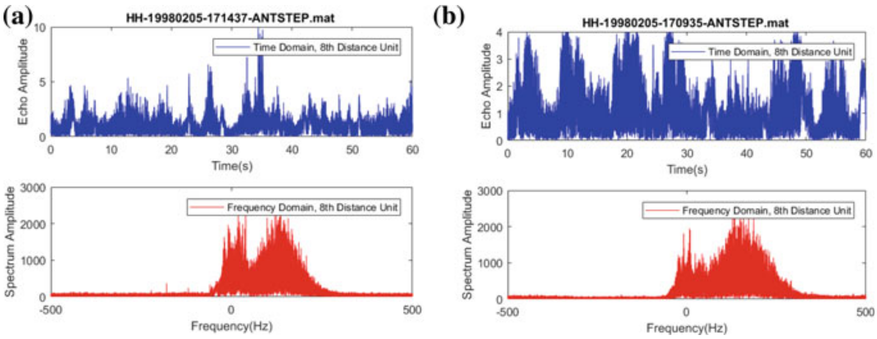


Fig. 3. Two examples of overlap between target spectrum and clutter spectrum

2.2 Two-Dimensional Form

Using one-dimensional input CNN for classification will greatly reduce the computational complexity of the network and have faster training speed, but it will also split the spatial–temporal correlation and destroy certain texture information. CNN is good at classifying two-dimensional graph structure data. Some radar image signals, such as SAR radar data and P-display image data, can be input in the form of pictures, similar to those in the field of face recognition, and have the same processing form, including normalization processing and so on. In this section, we discuss the method of obtaining suitable two-dimensional input samples from the time–distance matrix of radar echo data.

2.2.1 Input After Direct Interception

The premise of using CNN for classification is that the data samples conform to the hypothesis of locality and weight sharing, while the time–distance matrix of radar echo data has temporal and spatial correlation [8], which satisfies the basis of using CNN for classification. Similar to the interception method in Sect. 2.1.1, we use sliding interception with preset step size to obtain rectangular samples of the same size, as shown in Fig. 4. Rectangular samples may not be square. Similarly, when the amount of data is insufficient, the interception step can be smaller than the rectangular side length, allowing overlap. Compared with one-dimensional input, two-dimensional input can make use of temporal and spatial correlation at the same time, but it also needs to determine the interception step and the size of data samples according to the actual data. It is necessary to ensure that there are enough features for classification as well as the amount of data.

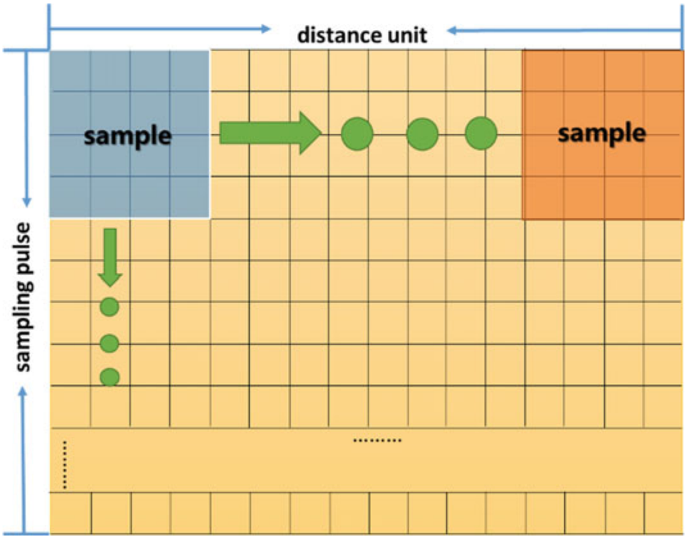


Fig. 4. Direct input in one-dimensional form

2.2.2 Input After Transform to Two-Dimensional

In addition to directly intercepting the two-dimensional matrix, one-dimensional samples can be transformed into two-dimensional forms through certain signal analysis methods to better reflect the characteristics of samples. FFT is a commonly used signal analysis method, which can get the characteristics of signal in frequency domain. However, the actual signal is often non-stationary. Frequency domain can not represent the change of signal spectrum with time. It is necessary to use the joint function of time and frequency to represent signal, that is, time–frequency analysis of signal [9]. Time–frequency analysis includes linear time–frequency analysis and non-linear time–frequency analysis. Typical linear time–frequency analysis mainly includes STFT, WT and Gabor expansion. Non-linear time–frequency analysis mainly includes various quadratic representation methods, such as WVD, Cohen class and so on. According to the number of sampling segments, time–frequency analysis is carried out on the intercepted one-dimensional radar echo sequence samples, and the data samples whose size is determined by the number of sampling segments and sampling points are obtained. After time–frequency analysis, the discrimination of the two types of data samples has been significantly improved, among which SPWVD is the best. Also take the IPIX data set as an example, as shown in Fig. 5.

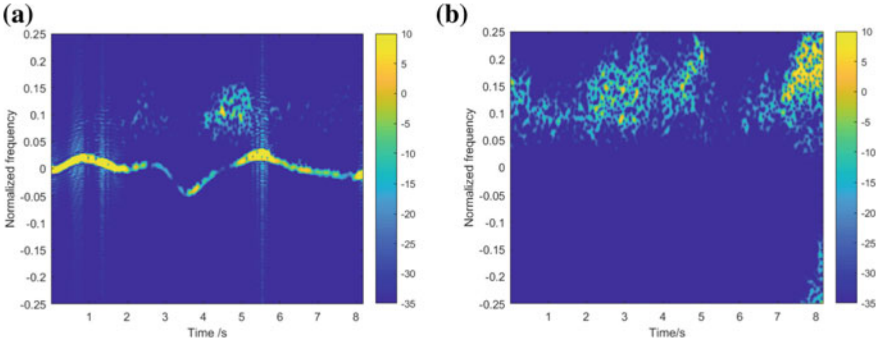


Fig. 5. Two-dimensional samples processed by SPWVD

Compared with the two kinds of samples processed by FFT in Fig. 2, the discrimination of the two kinds of samples obtained by time–frequency analysis is obviously improved. It is obvious that sea clutter gathers in the banded area with texture characteristics near the normalized center frequency of 0.1, and the target is the "bright strip" with pseudo-periodic characteristics. Using CNN can extract this contour feature well and achieve high accuracy.

3 Summary

In this paper, aiming at the specific application scenario of target detection in sea clutter, we discuss the difference between the deep learning method and the traditional application scenario in the selection of input mode, accuracy calculation, and the method to improve the accuracy without adjusting the CNN structure.

In terms of pretreatment mode, besides directly inputting radar SAR image or P-display image, one-dimensional time series sample input can also be obtained by interception, and the sample quality can be improved by certain signal processing methods, thus obtaining higher accuracy. It can also use the temporal and spatial correlation of sea clutter data to intercept two-dimensional sample input. Similarly, it can use time–frequency analysis and other methods to obtain two dimensions of feature representation and input network.

References

1. He Y, Huang Y, Guan J et al (2014) Overview of radar target detection technology in sea clutter. Modern Radar 36(12):1–9 (in Chinese)
2. Ding H, Dong YL, Liu NB et al (2016) Overview and prospects of research on sea clutter property cognition. J Radars 5(05):499–516 (in Chinese)
3. Simonyan K, Vedaldi A, Zisserman A (2014) Learning local feature descriptors using convex optimisation. IEEE Trans Pattern Anal Mach Intell 36(8):1573–1585
4. Lu HT, Zhang QC (2016) Summary of research on the application of deep convolution neural network in computer vision. J Data Acquis Process 31(1):1–17 (in Chinese)
5. Drosopoulos A (1994) Description of the OHGR database. Defence Research Establishment, Ottawa, Canada, Tech. Note. No. 94-14
6. Anderson F (2007) Awarenet: persistent, ubiquitous surveillance technologies for enhanced national security. Brochure, CSIR, Pretoria, South Africa, Oct 2007. Available: http://www.csir.co.za/dpss/brochures.html
7. Cui SP (2014) Target detection in sea clutter background based on time-frequency analysis. Harbin Institute of Technology, Harbin, pp 25–32 (in Chinese)
8. Tough JA, Ward KD, Shepherd PW (2005) The modelling and exploitation of spatial correlation in spiky sea clutter. In: IEEE radar conference, 2005, EURAD 2005, European, Nov 2005. https://doi.org/10.1109/eurad.2005.1605550
9. Shi SN, Shui PL (2018) Sea-surface floating small target detection by one-class classifier in time-frequency feature space. IEEE Trans Geosci Remote Sens 99:1–17

A Novel Endmember Extraction Method Based on Manifold Dimensionality Reduction

Xiaoyan Tang(✉) and Shangzheng Liu

School of Electronics and Electrical Engineering, Nanyang Institute
of Technology, Nanyang 473004, China
547717964@qq.com

Abstract. Because of multiple reflection and scattering, the mixed pixels in hyperspectral images are actually nonlinear spectral mixing. Traditional endmember extraction algorithm is based on linear spectral mixing model, so the extraction accuracy is not high. Aiming at the nonlinear structure of hyperspectral images, a novel endmember extraction method for hyperspectral images based on Euclidean distance and nonlinear dimensionality reduction is proposed. This method introduces Euclidean distance of image into the nonlinear dimensionality reduction algorithm of local tangent space permutation to remove redundant spatial information and spectral dimensional information in hyperspectral data and then extracts the endmembers from the reduced data by searching for the maximum volume of the simplex. Experiments on real hyperspectral data show that the proposed method has a good effect on hyperspectral image endmember extraction, and its performance is better than that of linear dimensionality reduction PCA and original LTSA algorithm.

Keywords: Hyperspectral image · Nonlinear dimensionality reduction · Euclidean distance of image · Endmember extraction

1 Introduction

Due to the limitation of spatial resolution of hyperspectral imaging spectrometer and the complexity and diversity of natural objects, some pixels of remote sensing images often contain many types of objects, namely mixed pixels [1]. How to effectively decompose mixed pixels has become an important direction in remote sensing research. Decomposition of mixed pixel includes two steps: endmember extraction and abundance estimation. Endmember extraction is the precondition of mixed pixel decomposition. At present, there are many endmember extraction algorithms, including PPI, VCA, N-FINDR [2], IEA [3], MVT, MNF [4, 5], and so on. These algorithms are based on the linear spectral mixing model. However, due to the nonlinearity of ground scattering, multiple scattering in a pixel, and the heterogeneity of sub-pixel components, hyperspectral data has inherent nonlinearity [6, 7].

One of the strategies to deal with the nonlinearity of hyperspectral images is nonlinear dimensionality reduction. In Ref. [4], isometric mapping (ISOMAP), a classical manifold learning algorithm, is introduced into hyperspectral image processing to replace PCA or MNF to reduce the dimensionality nonlinearly of

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 615–621, 2020.
https://doi.org/10.1007/978-981-15-0187-6_73

hyperspectral data and achieve better classification results. Document [8] uses spectral angle and spectral information divergence to replace Euclidean distance and combine with geodesic distance for feature extraction. Referring to the ISOMAP algorithm, the paper [9] proposes a maximum simplex volume decomposition algorithm based on geodesic distance. Although the intrinsic low-dimensional structure embedded in high-dimensional data can be obtained by using ISOMAP, these algorithms only consider the spectral characteristics of hyperspectral images and ignore their spatial information. Document [9] put forward an improved N-FINDR algorithm based on manifold learning and spatial information, which adds spatial information to the nonlinear dimensionality reduction data of local tangent space alignment (LTSA), but only the spectral information of hyperspectral images is considered in the dimensionality reduction process. In this paper, the LTSA algorithm is improved by using Euclidean distance of hyperspectral images, which combines the spatial information of hyperspectral image. The dimension of hyperspectral image is reduced nonlinearly, and the dimension-reduced data is extracted by referring to N-FINDR algorithm. The proposed algorithm takes full account of the nonlinear structure of hyperspectral images and the spatial information of images and improves the accuracy of endmember extraction.

2 Local Tangent Space Arrangement Algorithm

Local tangent space arrangement algorithm (LTSA) [10] is a classical manifold learning algorithm proposed by Zhenyue Zhang et al. For a given data set $[x_1, x_2, \ldots, x_N]$, $x_i \in R^m$, m dimension, N points, the LTSA algorithm can obtain its embedded low-dimensional manifold $y_i \in R^d (d < m)$.

The specific algorithm of LTSA is as follows:

(1) Construct neighborhoods for each point x_i. In the original high-dimensional input space, the nearest k neighbor to each point is found by Euclidean distance, so k-nearest neighbor of x_i is set as $X_i = \{x_{i1}, x_{i2}, \cdots, x_{ik}\}$ (including x_i).

(2) Obtain local tangent space coordinates. The local tangent space coordinates $\Theta_i = \left[\theta_1^{(i)}, \ldots, \theta_k^{(i)}\right]$ of x_i are calculated by Formula (1).

$$\min \sum_{j=1}^{k} \left\| x_{ij} - \left(\bar{x} + Q_i \theta_j^{(i)}\right) \right\|_F^2 = \min \left\| X_i - \left(\bar{x} 1_k^T + Q\Theta\right) \right\|_F^2 \tag{1}$$

Among them $\bar{x}_i = \frac{1}{k}\sum_{j=1}^{k} x_{ij}$, Q_i is the matrix composed of the left singular vectors of the largest d singular values of the centralized matrix $X_i - \bar{x}_i 1_k^T = [x_{i1} - \bar{x}_i, \ldots, x_{ik} - \bar{x}_i]$. The optimal solution of the formula is $\Theta_i = Q_i^T \left(X_i - \bar{x}_i 1_k^T\right)$.

(3) Obtain global low-dimensional embedded coordinates. Let Y_i be the global coordinates of the corresponding data points in X_i. Firstly, the alignment technique is

used to align the local tangent space coordinates and the global low-dimensional embedded coordinates by minimizing the local error shown in Formula (2).

$$E_i = \min_{L_i \in \mathbb{R}^{d\times d}} \left\| Y_i \left(I - \frac{1}{k} 1_k 1_k^T \right) - L_i \Theta_i \right\|_F^2 \tag{2}$$

3 Models and Methods

The original LTSA algorithm uses Euclidean distance to find the nearest k neighbor to each point. It is widely used for feature extraction and recognition of high-dimensional data such as face and handwriting. However, for hyperspectral image data, the distribution of objects in the image is continuous, and the spectrum of each endmember is closely related to its surrounding background pixels. According to the spatial characteristics of images, Wang et al. [11] proposed Euclidean distance of images in 2005. The core idea of Euclidean distance is to introduce spatial relations into the calculation of image block distance. The improved method [12] is applied to calculate the distance between hyperspectral image blocks. For hyperspectral data sets $X \in R^{W\times H\times L}$ with horizontal width W, vertical width H, and band number L, the Euclidean distance formula between data points $x_{l,r} \in R^{L\times 1}$ and $x_{p,q} \in R^{L\times 1}$ is as follows, when 3 * 3 image blocks are used:

$$d(x_{l,r}, x_{p,q}) = \sqrt{\sum_{m=1}^{3}\sum_{n=1}^{3}\left[\sum_{i=1}^{3}\sum_{j=1}^{3} g_{ij,mn}\left(x_{l-2+i,r-2+j} - x_{p-2+i,q-2+j}\right)^{\mathrm{T}}\left(x_{l-2+m,r-2+n} - x_{p-2+m,q-2+n}\right)\right]} \tag{3}$$

The matrix G of 9 * 9 is shown in Eq. (4). Among them, t is a spatial factor, which represents the influence degree of spatial relations. Because the image distance is not sensitive to t, it is usually 3 ~ 6.

$$g_{ij,mn} = \exp\left(-\frac{|i-m|^2 + |j-n|^2}{2t} \right) \tag{4}$$

It can be seen from Formulas (3) and (4) that the Euclidean distance of the image gives a good distance relationship between the eight neighborhoods centered on the observed pixels in hyperspectral images. In practical applications, when the distribution of objects is concentrated or large, the spatial relationship between the pixels in the image block is more significant, so the spatial factor t is smaller; on the contrary, the larger t is needed.

In this paper, Euclidean distance of image is used to replace Euclidean distance in LTSA to reduce the dimension of hyperspectral data. Referring to N-FINDR endmember extraction algorithm [13], for nonlinear dimensionality reduction data, random selection of p pixels spectral vector $A_1, A_2, \ldots, A_p$ constitutes a simplex $S(A_1, A_2, \ldots, A_p)$ and calculates the volume of the simplex. Each pixel is used to replace each currently selected spectral endmember in turn. If a substitution can obtain a larger volume of the simplex, the substitution is retained as an effective substitution. Otherwise, it is eliminated as an invalid substitution, and finally, P endmembers are obtained.

4 Experimental Results

In this section, actual hyperspectral data is used to test the performance of the algorithm. The proposed algorithm is compared with the following two-dimensionality reduction algorithms: PCA and LTSA. The data of dimensionality reduction is combined with N-FINDR endmember extraction algorithm to extract endmembers. The performance of these three algorithms is evaluated by the accuracy of endmember extraction.

4.1 AVIRIS Cuprite Data

AVIRIS Cuprite data is the data of Cuprite mining area in Nevada Prefecture obtained in July 1995. The size of the hyperspectral image data is 250 × 191, 224 bands (0.4–2.5 um). Spatial resolution and spectral resolution are 20 m and 10 nm, respectively. In this experiment, we intercept sub-graphs of 50 × 50. The remaining 188 bands were used after removing the bands with low water-gas absorption and signal-to-noise ratio. Pseudo-color image of experimental data (R: band 40, G: band 28, B: band 10) is shown in Fig. 1.

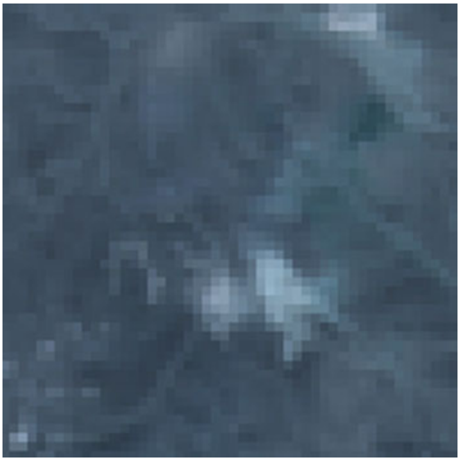


Fig. 1. False-color image of the AVIRIS Cuprite data set

Using NWHFC method [14] in the false alarm probability $P_F = 10^{-6}$ or less, the number of endpoints is calculated to be 4. According to the field survey map, the four main mineral components in Fig. 1 are calcareous garnet, blueline, muscovite, and banyan.

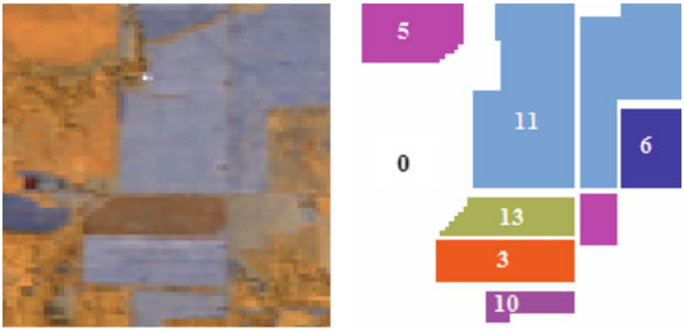
According to the distribution of real objects, spectral angle distance (SAD) and average spectral angle distance (ESAD) are calculated using the endmember spectra extracted by three-dimensionality reduction methods and the corresponding spectra in the USGS spectral library, which can show accuracy of the three algorithms. Table 1 shows the SAD and the ESAD distance between the endmembers extracted by the three algorithms and the corresponding spectrum in the USGS spectral library, where t = 3, k = 15. As shown in Table 1, LTSA and the proposed algorithm are superior to PCA in terms of the nonlinear structure of hyperspectral images. The proposed algorithm combines the spatial information of the image with the nonlinear dimensionality reduction, so its ESAD is smaller than the other algorithms, as shown in bold. That means the extracted endmembers by proposed method is the closest ones comparing with real endmembers.

Table 1. SAD-based similarity scores between spectral signatures contained in the USGS library and their corresponding endmembers extracted in AVIRIS Cuprite data set

Mineral in USGS	PCA	LTSA	Algorithm proposed in this paper
Calcareous garnet	0.147	0.097	0.085
Blueline	0.139	0.076	0.098
Muscovite	0.098	0.157	0.101
Banyan	0.131	0.138	0.146
ESAD	0.130	0.121	**0.110**

4.2 AVIRIS Indian Pine Data

The AVIRIS Indian Pine image has a spectral resolution of 10 nm and a spectral range of 0.4–2.5 M. It has 220 spectral bands with a spatial resolution of 17 m and an image size of 145 × 145 [15]. After removing the water absorption bands, 202 bands are remained for further processing. In order to reduce the computational time complexity, a sub-image with the size of 70 × 70 is intercepted from the left corner of the original image. Its pseudo-color image is shown in Fig. 2a, and the real value of the ground is shown in Fig. 2b. In order to quantitatively evaluate the performance of the algorithm, the reference spectra of each endmember are extracted artificially from the image according to the real distribution of the objects. As shown in Fig. 2b, besides the background, there are six endelements, Corn-Notill (3), Grass/pasture (5), Grass/trees (6), Soy-Notill (10), Soy-Mintill (11), and Wheat (13). We extract four points at the center of each type of pixel and use their mean as the reference spectrum of the endmember.



（a）False-color image （b）real data in the ground

Fig. 2. 70 × 70 subset of the AVIRIS Indian Pine data set

Table 2. SAD-based similarity scores between real spectral signatures and their corresponding endmembers extracted in AVIRIS Indian Pine

Endmembers	PCA	LTSA	Algorithm proposed in this paper
Corn-Notill (3)	0.027	0.057	0.072
Grass/pasture (5)	0.125	0.061	0.146
Grass/trees (6)	0.374	0.209	0.132
Soy-Notill (10)	0.224	0.059	0.172
Soy-Mintill (11)	0.060	0.253	0.065
Wheat (13)	0.056	0.035	0.121
ESAD	0.188	0.141	**0.124**

In this experiment, t = 4, k = 15 are selected to compare the SAD values between the real spectrum and the estimated spectrum obtained by the three algorithms as shown in Table 2. It can be shown in Table 2 that the optimal ESAD is obtained by the proposed algorithm, as shown in bold.

5 Conclusion

Aiming at the nonlinear characteristics of hyperspectral data, a novel endmember extraction method of hyperspectral image based on nonlinear dimensionality reduction of Euclidean distance is proposed. This method considers the physical characteristics of hyperspectral data and introduces Euclidean distance into the nonlinear dimensionality reduction algorithm of local tangent space arrangement to better remove redundant spatial information and spectral dimensional information in hyperspectral data sets. Experiments on real hyperspectral data show that the proposed method has a good effect on hyperspectral image endmember extraction, and its performance is better than that of linear dimensionality reduction PCA and original LTSA algorithm. Due to the Euclidean distance of the image, the complexity of the algorithm increases. How to improve the efficiency of the algorithm needs further study.

References

1. Keshava N, Mustard JF (2002) Spectral unmixing. IEEE Sig Process Mag 19(1):44–57
2. Plaza A, Martínez P, Pérez R et al (2004) A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. IEEE Trans Geosci Remote Sens 42(3):650–663
3. Neville RA, Staenz K, Szeredi T et al (1999) Automatic endmember extraction from hyperspectral data for mineral exploration. In: Proceedings of the 21st Canadian symposium. remote sensing, pp 21–24
4. Jia S, Qian Y (2009) Constrained nonnegative matrix factorization for hyperspectral unmixing. IEEE Trans Geosci Remote Sens 47(1):161–173
5. Sun X, Cai J, Xu Z et al (2013) Hyperspectral unmixing based on nonnegative matrix factorization. Opto-Electron Eng 39(12):97–102
6. Bioucas-Dias JM, Plaza A, Dobigeon N et al (2012) Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. IEEE J Sel Top Appl Earth Obs Remote Sens 5(2):354–379
7. Bachmann CM, Ainsworth TL, Fusina RA (2005) Exploiting manifold geometry in hyperspectral imagery. IEEE Trans Geosci Remote Sens 43(3):441–454
8. Du P, Wang X, Tan K et al (2011) Dimensionality reduction and feature extraction from hyperspectral remote sensing imagery based on manifold learning. Geomatics Inf Sci Wuhan Univ 36(2):148–152
9. Tang X, Gao K, Ni G et al (2013) An improved N-FINDR endmember extraction algorithm based on manifold learning and spatial information. Spectrosc Spectral Anal 33(09):2435–2440
10. Zhang Z, Zha H (2002) Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. J Shanghai Univ 8(4):406–424
11. Wang L, Zhang Y, Feng J (2005) On the Euclidean distance of images. IEEE Trans Pattern Anal Mach Intell 27(8):1334–1339
12. Chen H, Pu H-Y, Wang B et al (2013) Image Euclidean distance-based manifold dimensionality reduction algorithm for hyperspectral imagery. J Infrared Millimeter Waves 32(5):450–455
13. Winter ME (1999) N-FINDR: an algorithm for fast autonomous spectral endmember determination in hyperspectral data. In: SPIE's international symposium on optical science, engineering, and instrumentation. International Society for Optics and Photonics, pp 266–275
14. Chang CI, Du Q (2004) Estimation of number of spectrally distinct signal sources in hyperspectral imagery. IEEE Trans Geosci Remote Sens 42(3):608–619
15. https://engineering.purdue.edu/∼biehl/MultiSpec/hyperspectral.html

Research on Chinese Short Text Clustering Ensemble via Convolutional Neural Networks

Haowen Wan, Bo Ning(✉), Xiaoyu Tao, and Jianfei Long

Dalian Maritime University, Dalian 116026, China
ningbo@dlmu.edu.cn

Abstract. Different from traditional text, short texts are characterized by high dimensionality, sparseness, and large text size. At the same time, some existing clustering ensemble algorithms treat each clusters equally, which will lead to the problem that the clustering results are not good enough. To solve this problem, this paper proposed a short text clustering ensemble algorithm based on convolution neural network (CNN). Firstly, the word2vec model is used to preserve the semantic relationship between words and obtain the multi-dimensional word vector representation; secondly, the feature is extracted from the original vector combining with the CNN; thirdly, clustering methods are used to cluster vectors; and then finally, Gini coefficient is used to measure the reliability of clustering, and the final clustering ensemble is carried out.

Keywords: Short text · CNN · Clustering ensemble

1 Introduction

Text clustering has always been a classical problem in the field of text mining. It is also the basis of natural language processing applications such as recommendation algorithms. A text will be treated as the short text when its vocabulary is less than 140 words. It is different from the traditional text. The vocabulary of short text is very few, many words even appear only once, and so it has the problem of sparsity. Traditional short text clustering methods such as vector space model [1] cannot solve the problems caused by high-dimensional and sparse features. Some researchers work on expanding and enriching the context of data from Wikipedia [2] or an ontology [3]. Some researchers have proposed short text clustering based on semantics [4, 5]. But these algorithms have some subjective choices and judgments more or less in vector representation, and the result is still high-dimensional word vector representation. With the development of deep learning in recent years, convolutional neural network (CNN) is widely used in image feature extraction, short text classification, and other fields, but it is seldom used in short text clustering process.

Various clustering algorithms have their own advantages and disadvantages. Few single clustering algorithms can be applied to any structured data. Compared with clustering, clustering ensemble combines multi-base clustering to obtain better clustering results, so clustering ensemble has attracted more and more people in recent years [6–10]. However, many existing clustering ensemble methods do not consider the reliability of each base clustering, which leads to the low quality of the final clustering

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 622–628, 2020.
https://doi.org/10.1007/978-981-15-0187-6_74

results. Some researchers consider the reliability of basic clustering, but consider a basic clustering as an individual without considering each cluster [9, 11, 12]. In this paper, Gini coefficient is used to evaluate the reliability of base clustering, and hierarchical clustering algorithm is used to integrate the final clustering.

To solve the above problems, this paper proposes a short text ensemble clustering method based on CNN. Firstly, the high-dimensional representation of word vectors is trained by word2vec method. Then, the feature vectors are extracted by CNN, and the lower-dimensional representation of word vectors is obtained. Next, the feature vectors are clustered by some traditional clustering methods. At last, the clustering ensemble algorithm proposed in this paper is used for clustering ensemble.

2 Vectorization of Short Text

2.1 Pre-processing Short Texts

For the Chinese text, the pretreatment process mainly includes the removal of punctuation marks, word segmentation, and removal of stop words. Punctuation is removed for subsequent word segmentation of the remaining plain text where word segmentation refers to the segmentation of a continuous Chinese text into separate entries. In Chinese texts, there are many words with high frequency but no specific meanings, such as “的” and “在,” which are called as stop words. The purpose of eliminating stop words is to preserve the core words as much as possible so that the subsequent generated word vectors are not affected by noise and improve the quality of word vectors.

2.2 Generating Word Vectors

The so-called word embedding refers to the technology mapping words into a new space and expressing them in the form of vectors. Traditional word vector representations, such as one-hot, are sparse and multifarious, which can cause dimension disasters and ineffective access to semantic information in practical use. Mikolov et al. proposed word2vec model which is based on neural network [13]. Word2vec model trains the language model through the context and context semantic relations of words and obtains word vector as the auxiliary product in the training process.

The word2vec has two important models—CBOW model and Skip-gram model. CBOW model uses context to predict center word, while Skip-gram model uses center word to predict context. This paper chooses CBOW model to train to get the vector representation of k-dimension of each word and fills all short texts according to the maximum length N in short texts, so that all texts are presented in the matrix form of $n * K$. The uniform format is also convenient for subsequent batch convolution operations.

2.3 Extracting Feature Vectors

The structure of neural network used to process short text in this paper includes input layer, convolution layer, max-pooling layer, and full connection layer.

The input layer is the word vector matrix obtained in the previous step. Different from the common input forms of images, each line in the word vector matrix represents a word. In order to ensure that the region slipped by convolution kernel is always a complete word, the convolution kernel is required to only slip vertically in height. Its width is guaranteed to be consistent with the dimension of the word vector. In this paper, we use convolution kernels of various window sizes to perform convolution operations and obtain vector representations of different sizes. Next, the operation of the max-pooling layer is processed; that is, the location of the maximum value in the upper results is selected. After the max-pooling layer is processed, a scalar is obtained. Then, the scalars processed from the same size convolution kernel are joined together to obtain the feature vectors under the same window size. The last layer is a fully connected layer with dropout. Dropout is mainly designed to prevent over-fitting and improve the generalization ability of the model. The final complete feature vectors are the representation of short text after processing by CNN and also serve as the input of clustering algorithm.

3 Clustering Ensemble of Short Text

Clustering is to discover the potential relationships between data structures and obtain more useful information. However, there is no single clustering algorithm that is able to deal with all types of data structures. Each clustering algorithm has its special advantages as well as weakness. Given the same data set, different clustering algorithms and even different parameters of the same algorithm will lead to different clustering results, but it is difficult to directly select any clustering results without prior knowledge. Clustering ensemble algorithm combines various clustering algorithms to obtain a more accurate clustering result. Clustering ensemble has shown advantages in finding bizarre clusters and dealing with noise [10]. In clustering ensemble, each clustering as input is called base clustering, and the final result is called consensus clustering.

3.1 Selecting Base Clustering

In this paper, several different types of classical clustering algorithms are selected as the base clustering. They are *K*-means algorithm based on partition, AGNES algorithm which is a typically hierarchical clustering algorithm, and DBSCAN algorithm which is based on density.

K-means algorithm is a traditional partition-based clustering algorithm. The main idea of the algorithm is roughly as follows: *K* samples are randomly selected from the sample set as clustering centers, and the distance between all samples and the *K* clustering centers is calculated. For each sample, it is divided into the clustering centers nearest to the center, and the new clustering centers of each cluster are calculated for the new clustering.

AGENS is a hierarchical clustering algorithm based on bottom-up clustering strategy. It considers each sample in the data set as an initial clustering and then combines the two nearest clusters at each step of the algorithm. The process repeats until the number of preset clusters is reached.

DBSCAN algorithm is one of the most typical representative algorithms in this kind of methods. Its core idea is to find the points with higher density first, then connect the similar high-density points step by step, and then generate various clustering.

3.2 Clustering Ensemble

In the process of ensembling, the result is easy to be affected by low-quality clustering. In order to solve this problem, this paper proposes to evaluate the reliability of each base cluster based on Gini coefficient and consider the internal structure of the cluster, instead of simply treating each cluster as an indivisible individual, so as to give them weight to achieve the goal of improving the credibility of consistent clustering. Finally, a consistent result is obtained by combining the hierarchical clustering method.

3.2.1 Gini Coefficient

Gini coefficient reflects the probability that two samples are randomly sampled from the sample set. It is a kind of index that can be used to show the purity of data. The smaller the Gini coefficient is, the purer the representative data is, that is to say, the higher the certainty of classification. It is defined as:

$$\mathrm{Gini}(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{1}$$

where K denotes the number of classification, the probability of object belongs to the kth category is p_k.

3.2.2 Formulation of the Ensemble Clustering Problem

In this section, we introduce the formulation of the clustering ensemble problem. Let $D = \{d_1, d_2, \ldots, d_N\}$ be the dataset, where d_i is the ith data and N is the number of data. For the dataset D, there are M partitions, each of which is treated as a base clustering and consists of a certain number of clusters. It can be defined as:

$$P = \{p^1, p^2, \ldots, p^M\} \tag{2}$$

where

$$p^m = \{C_1^m, \ldots, C_k^m\} \tag{3}$$

denotes the mth base clustering in P, C_i^m and denotes the ith cluster in p^m. Obviously, the union of all clusters in the same base clustering covers the entire data set. Let $F^m(d_i)$ denote the cluster in p^m that object d_i belongs to. That is, if d_i belongs to the kth cluster in p^m, i.e., $d_i \in C_k^m$, then we have $F^m(d_i) = C_k^m$. For convenience, we present the set of all clusters in the ensemble P as $\sigma = \{C_1, \ldots, C_n\}$.

3.2.3 Reliability of Clusters

As introduced in Sect. 3.2.1, Gini coefficient is a measure of uncertainty associated with a random variable. Each cluster is a set of data objects. Given a cluster C_i does not belong to p^m, then it is possible that the objects in C_i belong to more than one cluster in p^m. The uncertainty of C_i with respect to p^m can be computed by considering how the objects in C_i are clustered in p^m.

$$\mathrm{Gini}^m(C_i) = \sum_{j=1}^{n^m} p(C_i, C_j^m)\left[1 - p(c_i, c_j^m)\right] = 1 - \sum_{j=1}^{n^m} p^2(C_i, C_j^m) \tag{4}$$

with

$$p(C_i, C_j^m) = \frac{\left|C_i \cap C_j^m\right|}{|C_i|} \tag{5}$$

where n^m is the number of clusters in p^m, C_j^m and is the jth cluster in p^m. Given the ensemble P, the uncertainty of cluster C_i with the respect to the entire ensemble P is computed as:

$$\mathrm{Gini}^P(C_i) = \sum_{m=1}^{M} \mathrm{Gini}^m(C_i) \tag{6}$$

where M is the number of base clusterings in P. After defining the uncertainty of cluster, an index to measure the reliability of clustering through considering the uncertainty of clustering for the whole partition to measure the credibility is proposed which is called ROC. It can be defined as:

$$ROC(C_i) = e^{-\frac{\mathrm{Gini}^P(C_i)}{\theta \cdot M}} \tag{7}$$

where $\theta > 0$ is a parameter to adjust the influence of the cluster uncertainty over the index. Obviously, smaller uncertainty of a cluster leads to a greater ROC value.

3.2.4 Consensus Functions

The CA matrix [8] reflects how many times two objects are grouped into the same cluster in the ensemble. It is widely used for dealing with the ensemble clustering problem [6, 7]. But traditional CA matrix treats the clusters equally without considering their weight. Huang [9] exploited the NCAI index to weighted CA matrix, which only considers the reliability of base clusters, but does not consider the local diversity of clusters. In this paper, a new CA matrix is redefined based on the clustering reliability proposed in the previous section which is named ROC-CA matrix. It is defined as:

$$A = \{a_{ij}\}_{N \times N} \tag{8}$$

with

$$a_{ij} = \frac{1}{M} \cdot \sum_{m=1}^{M} \omega_i^m \cdot \xi_{ij}^m \tag{9}$$

$$\omega_i^m = ROC(F^m(d_i)) \tag{10}$$

$$\xi_{ij}^{m} = \begin{cases} 1, & F^{m}(d_i) = F^{m}(d_j) \\ 0, & F^{m}(d_i) \neq F^{m}(d_j) \end{cases} \tag{11}$$

In the base clusters, each cluster can be treated as a local region, so the reliability of clusters should be considered, while weighting term ω_{i}^{m} is incorporated to assign weights to clusters via the *ROC* measure. So the *ROC-CA* matrix not only considers the reliability of base clusters but also considers the locally diversity.

The final consistency method adopted in this paper is a hierarchical clustering algorithm combined with *ROC-CA* matrix. Hierarchical agglomerative clustering is a typically used clustering method, which always takes a similarity matrix as input and performs region merging iteratively to achieve a dendrogram, i.e., a hierarchical representation of clusterings. The input is *ROC-CA* matrix, which can denote as: $Q^{(0)} = \{Q_{ij}^{(0)}\}_{N \times N}$, where $Q_{ij}^{(0)} = a_{ij}$. The N original data are treated as the N regions, denoted as: $R^{(0)} = \{R_{1}^{(0)}, \ldots, R_{N}^{(0)}\}$. According to the given initial region and similarity matrix, the region merging iteration begins, and two regions with the highest similarity are selected for merging each time. The region in the tth is denoted as: $R^{(t)} = \{R_{1}^{(t)}, \ldots, R_{|R^{(t)}|}^{(t)}\}$, the similarity matrix is denoted as: $Q^{(t)} = \{Q_{ij}^{(t)}\}_{|R^{(t)}| \times |R^{(t)}|}$, where $Q_{ij}^{(t)} = \frac{1}{|R_{i}^{(t)}| \times |R_{j}^{(t)}|} \sum_{d_k \in R_{i}^{(t)}, d_l \in R_{j}^{(t)}} a_{kl}$.

Algorithm 4: Locally Weighted Consensus Function
Input: *P*: Ensemble of base clusterings; *k*: number of clusters;
Method:

```
C_{i} in P;
compute the ROC measure of the clusters in P;
formulate the ROC-CA matrix;
initialize the set of regions $R^{(0)}$ and the similarity matrix $Q^{(0)}$;
construct the dendrogram iteratively:
    for $t$ = 1, 2, …, $N-1$
    merge the two most similar regions in $R^{(t-1)}$
    obtain the new set of regions $R^{(t)}$
    obtain the new similarity matrix $Q^{(t)}$
    end for
obtain the clustering with $k$ clusters in the dendrogram
```

Output: the consensus clustering p^{*}

4 Conclusion

To overcome the weakness of short text such as its sparsity and a single clustering algorithm may not be adapted to any type of data structure, this paper proposed a clustering ensemble algorithm based on CNN for short text. The short text vectors generated by word2vec are further extracted by CNN to obtain low-dimensional feature vectors. Then combining with Gini coefficient an index to evaluate the reliability of clustering is proposed which is called *ROC*. The *CA* matrix is redefined on the basis of considering the reliability of clustering and the diversity within clustering. The *ROC-CA* matrix proposed in this paper is used as the input of the final hierarchical clustering function, and the final consistency result is obtained. The above contents are the research results so far. The experiment in this paper is in progress, so no specific data charts are shown for the time being. In the future work, detailed experiments will be carried out to obtain sufficient data to support the method in this paper.

References

1. Lin D (1998) An information-theoretic definition of similarity. In: ICML, pp 296–304
2. Ramage D, Heymann P, Manning CD et al (2009) Clustering the tagged web. In: Second ACM international conference on web search and data mining, ACM, pp 54–63
3. Suo H et al (2006) An improved K-means algorithm for document clustering. J Shandong Univ (Natural Science) 43(1):60–64
4. Yang Z et al (2014) Online comment clustering based on an improved semantic distance. J Softw 25(12):2777–2789
5. Zhang Q et al (2016) Short text clustering algorithm combined with context semantic information. Comput Sci 43(s2):443–446
6. Wang T (2011) CA-tree: a hierarchical structure for efficient and scalable coassociation-based cluster ensembles. IEEE TRANS Syst Man Cybern B Cybern 41(3):686–698
7. Yu Z, Li L, Liu J (2015) Adaptive noise immune cluster ensemble using affinity propagation. IEEE Trans Knowl Data Eng 27(12):3176–3189
8. Fred ALN, Jain AK (2005) Combing multiple clusterings using evidence accumulation. IEEE Trans Pattern Anal March Intell 27(6):835–850
9. Huang D, Lai J-H (2015) Combining multiple clusterings via crows agreement estimation and multi-granularity link analysis. Neurocomputing 170:240–250
10. Strehl A, Gosh J (2005) Cluster ensembles–A knowledge reuse framework for combining multiple partitions. J Mach Learn Res 3:583–850
11. Li T, Ding C (2008) Weighted consensus clustering. In: Proceedings of the 2008 SIAM international conference on data mining (SDM), Atlanta, GA, USA, 2008, pp 798–809
12. Yu Z et al (2014) Hybrid clustering solution selection strategy. Pattern Recognit 47 (10):3362–3375
13. Blei DM, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res Achieve 993–1022

A Survey of Moving Objects kNN Query in Road Network Environment

Wei Jiang[1], Guanyu Li[1](✉), Jingmin An[2], Yunhao Sun[1],
Heng Chen[1,3], and Xinying Chen[1,4]

[1] Faculty of Information Science and Technology, Dalian Maritime University, Dalian, China
rabitlee@163.com

[2] Faculty of Computer and Software, Dalian Neusoft University of Information, Dalian, China

[3] School of Software Technology, Dalian University of Foreign Languages, Dalian, China

[4] School of Software Technology, Dalian Jiao Tong University, Dalian, China

Abstract. With the widespread application of the global positioning service (LBS) technology in the road network environment, the *k*-nearest neighbor query problem of moving objects in the road network has become a research hot spot for many scholars. This paper introduces the research status of *k*-nearest neighbor query of moving objects in the road network, analyzes and summarizes the research results of scholars at home and abroad, and finally analyzes the challenges of *k*-nearest neighbor queries for moving objects on the road network in the future.

Keywords: Road network environment · Moving object · *k*-nearest neighbor query

1 Introduction

With the rapid development of mobile communication and spatial positioning technology, the location-based services (LBS) have been promoted. The research of *k*-nearest neighbor query technology for moving objects in road networks has very important social price and commercial value [1, 2]. The *k*-nearest neighbor query is one of the common queries in the location service, which refers to finding the *k* target points of interest (PoI) of the nearest location to the query requester, for example, finding the nearest *k* taxis or gas stations from the query requester.

In practical applications, such as sharing the renter, according to the query results of the *k* satisfying conditions returned by the query requester according to the request, the system determines whether the different query requesters have the same riding route, thereby determining whether to give them a carpool. Aiming at the demand of more and more users, the k-nearest neighbor trajectory query processing method of moving objects in the road network needs to be solved urgently, which has important practical significance.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 629–636, 2020.
https://doi.org/10.1007/978-981-15-0187-6_75

2 Problem Description

The road network can be expressed as an undirected graph with weights $G = (V, E)$, where $V = (v_1, v_2, \ldots, v_n)$ represents the set of nodes in the road network, and E represents the set of edges which connects two nodes. Each edge e can be represented as e (v_i, v_j, w), where v_i and v_j represent the start node and the end node of the edge, respectively, and w is represented as the distance or transit time between the two nodes.

Figure 1 shows the road network of the section of the Dalian city section, and Fig. 2 shows the road network structure of the network section of the area. In Fig. 2, ①, ②…, respectively, represent the corresponding nodes V_1, V_2…. In Fig. 3, △1, △2… respectively, represent the moving objects m_1, m_2… on the road network; in this example, taxis, q_1, q_2, q_3 and q_4 are at the query point, and the query requester sends a taxi request at the place. This type of query is an NN query. At this time, the query result of q_1 is $R = \{m_{11}\}$. If you want to query the nearest three taxis, it is a kNN query. $k = 3$, the query result is $R = \{m_{11}, m_{12}, m_{13}\}$.

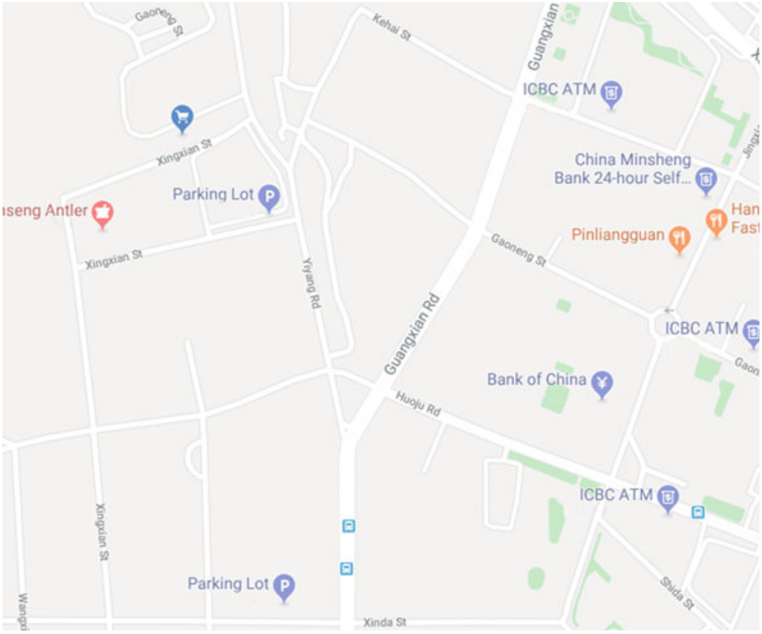


Fig. 1. Regional road network of Dalian

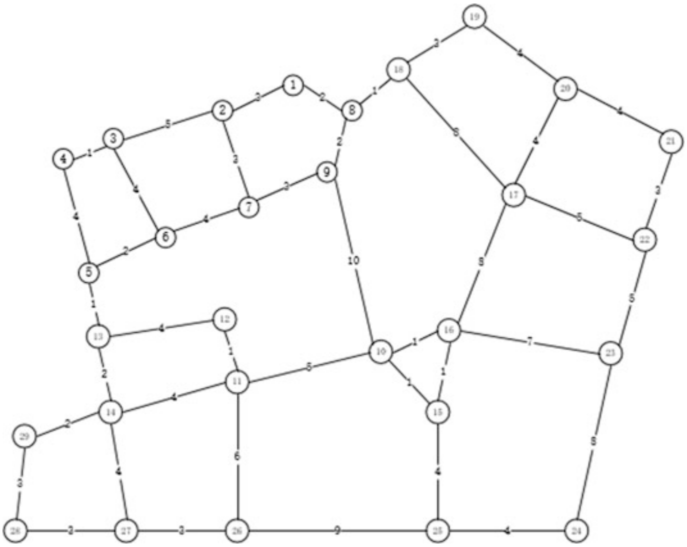


Fig. 2. Road network modeling

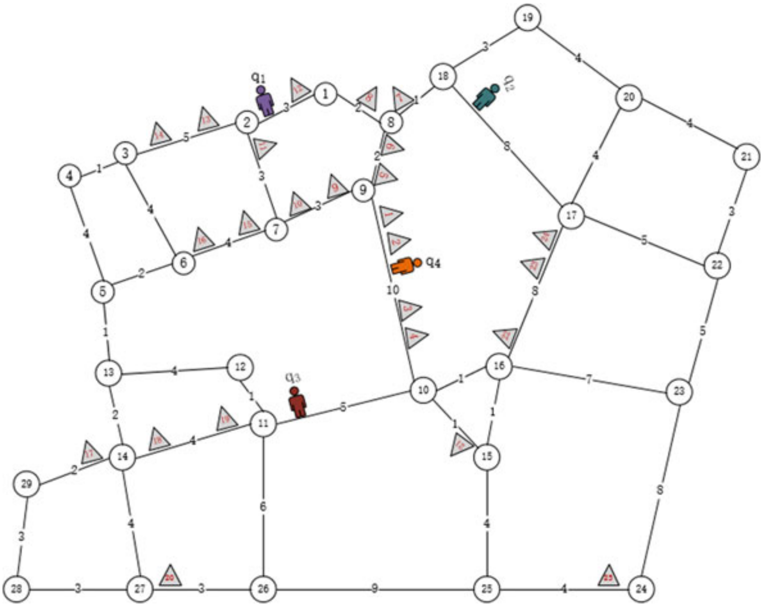


Fig. 3. kNN query

The nearest neighbor query is an important query type based on location service. It was proposed by Knuth in 1973 [3]. After a lot of scholars' unremitting research, it has enriched its theoretical knowledge and also broadened the application field of nearest neighbor query. Neighbor queries play a pivotal role in location-based services. A neighbor query is a data object that returns the closest distance from the query point q in a given data set D. For example, a taxi on a traffic road looks for a gas station closest to your own k. The nearest neighbor query has been developed to date, including nearest neighbor (NN) [4, 5], reverse nearest neighbor (RNN) [6], continuous nearest neighbor (CNN) [7] and aggregate nearest neighbor (ANN) [8].

The environmental characteristics and object characteristics of the mobile network object query determine the two main issues in the query process: (1) road network environment representation; (2) mobile object processing. Road network environment, the existing research, mainly adopts data structure in figure [9] to simulate the actual road network $G = (V, E)$, where V is the point set in the figure to represent the connection between the road segments, and E is the edge set in the figure to represent the road segment. In the section, store some additional information (segment cost, waiting time, etc. [10]).

3 Existing Solutions and Existing Problems

At present, many universities and research institutes have conducted in-depth research on NN queries in the road network environment, the research on NN query technology in the road network environment has obtained many valuable research results, literature in [11], and the nearest neighbor query technology in the road network environment is analyzed and compared. There are some typical nearest neighbor query methods: INE [12], incremental Euclidean restriction (IER) [12], algorithms for processing, as well as Voronoi-based network nearest Neighbor (VN^3) [13] were proposed by Kolahdouzan et al. in 2004, DisBrw [14] was proposed by Samet et al. in 2008, ROAD [15] was proposed by Tian et al. in 2010, calculating the shortest path of each subgraph using the junction points of the two subgraphs. Figure 4 is the ROAD framework structure. G-tree (graph tree) [16] was proposed by Li et al. in 2015.

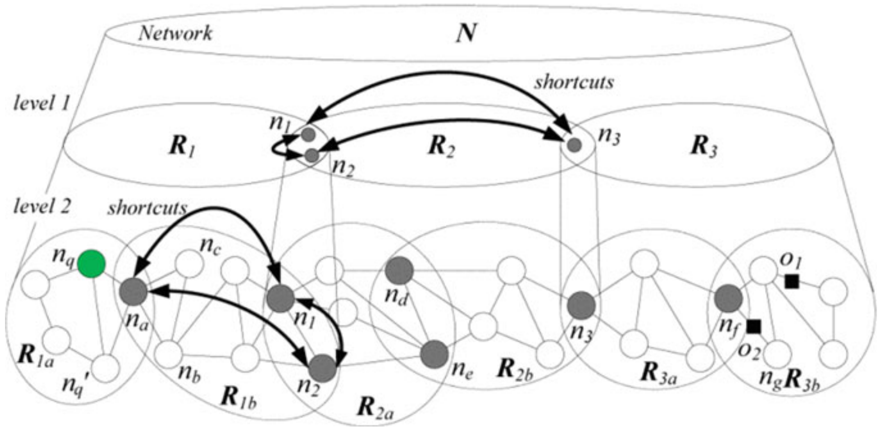


Fig. 4. Road framework structure [15]

The INE algorithm is based on the idea of the Dijkstra algorithm, which accesses the network nodes based on their increasing distance from the source of the query. It takes the distance from the traversed node to the query point into the priority queue, and each time the node with the smallest distance is selected to expand its neighbor. The IER algorithm first finds the nearest query object of *k* Euclidean distance [17] and then uses the shortest path algorithm to calculate the distance of the road network from the query point one by one. The *k* query objects form the initial candidate set and then continue to find the next neighbor according to the Euclidean distance, calculate the distance of the road network from the query point with the shortest path algorithm and then update the candidate set. When the Euclidean distance of the query object is greater than or equal to the current candidate set distance, the calculation is stopped, and the distance is the *K*th largest value. At the end of the search, the *k* query objects in the candidate set are the *k*-nearest neighbors we are looking for. The shortest path algorithm here is optional A^* algorithm [18]. The VN^3 algorithm uses the geometric properties of spatial objects to divide a large road network into multiple small Voronoi regions and pre-calculate the distance between them. The G-tree algorithm proposes an index of a balanced search tree called G-tree (graph tree) [16], which is a graph tree. Its index structure is very similar to the index structure of the R-tree, which is constructed by recursively dividing the road map into smaller subgraphs. Each node in the G tree corresponds to a sub-network. Similarly, it pre-calculates the minimum road distance between some sub-networks. Then searching for the query object in the corresponding sub-networks on the G-tree.

Existing research focuses on kNN search for static objects [19–24] or continuous kNN search and Euclidean distance constraints [25–34]. The former does not support dynamic updates of moving objects because updating indexes is quite expensive. The latter cannot effectively calculate the distance on the road network. Therefore, they cannot effectively solve this problem.

The moving object in the road network is a special kind of moving object. Compared with the moving object in free space, the moving object of the road network has the following characteristics.

a. The motion trajectory of the moving object can only be moved within the road network by the limitation of the road network.
b. The movement process of the moving object is subject to traffic regulations. For example, in certain road sections, it must be moved in accordance with the specified direction, and the control of traffic lights must be observed.
c. Both the query object and the object being queried are located in the road network, so the distance in the query process no longer only represents the Euclidean distance, but the distance of the road network needs to be calculated.

Another important issue is that *k*-nearest neighbors (kNN) search for moving objects with road network constraints, and in the case of moving objects and queries on a given road network, find *k* objects that are closest to the query location. Existing research has focused on kNN searches for static objects or continuous kNN searches with Euclidean distance constraints. The former does not support dynamic updates of moving objects, while the latter does not support road networks. Since objects move dynamically on the road network, there are two main challenges. The first is how to express moving objects on the road network, and the second is how to find the k-nearest moving objects.

Given a set of query points $Q = (q_1, q_2, \ldots, q_n)$ and a set of spatial object points $O(o_1, o_2, \ldots, o_\mathrm{n})$ to be queried, find k objects that are closest to any point in Q. That is, the query returns a set of k pairs of points (k different points o) consisting of point pairs (q_i, o_j), such that for any $(q, o) \in D$, $q' \in Q$, $o' \in O/D$, both have dist. $(q, o) \leq$ dist. (q', o').

For example, in Fig. 1c, given a set of queries $Q = (q_1, q_2, q_3, q_4)$, $K = (k_1, k_2, k_3, k_4)$ neighboring objects from corresponding points in Q are, respectively, queried.

The most intuitive and simple method is to find the road network distance of all the points in Q to all the points in O and then return the pair of points with the smallest distance value of k road networks. However, if there are many object points, then the number of pairs of points will be large, and because of the many nodes on the road network, the cost of calculating the distance between each point and the network is high, so this method is very inefficient.

In fact, in the above simple method, most of the distance calculation between points is useless. The k objects closest to the query point set must belong to the kNN of several points in the query point set. Therefore, we only need to find the kNN of each query point one by one, and we can get the kNN of the query point set. There are many ways to find kNN. The most common ones are Dijkstra-based algorithms and IER-based algorithms.

4 Outlook

(1) Road network dynamic update. Road network data is dynamically changed: First, the nodes, road segments and points of interest in the road network are dynamically changed; secondly, the weight of the road in the road network is dynamically changed; in addition, both the query point and the query object may be dynamically changed. Under the above circumstances, how to adapt to the dynamic update of the road network data and efficiently complete the nearest neighbor query processing is a major challenge for the nearest neighbor query technology in the road network environment.

(2) Consider the user's personalized query. With the wide application of location-based services, users often have a lot of personalized location query requirements, most of which are based on nearest neighbor queries, such as with time constraints or A nearest neighbor query problem with keywords.

(3) Consider the road semantics. With the application of smart sensors, users can combine the road semantic information provided by smart devices to perform fuzzy query. For example, the background detects the road semantics through the smartphone in real time, so that the fuzzy query of the user can obtain more accurate results.

(4) Scalability. Different road networks have different scales and different object distribution densities, and the scale and object distribution density of the same regional road network are also increasing. The scalability of the query method is very important. Currently, recently neighboring technology scalability needs to be further improved.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant (No. 61371090, No. 61602076 and No. 61702072), the China Postdoctoral Science Foundation Funded Project (2017M621122 and 2017M611211), the Natural Science Foundation of Liaoning Province (No. 20170540144, No. 20170540232 and No. 20180540003) and the Fundamental Research Funds for the Central Universities (No. 3132017118, No. 3132017121 and No. 3132017123).

References

1. Zhiming D, Guo L, Li X, et al (2008) Data model and continuous query processing of location dependent data based on object-relational database model. J Comput Res Dev 45 (Suppl.):88–94 (in Chinese)
2. Qiao SJ, Han N, Wang C, Zhu F, Tang CJ (2014) A two-tiered dynamic index structure of moving objects based on constrained networks. Chin J Comput (9)
3. Donald K (1973) The art of computer programming, vol 3. Addison-Wesley Professional, Indianapolis
4. Roussopoulos N, Kelley S, Vincent F (1995) Nearest neighbor queries. In: ACM sigmod record, 24(2):71–79. https://doi.org/10.1145/223784.223794
5. Zhu H, Yang X, Wang B, et al (2016) Range-based obstructed nearest neighbor queries. In: Proceedings of the SIGMOD, pp 2053–2068. https://doi.org/10.1145/2882903.2915234
6. Zeberga K, Cho HJ, Chung TS (2014) A safe-region approach to k-RNN queries in directed road network. In: Proceedings of the IEEE international conference on computational science and engineering (CSE), pp 818–824
7. Cho HJ, Chung CW (2005) An efficient and scalable approach to CNN queries in a road network. In: Proceedings of the VLDB, pp 865–876
8. Yiu M, Mamoulis N, Papadias D (2005) Aggregate nearest neighbor queries in road networks. TKDE 17(6): 820–833
9. Almeida VTD (2004) Indexing the trajectories of moving objects in networks. In: Proceedings of the IEEE scientific and statistical database management international conference, pp 115–115. https://doi.org/10.1109/ssdm.2004.1311200
10. Feng J, Zhu YL, Mukai N, Watanabe T (2007) Search on transportation networks for location-based service. Appl Intell 26(1):69–79. https://doi.org/10.1007/s10489-006-0004-4
11. Tao Y, Papadias D, Lian X (2004) Reverse kNN search in arbitrary dimensionality. In: Processing of the 30th VLDB Conference, Canada, Toronto, pp 744–755
12. Papadias D, Zhang J, Mamoulis N, Tao Y (2003) Query processing in spatial network databases. VLDB 2003:802–813
13. Kolahdouzan M, Shahabi C (2004) Voronoi-based k nearest neighbor search for spatial network databases. In: VLDB'04 Proceedings of the thirtieth international conference on very large data bases, pp 840–851

14. Samet H, Sankaranarayanan J, Alborzi H (2008) Scalable network distance browsing in spatial databases. In: Proceedings of the SIGMOD, pp 43–54. https://doi.org/10.1145/1376616.1376623
15. Lee KCK, Lee WC, Zheng B, Tian Y (2010) Road: a new spatial object search framework for road networks. IEEE Trans Knowl Data Eng 24(3):547–560. https://doi.org/10.1109/tkde.2010.243
16. Zhong RC, Li GL, Tan KL, Zhou LZ (2013) G-tree: an efficient index for KNN search on road networks. In: Proceedings of the CIKM, pp 39–48. https://doi.org/10.1145/2505515.2505749
17. Hjaltason GR, Samet H (1999) Distance browsing in spatial databases. TODS 24(2):265–318
18. Deng K, Zhou X, Shen HT, Sadiq SW, Li X (2009) Instance optimal query processing in spatial networks. VLDB 18(3):675–693
19. Leclerc L, Becarie C (2012) Meso lighthill-whitham and Richards's model designed for network applications. Transp Res Part B Methodol 15, 25
20. Lee KCK, Lee W, Zheng B (2009) Fast object search on road networks. In: EDBT, pp 1018–1029
21. Cheema MA, Lin X, Zhang W (2010) Probabilistic reverse nearest neighbor queries on uncertain data. IEEE Trans Knowl Data Eng 22(4):1–14
22. Huang W, Li G, Tan K, Feng J (2012) Efficient safe-region construction for moving top-k spatial keyword queries. In: CIKM, pp 932–941
23. Zhong R, Li G, Tan KL, Zhou L, Gong Z (2015) G-tree: an efficient and scalable index for spatial search on road networks. TKDE 27(8):2175–2189
24. Saltenis SV, Jensen CS, Leutenegger ST, Lopez M (2000) A indexing the positions of continuously moving objects. SIGMOD 29(2):331–342
25. Jensen CS, Lin D, Ooi BC (2004) Query and update efficient b$^+$-tree based indexing of moving objects. In: Proceedings of the Thirtieth international conference on Very large data bases, VLDB endowment, 2004, pp 768–779
26. Pfoser D, Jensen CS, Theodoridis Y et al (2000) Novel approaches to the indexing of moving object trajectories. In: Proceedings of VLDB, pp 395–406
27. Wang H, Zimmermann R (2008) Snapshot location-based query processing on moving objects in road networks. In: Proceedings of the 16th ACM SIGSPATIAL international conference on advances in geographic information systems, ACM, p. 50
28. Cho HJ, Jin R (2016) Efficient processing of moving k-range nearest neighbor queries in directed and dynamic spatial networks. Mob Inf Syst
29. Hu H, Xu J, Lee DL (2005) A generic framework for monitoring continuous spatial queries over moving objects. In: *Proceedings of the 2005 ACM SIGMOD international conference on management of data*, ACM, pp 479–490
30. Cho HJ, Jin R, Chung TS (2015) A collaborative approach to moving k-nearest neighbor queries in directed and dynamic road networks. Pervasive Mob Comput 17(Part A):139–156
31. Li G, Fan P, Li Y, Du J (2010) An efficient technique for continuous k-nearest neighbor query processing on moving objects in a road network. In: 2010 10th IEEE International Conference on Computer and Information Technology, pp 627–634
32. Wang H, Zimmermann R (2011) Processing of continuous location-based range queries on moving objects in road networks. TKDE 23(7):1065–1078
33. Yu Z, Liu Y, Yu X, Pu KQ (2015) Scalable distributed processing of k nearest neighbor queries over moving objects. TKDE 7(5):1383–1396
34. Guohui L, Yanhong L, Jianjun L, Shu L, Fumin Y (2010) Continuous reverse k nearest neighbor monitoring on moving objects in road networks. Inf Syst 35(8):860–883

The Importance of Researching and Developing the Semantic Web of Things

Xinying Chen[1,2], Guanyu Li[1](✉), Yunhao Sun[1], Heng Chen[1,3], and Wei Jiang[1]

[1] DaLian Maritime University, DaLian, LiaoNing 116026, China
rabitlee@163.com
[2] Dalian JiaoTong University, Dalian, Liaoning 116028, China
[3] DaLian University of Foreign Languages, DaLian, LiaoNing 116044, China

Abstract. As an unprecedented technological innovation, the Internet of Things has its inherent contradictions. The Semantic Web of Things is the solution to the internal contradiction of the Internet of Things. As an Intelligent Internet of Things, the Semantic Web of Things transforms grammar matching into semantic matching, which enhances the essence of the Internet of Things. This paper introduces the background and research significance of the Semantic Web of Things and introduces the branch of the current Intelligent Internet of Things. The related definition is given and the semantic relational structured network model of the Semantic Web of Things is proposed. This paper provides a theoretical basis and research basis for further research on Semantic Web of Things.

Keywords: Internet of Things · Intelligent Internet of Things · Semantic Web of Things

1 The Inherent Contradiction of the Internet of Things and Its Countermeasures

In 1999, the Internet of Things was first proposed by Ashton. As an unprecedented technological innovation, the Internet of Things provides us with ubiquitous connectivity, computing, and communication [1]. In the past ten years, in order to promote the development of the Internet of Things and transform it from an academic concept to a practical application, academic research teams, service providers, and network operators around the world have been making unremitting efforts. Among them, the focus of attention is mostly on how to realize the connection, communication, and calculation of the Internet of Things, that is, how to obtain information from the environment and share information [1].

The Internet of Things should not only have limbs with low levels of perception such as sight, smell, hearing, and touch, but also high-level intelligent services that can be understood, thought, and learned. This requires expanding

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 637–644, 2020.
https://doi.org/10.1007/978-981-15-0187-6_76

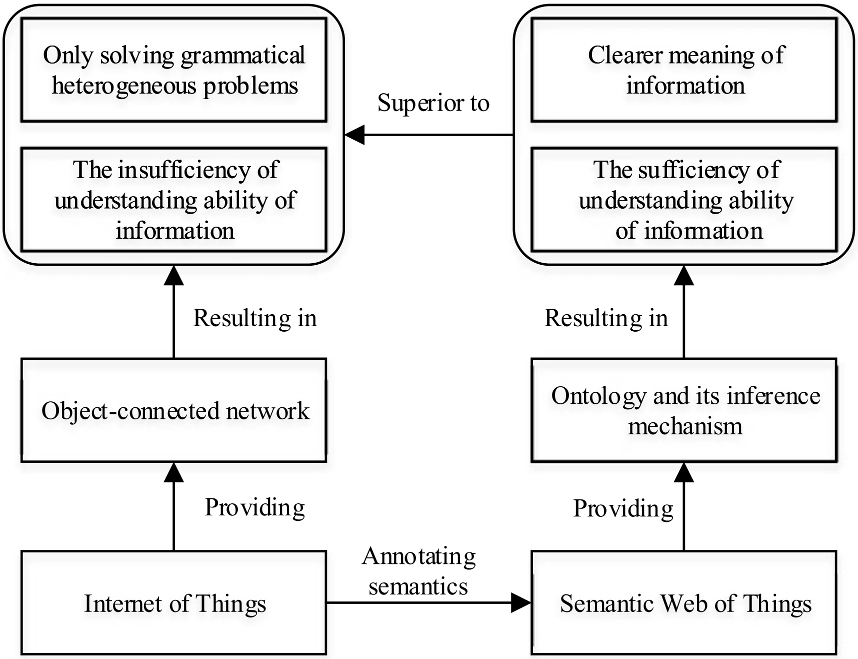

Only solving grammatical heterogeneous problems
The insufficiency of understanding ability of information
Superior to
Clearer meaning of information
The sufficiency of understanding ability of information
Resulting in
Resulting in
Object-connected network
Ontology and its inference mechanism
Providing
Providing
Internet of Things
Annotating semantics
Semantic Web of Things

Fig. 1. A causal path diagram of SWoT

the Internet of Things into a smart network and an intelligent open network that supports smart sharing and intelligent services.

In recent years, many scholars have explored how to attach semantics to the Internet of Things to resolve their inherent contradictions. The introduction of semantic annotation and ontology can transform grammar matching (i.e., keyword matching) into semantic matching (i.e., meaning matching), which can improve the ability of the using subject to understand and further reason to obtain related information, so as to enhance the function of the Internet of Things. This paper refers to this Intelligent Internet of Things, which is improved in essence, as the Semantic Web of Things (SWoT). Figure 1 illustrates interrelationships between the Internet of Things and the Semantic Web of Things and their respective characteristics and reveals the causal logic of the Semantic Web of Things.

2 Research Trends of Intelligent Internet of Things

It is estimated that by 2020, the number of all networked devices, including networked computers and connected mobile phones, will reach 500–100 billion [2]. And the global sensor market is expected to increase to 123.5 billion. In the "Internet of Things in 2020" analysis report, the European research institute EPOSS pointed out that the development of the Internet of Things will go through four phases: the closed-loop industrial application phase before 2010; the object-interconnection phase in 2010–2015; the semi-intelligent phase in 2015–2020; and the fully intelligent phase after 2020, that is, the Intelligent Internet of Things phase. Searching for “Semantic Web of Things” by “Google Scholar” with “full sentence” and “full text” gets results: by 2018, there are more than 400 results, including about 282 results from 2010 to 2015. Searching for “Cognitive Internet of Things” by “Google Scholar” with “full sentence” and “full text” gets results: By 2018, there are nearly 200 results, including about 90 results from 2010 to 2015. Searching for “Wisdom Web of Things” by “Google Scholar” with “full sentence” and “full text” gets results: There are about 200 results were included in 2018, including about 81 results from 2010 to 2015.

This shows that the international academic research on “Intelligent Internet of Things” is on the rise. It can be seen that although the “Intelligent Internet of Things” is still a developing concept and the related technologies of the Internet of Things are still at the initial stage of development, but linking everything into the Internet is an inevitable trend of information development. The emergence of “Intelligent Internet of Things” is inevitable, and it will have a profound impact on economic development and social life.

3 Various Branches of the Intelligent Internet of Things

Unlike the Internet of Things, which focuses on connectivity and communication, with the development of IoT-related disciplines such as the information field, researchers in various fields are based on different “smart” thinking starting points, respectively, study the “Intelligent Internet of Things” from different perspectives. At present, compared with the traditional Internet of Things, the research on Intelligent Internet of Things is mainly based on foreign literature, mainly including the following three branches.

(1) Cognitive branch
The term cognitive refers to the process of information extraction and information processing of external things acting on human sensory organs through activities such as perception, feeling, representation, memory, thinking, imagination, and speech [3]. Wu et al. [4] discussed a Cognitive Internet of Things, referred to the definition of Cognitive Internet of Things based on the operational mechanism of human brain proposed by Haykin [3], proposed a cognitive operation framework, and then proposed several basic cognitive tasks that may be involved in the development and research of the Cognitive Internet of Things. Foteinos Vassilis and Vlacheas Panagiotis et al. proposed a cognitive management framework based on the Internet of Things to better support the sustainable development of smart cities [5].

(2) Semantic branch
In the study of Semantic Web of Things, semantics refers to the introduction of semantic annotation and ontology. Kotis et al. [6] pointed out that in order to achieve seamless integration of information in the Internet of Things, Semantic Web technology needs to be integrated into the Internet of Things. They

proposed the ontology merging method in the Internet of Things environment. Ali et al. [7] pointed out that the massiveness, heterogeneity, and dynamics of data information on the Internet of Things are the main factors hindering interoperability of the Internet of Things. Jara et al. [8] proposed the Semantic Web of Things based on the Semantic Web and the Internet of Things to realize the semantic interoperability of heterogeneous resources, devices, objects, and systems in the Internet of Things and analyzed the integration of heterogeneous devices, device abstraction, and semantic inter-operability issues at different levels in a bottom-up manner. Zhuge [9] believes that through the network can create a global ecological environment management service system, proposed a multi-dimensional complex space Cyber-Physical Society, and proposed a Semantic Space Model [10].

(3) Wisdom branch
Zhong [11] believes that wisdom refers to that all things in the Internet of Things are self-aware and mutually sensible to provide the right services for the right people at the right time and context, the foundation of the Wisdom Web of Things is data. He proposed the data loop "object-data-information-knowledge-wisdom-service-human-object", data circulation system, and data conversion mechanism. Eguchi et al. [12] explore and exemplify how to integrate WI, BI, UI, and CI technologies and effectively use technology to form the data cycle. Gao et al. [13] believe that as a next-generation network, Wisdom Web of Things should be able to provide ubiquitous smart services in this ubiquitous network. In order to realize it, they proposed the adaptive definition and adaptive support framework of the Wisdom Web of Things.

(4) Summary and comparison of the three branches
The summary and comparison of the branches of the Intelligent Internet of Things are shown in Table 1. It can be seen from the above analysis that although the starting point and the form of expression are different, the research hot spots of the three branches focus on: open (ubiquitous), interoperability (collaboration), adaptive (self-management), and intelligent reasoning.

Table 1. Branches of Intelligent Internet of Things

Branch	Research results	Characteristics
Cognitive	[3–5]	This branch can realize the intelligent service of the Internet of Things through cognitive, self-management, and cooperation mechanisms
Semantic	[6–10]	This branch can solve heterogeneous information interoperability issues on the Internet of Things based on Semantic Web technologies
Wisdom	[11–13]	The branch can provide intelligent services that adapt to the user's dynamic needs, changing data, dynamic computing resources, and security requirements

4 Semantic Web of Things—Intelligent Reasoning and Smart Service Networking

The Internet of Things is the first step toward a ubiquitous network, so the Intelligent Internet of Things should first be a ubiquitous network. And the release and use of the same information requires the support of the semantic context of the relevant knowledge in order to facilitate the correct and comprehensive understanding of the information. For example, the intelligent cognition of Cognitive Internet of Things requires semantic driving as the basis [9], and the top 10 problems that need to be solved in the network intelligence (WI) mentioned in the Wisdom Web of Things include semantic processing problems [13].

Intelligent Internet of Things should be dynamic and adaptive. ZhongNing et al. proposed that the wisdom of the Internet of Things should be reflected in the timely provision of appropriate services for an object in the corresponding context. In other words, the individual objects in the Intelligent Internet of Things must have the ability to perceive and adapt to their own and surrounding dynamic environments and have the ability to adapt to their own dynamic changes and dynamic changes in their surrounding environment.

The main task of the Intelligent Internet of Things is to provide extensive semantic interoperability, as well as the sharing and reuse of various resources and services. It needs to be built in the mode of pervasive computing and also needs to be built in the environment of multi-agent systems. At the bottom level, semantic interoperability is to achieve information exchange and information sharing at the semantic level. At the top level, the ultimate goal is to achieve interoperability of services or service collaboration. Therefore, the semantic collaboration and semantic inter-operability of smart services should firstly realize the synergy and interoperability of sensing data information, in order to provide users with the required services in real time and dynamically in the Intelligent Internet of Things environment. Therefore, for the sensory data from aware devices, the preprocessing and data analysis operations must be performed first; then, the obtained information should be represented and constructed in a semantic form, and the related information should be combined and processed to finally realize the selection and management of data-as-a-service, information-as-a-service, and knowledge-as-a-service. Then, the "data information chain" of the Intelligent Internet of Things should be composed of data, information, knowledge, and services. Since the source data object composed of the perceptual data is at the bottom of the data chain, the adaptability of sensing as a service is the basis of the adaptability of the Internet of Things. Perception information as a service is the foundation of everything as a service.

Based on the above analysis, it can be seen that both cognitive networks and intelligent networks are inseparable from the support of semantic technology, and both can be regarded as a feasible solution for semantic networks, so this paper will use the Semantic Web of Things as the synonymous with the Intelligent Internet of Things. It is believed that the Semantic Web of Things should be a ubiquitous and intelligent service system, should be based on pervasive computing and multi-agent, and should have the ability of intelligent reasoning,

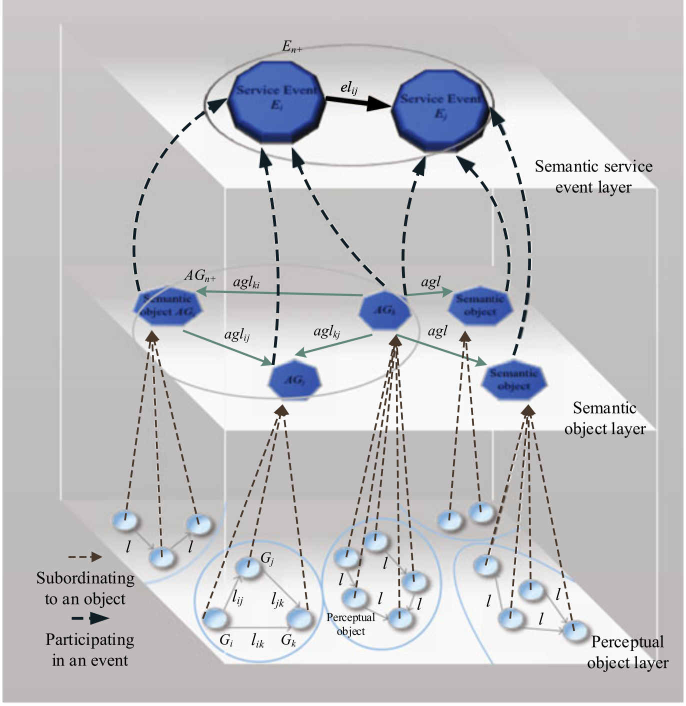


Fig. 2. Semantic relationship structured network model

adaptive, should realize object coordination and provide wisdom services based on sensing as a service and semantic collaboration.

Based on the existing literatures and the above analysis, and the semantic chain network model [9–11], etc., design an ontology-based multi-level semantic relationship structured network, as shown in Fig. 2. In Fig. 2, nodes represent object nodes at each level, and directed arcs between nodes represent semantic associations between objects, and semantic associations are used to complete reasoning and prediction. The semantic relationship structured network model is divided into three layers: the perceptual element layer, the semantic object layer, and the semantic service event layer. The model relies on perceptual elements to obtain data, uses intelligent objects to aggregate data information, interprets information based on events, uses event class libraries (ontology) to store and reason event information and knowledge, and provides intelligent services based on data, information, and knowledge. The three layers of the model all adopt an extended semantic link network model. Subsequent research will explore and analyze the semantic model and semantic inference rules of the perceptual element layer, the semantic object layer, and the semantic service event layer in detail.

5 Conclusion

Starting from the inherent contradiction of the Internet of Things, this paper considers that Semantic Web of Things is the solution to the inherent contradiction of the Internet of Things. Ontology-based semantic support and semantic reasoning are the key to realize the Semantic Web of Things and also the key to solving the contradiction of the Internet of Things. This paper discusses the research background and significance of the Intelligent Internet of Things through data analysis and literature citation, introduces the branches of the current Intelligent Internet of Things, gives the definition, and proposes the semantic relationship structured network model.

In the follow-up, will further analyze and study the existing semantic information organization model and semantic information system model, strive to implement a formal analysis and formal representation of the semantic relationship structured network supported by the ontology from the perspective of abstraction and granulation, and solve the semantic linking and reasoning between semantic objects based on ontology support.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grant (No. 61371090, No. 61602076 and No. 61702072). The China Postdoctoral Science Foundation Funded Project (2017M621122 and 2017M611211). The Natural Science Foundation of LiaoNing Province (No. 20170540144, No. 20170540 232 and No. 20180540003) and the Fundamental Research Funds for the Central Universities (No. 3132017118, No. 3132017121 and No. 3132017123).

References

1. Aggarwal CC, Ashish N, Sheth A (2013) The internet of things: a survey from the data-centric perspective. In: Managing and mining sensor data. Springer US, pp 383–428
2. Vilmos A, Medaglia C, Moroni A (2010) Vision and challenges for realising the internet of things. Hot Working Technol 35(2):59–60
3. Haykin S (2012) Cognitive dynamic systems: perception-action cycle, radar and radio. Cambridge University Press, London, UK
4. Wu Q, Ding G, Xu Y et al (2014) Cognitive internet of things: a new paradigm beyond connection. IEEE Internet Things J 1(2):129–143
5. Foteinos V, Kelaidonis D, Poulios G et al (2013) A cognitive management framework for empowering the internet of things. Future Internet 51(6):187–199
6. Kotis K, Katasonov A (2012) Semantic interoperability on the web of things: the semantic smart gateway framework. In: Sixth international conference on complex, intelligent and software intensive systems. IEEE, pp 630–635
7. Ali MI, Ono N, Kaysar M et al (2015) A semantic processing framework for IoT-enabled communication systems. In: International semantic web conference, pp 241–258
8. Jara AJ, Olivieri AC, Bocchi Y et al (2014) Semantic web of things: an analysis of the application semantics for the IoT moving towards the IoT convergence. Int J Web Grid Serv 10(2):244–272

9. Zhuge H (2011) Semantic linking through spaces for cyber-physical-socio intelligence: a methodology. Artif Intell 175(5):988–1019
10. Zhuge H (2011) The complex semantic space model. In: 20th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises (WETICE). IEEE, pp 9–15
11. Zhong N, Ma JH, Huang RH et al (2013) Research challenges and perspectives on wisdom web of things (W2T). J Supercomput 64(3):862–882
12. Eguchi A, Nguyen H, Thompson CW (2013) Everything is alive: towards the future wisdom web of things. World Wide Web 16(4):357–378
13. Gao Y, Lin M, Wang R (2013) Adaptive support framework for wisdom web of things. World Wide Web 16(4):379–398

Facial Expression Recognition Based on Strengthened Deep Belief Network with Eye Movements Information

Bo Lu(✉) and Xiaodong Duan

Dalian Key Lab of Digital Technology for National Culture,
College of Computer Science and Engineering, Dalian MinZu University,
Dalian, Liaoning 116600, China
lubo@dlnu.edu.cn

Abstract. Facial expression recognition is an important application in computer vision. Generally, features which are used for facial expression recognition are mostly based on geometric and appearance features of image. This paper presents a novel method to identify facial expressions which exploring eye movements data labels as auxiliary labels to construct classifier, a Strengthened Deep Belief Network (SDBN) in a united cycle framework is constructed. This framework is formed as strong classifier by multi-weak classifiers voted. Experiments on Cohn-Kanade database showed that the proposed method achieved a better improvement in the task of facial expression recognition.

Keywords: Facial expression recognition · Eye movement information · Deep belief network

1 Introduction

Facial expression is a variety of emotional states through the eye muscles, facial muscles, and mouth muscles. It is the most powerful, natural, and direct way to emotional communication in daily social interaction [1]. In the past few decades, automated facial expression analysis has become a popular and challenging topic in computer vision and pattern recognition fields, such as security, human–computer interaction, and multimedia [2,3]. Though this process has been made much progress, high accuracy in facial expression recognition is still very different due to the complexity, variability, and subtlety in facial expressions.

Recently, some studies use eye-tracking technology to study the facial expression recognition. They use eye-tracking device obtained eye movements data and then use eye movements data for further analysis. Facial features have been found in some regions, including eyes, eyebrows, mouth, and nose, these areas provide key clues in the recognition of facial expressions [4], and this is consistent with psychological findings [5]. Researchers divided these areas into Region Of Interest

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 645–652, 2020.
https://doi.org/10.1007/978-981-15-0187-6_77

(ROI) in order to better recognize facial expressions. Since studies have shown that facial expression recognition features mainly concentrated in a few large pieces, determining which areas of the face focused the most important features can help us to understand the implicit mechanism of non-verbal communication.

In this paper, we propose a united cycle Strengthened Deep Belief Network (SDBN) framework, which uses eye movements data as auxiliary label to improve the accuracy of recognition of facial expression. The contributions of this study are (1) use eye tracking data as an expression labels to study facial expression recognition; (2) fusion eye movements data labels and facial expression labels in a unite feature learning, feature selection, and classifier construction framework; (3) the initial input is based on the patch of images, which is very suitable for the analysis of facial expressions, rather than the traditional use of the whole face region as the input, the framework is made of a plurality of DBNs by voting formed a strong classifier. Experiments on the extended Cohn-Kanade (CK+) [6] database show that the proposed method achieved a better performance in facial expression recognition. Besides, as the iteration goes, some important features were selected, the final strong classifier just uses a small number of features, and it also proved that using SDBN framework can improve the characteristics of a learning process.

2 Methodology

In this section, we firstly introduce the overview of the proposed framework. Secondly, eye movements data labels preprocessing is presented. Furthermore, we introduce the top-down fine-tuning process.

2.1 Brief in SDBN Framework

As shown in Fig. 1, the framework contains a group of DBNs structures. Given the training set in the same location of image patches, each DBN learns a hierarchy feature representation. These DBNs are joined together by a strengthened classifier. The proposed SDBN framework includes two major learning processes: the feature representations is firstly learned based on the bottom-up unsupervised feature learning process. Then, the eye movement data labels and the expression labels are fused together, which is explored to fine-tune the weights based on the top-down supervised strengthen process.

2.2 The Construction of the SDBN Framework

As shown in Fig. 1, each DBN consists of one visual layer (the lowest layer) and five hidden layers, which is similar as traditional DBN in [7]. The input of the lowest layer is the same patch location of training data.

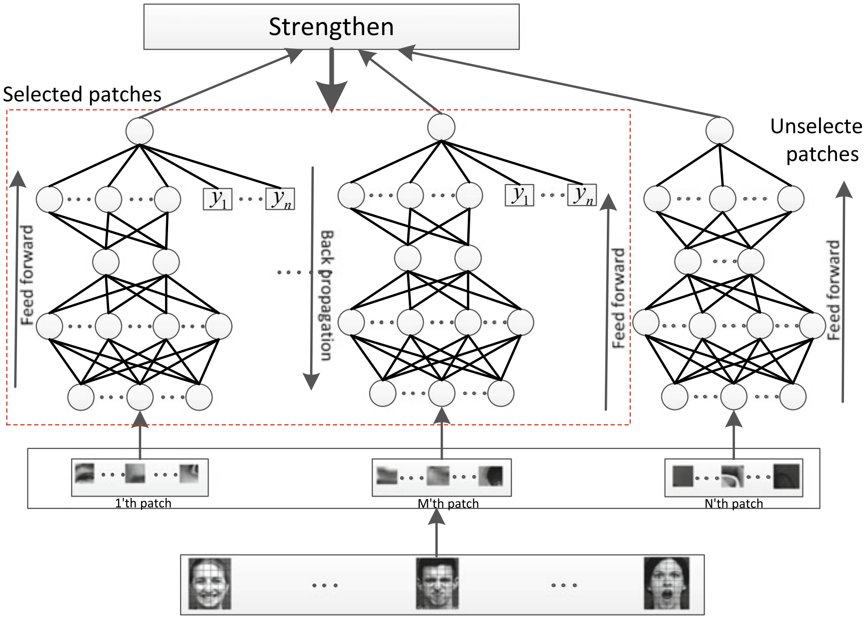


Fig. 1. The proposed SDBN framework. This framework contains multiple DBNs. Each DBN has multiple layers, layer by layer through learning feature representation at a specific location of image patches, only the DBNs surrounded in the red rectangular are selected by strengthening will be fine-tuning jointly

Each layer of DBN has conditional dependences, there is a connection between the layers, but inter-cell layer connection does not exist, except for the top two layers. For $L+1$ layer DBN, the relationship between visual layer and the upper L hidden layer is given as follows:

$$P(\mathbf{h}^0, \mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^L) = P(\mathbf{h}^0|\mathbf{h}^1)P(\mathbf{h}^1|\mathbf{h}^2)...P(\mathbf{h}^{L-2}|\mathbf{h}^{L-1})P(\mathbf{h}^{L-1}|\mathbf{h}^L) \tag{1}$$

$\mathbf{h}^0$ is the visual layer, and $\mathbf{h}^0, \mathbf{h}^1, \mathbf{h}^2, ..., \mathbf{h}^L$ are hidden layers. The value of the hidden nodes can be calculated as Eq. (2) by the known visual nodes.

$$P(\mathbf{H}^{l+1}) = \frac{1}{1 + \exp(-\mathbf{b}_h^{l+1} - (\mathbf{W}^{l,l+1})^T \mathbf{H}^l)} \tag{2}$$

The highest layer of DBN which can be calculated as

$$\mathbf{H}^L = \mathbf{W}^{L-1,L}\mathbf{H}^{L-1} \tag{3}$$

where $\mathbf{W}^{L-1,L}$ represents the weight matrix of the highest two layers. The initial parameters of $\mathbf{b}_v^l$, $\mathbf{b}_h^{l+1}$,$\mathbf{W}^{l,l+1}$ can be computed using an unsupervised bottom-up learning strategies [9].

2.3 Eye Movements Data Labels

The eye movements data is collected by Tobii eye tracking. We divided the facial image into nine areas of interest (AOI) in order to better analysis of regional data, and these AOIs covered most of AUs in Facial Action Coding System (FACS) [8]. Eye movements data is associated with AUs; AUs are related with facial expressions, so we make an assumption that using eye movements data as a label might enhance the facial expression recognition rate.

We defined f_i as the duration of AOI_j on i, i is the sequence number of the participants' fixation, j is AOI number, w_1 denotes the order to distinguish the importance factor, w_1 value is in [1, 2], the first fixation factor value is 2, after fixation factor in turn reduce the i/p, p is the number of all the fixation point of participant, and so according to the order of fixation, we can get each AOI fixation time for T_1.

$$T_1 = f_i\,(2 - i/p) \tag{4}$$

The AOI's weight is $(T_1 + T_2)/\text{total}$, and total is the sum of all the fixation and regression time of AOI. By (1) and (2) using the literature [8], obtain each AOI's Weight_j.

$$\text{Weight}_j = \frac{\sum_{i=0}^{i=p} f_i\,(2 - i/p)\,(1 + c_i/20)}{\text{total}} \tag{5}$$

Weight_j is the weight in AOI_j which ranges [0, 1]; Weights' value is closer to 1, indicating that the AOI is more important in facial expression recognition.

2.4 Top-Down Supervised Fine-Tuning Process

The fine-tuning process in a top-down way by first updating weight matrix $\mathbf{W}^{L-1,L}$. For a set of M_I samples of the image set, the overall prediction error is

$$\varepsilon_{\text{overall}} = \sum_{i=1}^{M_I} \alpha_i \left[\frac{1}{1 + \exp(-\sum_{j=1}^{N} \beta_j \text{sgn}(\mathbf{W}_j^{L-1,L}\mathbf{H}_{i,j}^{L-1} - R_j))} - \mathbf{T}_i \right]^2 \tag{6}$$

where $\mathbf{T}_i$ is the fusion label of ith image, β and R are the N selected weak classifiers of the Adaboost classifier of weights and thresholds, $\mathbf{W}_j^{L-1,L}\mathbf{H}_{i,j}^{L-1} = \mathbf{H}_{i,j}^{L}$ is the DBN output of jth selected image patch in ith image. α_i is an adjustment factor of positive and negative samples, as the number of negative samples is much more than the number of positive samples. sgn(·) is a symbolic function as:

$$\text{sgn}(\mathbf{W}_j^{L-1,L}\mathbf{H}_{i,j}^{L-1} - R_j) = \begin{cases} 1 & \text{when } \mathbf{W}_j^{L-1,L}\mathbf{H}_{i,j}^{L-1} \geq R_j \\ -1 & \text{otherwise} \end{cases} \tag{7}$$

3 Experiments

Experiments are conducted on the extended Cohn-Kanade (CK+) database, which is widely used to assess facial recognition systems.

Eye movements data is obtained by Tobii X2-30 eye tracker, forty college students participated in the experiment, they sat in front of the computer about 60 cm, and each image was showed 10 s. All the selected images scaled the dataset to 167 $*$ 137 pixels. Then, we divided the image into patches of the size 24 $*$ 24. For each DBN, the lowest visual node is 576 (24 $*$ 24), which consists of the number of the LBP features, the hidden layer nodes from the lowest to highest are 500, 500, 1000, 1000, and 1, respectively.

In our experiment, through iterations, the patches were selected by the final strong classifiers which are marketed by the black boxes in Fig. 2, we show only the selected more frequency (the feature was selected more than 5 times in all 8 runs) patches in 8 runs. We can find out most of the patches that were selected concentrated in the eyes, eyebrows, mouth, and the nose, which is consistent with psychological research [8].

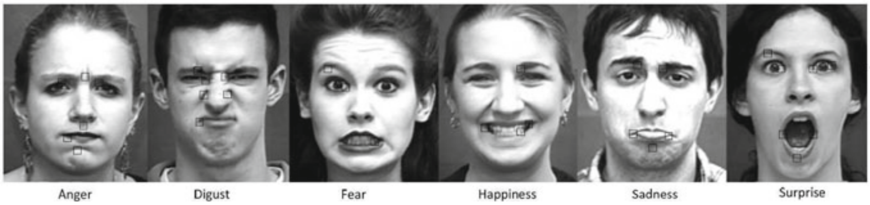


Fig. 2. Recognition accuracy with eight times run of two SDBN modes

We evaluated the relationship between the common patches and the recognition performance. Figure 3 shows the expression recognition with different number of patches. We find that the recognition rate increased rapidly in the first few patches, and when it reaches around the number of 80, almost all have reached the highest recognition rate. When too many patches are divided, the recognition performance might be slightly lower or become fluctuates.

Then, we executed a bottom-up feature learning. We used both bottom-up and top-down feature learning in each DBN (with the fine-tuning in SDBN). Figure 4 shows the recognition accuracy with eight times run of two SDBN modes. Table 1 shows the average performance of these two experimental results.

In order to verify that the use of eye movements data labels and expression labels information may get better results, we use only eye movements data labels and expression labels to do the experiment, respectively. Figure 5 shows the average recognition rate of six expressions of the three different labels.

As shown in Fig. 5, if we only use the expression labels, fear and sadness are better; if we only use eye movements data labels, anger and disgust expression recognition rates are better. Specially, the recognition rates of both happy and

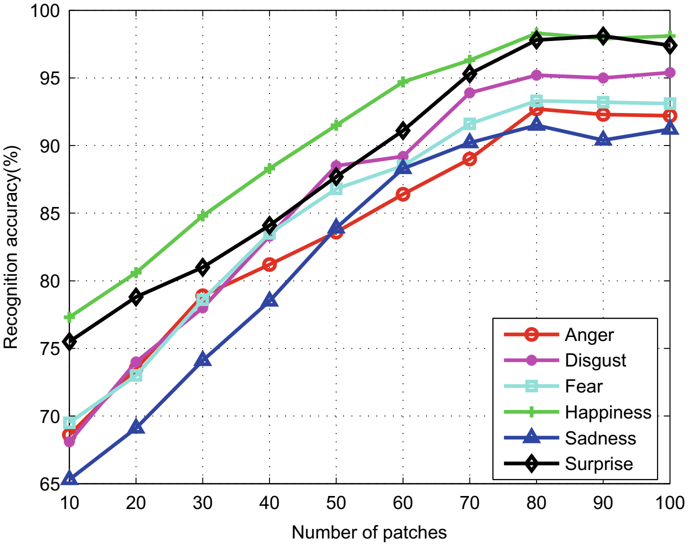


Fig. 3. Expression recognition with different number of patches

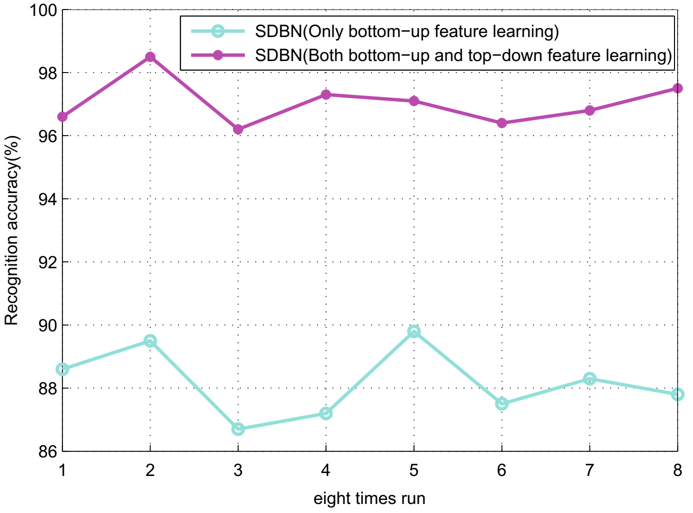


Fig. 4. Recognition accuracy with eight times run of two SDBN mode

Table 1. Average recognition rates (%) of methods

Methods	Accuracy
SDBN (Only bottom-up feature learning)	88.2
SDBN (Both bottom-up and top-down feature learning)	97.1

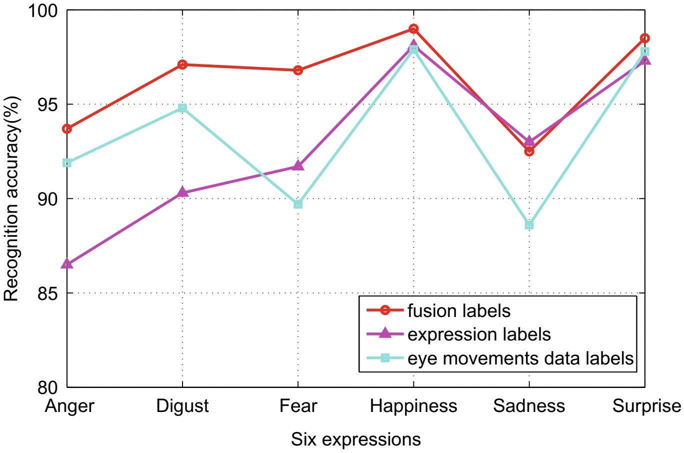


Fig. 5. Average recognition rate of six expressions of the three different labels

surprise expression are almost the same. The recognition rates of all expressions is significantly improved by exploring the fusion labels, but except the sad expression. However, the experiment still proves that the eye movements data labels can facilitate to improve the recognition rates of expression, which is consistent with our hypothesis.

4 Conclusion

In this paper, we use the SDBN framework for facial expression recognition with eye movements information, different from the existing method, and we add the eye movements data labels in supervised process. Experiment proved using eye movements information labels and the use of facial expression labels obtained facial expression recognition rate are almost the same, which explained use eye movements information and facial expression information to identify facial expression exist differences, there are also exist commonalities mode. At the same time, it also proved the fusion of eye movements data labels and expression labels outperform than only using expression labels or eye movements data labels. Only using expression labels, the recognition of sad and fear expression is better, only using eye movements data labels the recognition of anger and disgust expression is better. For happy and surprise expression, the recognition rate of the three methods is almost the same. Meanwhile, we proved that using eye movements information as an auxiliary label can improve the recognition rate of expression.

Acknowledgements. This work was supported by Natural Science Foundation of China (No.61602085, No.61370146).

References

1. Buck R (1994) Social and emotional functions in facial expression and communication: the readout hypothesis. Biol Psychol 38(2–3):95–115
2. Vinciarelli A, Pantic M, Bourlard H (2009) Social signal processing: survey of an emerging domain. Image Vis Comput 31(1):1743–1759
3. Ryan A et al (2009) Automated facial expression recognition system. In: Proceedings of 43rd annual international Carnahan conference on security technology, Zurich, Switzerland, pp 172–177
4. Russell TA, Green MJ, Simpson I, Coltheart M (2008) Remediation of facial emotion perception in schizophrenia: concomitant changes in visual attention. Schizophr Res 103(1–3):248–256
5. Cohn JF, Zlochower A (1995) A computerized analysis of facial expression: feasibility of automated discrimination. American Psychological Society
6. Lucey P, Cohn JF, Kanade T, Saragih J, Ambadar Z, Matthews I (2010) The extended Cohn-Kanade dataset (CK+): a complete dataset for action unit and emotion-specified expression. In: 2010 IEEE computer society conference on Computer vision and pattern recognition workshops (CVPRW), pp 94–101
7. Hinton G, Salakhutdinov R (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507
8. Chen J, Yan H, Xiang J (2011) Study of decoding mental state based on eye tracks using SVM. Comput Eng Appl 47(11):39–42
9. Hinton G, Osindero S, Teh Y (2006) A fast learning algorithm for deep belief nets. Neural Comput 18:1527–1554

Real Image Reproduction Algorithm Based on Sigmoid-iCAM Color Model

Dongyue Xiao(✉) and Xiaoyan Tang

School of Electronics and Electrical Engineering,
Nanyang Institute of Technology, Nanyang 473004, China
568044943@qq.com

Abstract. Aiming at the characteristics of color prediction and color matching in color appearance model, a vision local adaptation mechanism based on Sigmoid function of artificial neural network is introduced on the basis of iCAM color appearance model, and a real image reproduction algorithm based on Sigmoid-iCAM color appearance model is proposed. Compared with the real image reproduction algorithm based on CIECAM02 and iCAM color model, this algorithm can effectively improve the image contrast and local details while restoring the color of tone-distorted image.

Keywords: Real image reproduction · Color model · Visual adaptation · Sigmoid function

1 Introduction

Color appearance model (CAM) arises with the development of colorimetry. Its original intention is to solve the problem of true color reproduction of different devices, and it has been gradually used in color image quality evaluation. In recent years, more and more scholars combine color matching and prediction model with image tone mapping algorithm to achieve real image reproduction of cross-media images under different observation conditions.

The early color appearance models include Hunt color appearance model [1], Nayatani color appearance model [2], RLAB color appearance model [3] and LLAB color appearance model [4]. On the basis of the above models, CIE successively introduced CIECAM97s [5] color model, CIECAM02 color model [6] and image color model iCAM [7].

In 2002, Akyuz [6] first applied CIECAM02 color model to HDR display. In 2003, Fairchild [7] proposed the image color appearance model (iCAM), which attempts to simulate the complex visual response of the human eye, providing color prediction, color difference measurement and other functions. In 2007, Kuang proposed iCAM06 model [8] based on iCAM color model, which uses bilateral filtering technology to achieve high dynamic range compression while effectively retaining image details.

In 2009, Kunkel [9] combined color adaptation transformation with nonlinear compression, and improved the response of CIECAM02 nonlinear cone cells based on psychological experimental data, making the semi-saturation constant dependent on environmental light and fitness factors.

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 653–658, 2020.
https://doi.org/10.1007/978-981-15-0187-6_78

In China, Wan [10] of Wuhan University proposed a new method for calculating the color adaptation factor, which has a certain degree of detail preservation and color invariance.

The purpose of this paper is to study the real image reproduction technology of color image from the perspective of chromaticity. Aiming at the shortcomings of blurred image and low contrast in color model, a real image reproduction algorithm based on Sigmoid-iCAM color model is proposed by introducing visual local adaptation mechanism on the basis of iCAM color model, in order to improve the contrast and local details of the image while restoring the color of tone-distorted image.

2 Models and Methods

CAM is a complex nonlinear transformation system. It quantitatively calculates the color appearance attributes by observing the changes of white spots, brightness and surrounding environment. An ideal color model should take into account the changes of observation conditions, spatial and temporal characteristics of the image to predict the color attributes and quality changes of the image. Real image reproduction based on color appearance model is similar to the principle used in cross-media color reproduction. Firstly, RGB value of distorted image is converted to tristimulus value X, Y and Z in CIE-XYZ space. Then, ideal image is obtained by steps of color appearance transformation, gamut mapping, inverse color appearance transformation and space inverse transformation.

iCAM is a complex mathematical model with many parameters and nonlinear transformation. Structurally, it can be divided into two parts: positive transformation and inverse transformation. When iCAM is used in real image reconstruction, its positive transformation includes three parts: calculation of observation condition parameters, linear color adaptation transformation and nonlinear hyperbolic response compression.

Physiological studies have shown that visual adaptation is S-shaped and nonlinear. So, Naka and Rushton [11] based on electrophysiological models simulated the S-shaped response equation of rod and cone cells in the process of brightness adaptation and presented a brightness perception model at any level of adaptation, as shown in Eq. (1):

$$R(I) = \frac{I^n}{I^n + \sigma^n} \tag{1}$$

R is the cell response. I is the input brightness. σ is the semi-saturated parameter when the response reaches half of the system, and n is the sensitivity constant. Different σ represents different levels of adaptation.

In addition, the visual system can produce different brightness adaptation responses to different illumination conditions in the same scene. Local adaptation [12] refers to the change of brightness adaptation level in an image. In this paper, the Sigmoid function, a nonlinear function of artificial neural network, is introduced on the basis of brightness adaptation model, as shown in Eq. (2):

$$f(x) = \frac{1}{1 + \exp[-\xi(x - \eta)]} \tag{2}$$

The function ξ, η can be scaled and shifted to a certain extent according to the requirements of different applications.

Discarding the power exponential compression function of iCAM model, combining CIECAM02 nonlinear compression function (Formula 8) with Sigmoid function, introducing color adaptation factor D and semi-saturated parameter control factor α, β, a nonlinear local adaptation model based on Sigmoid is obtained. The calculation is as follows:

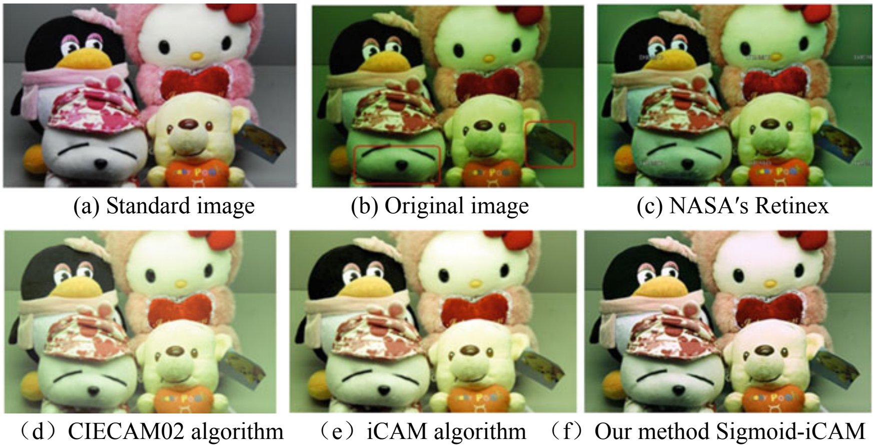
$$\begin{aligned} L^{'} &= \frac{400(F_L LD/100)^n}{(F_L LD/100)^n + [L_a \exp(-\alpha(\log(L) - \beta))]^n} \\ S^{'} &= \frac{400(F_L S/100)^n}{(F_L S/100)^n + [S_a \exp(-\alpha(\log(S) - \beta))]^n} \\ M^{'} &= \frac{400(F_L M/100)^n}{(F_L M/100)^n + [M_a \exp(-\alpha(\log(M) - \beta))]^n} \end{aligned} \tag{3}$$

L_a, M_a and S_a are the local illumination intensity of the image, respectively. The input intensity is obtained by Gauss low-pass filtering. n, α and β are the control factors, which can generate compression curves of different shapes according to the local adaptation level of each region of the image.

The color model of Sigmoid-iCAM can be obtained by adjusting the local contrast of image by Formula (3). After testing a large number of images, it is found that the parameters of n, α and β are different for different content and tone images. The statistical rules are as follows: $n = 0.9 - \left|\frac{\bar{S}-150}{200}\right|$, α, β are constant, usually 0.3, 0.6, 1, 1.5, 2, 2.5, 3, etc.

3 Experimental Results

Figure 1 shows the processing results of different algorithms for tone distortion images. (a) Standard image taken under $D65$ light source. (b) Distorted images obtained with green filter paper. (c) In order to equalize the histogram of distorted image, the influence of light source is removed to some extent, but compared with the standard image; the tone is bluer, such as the area shown in the red circle of the original image. (d) For the CRMSR processing results obtained by PhotoFlair software, the brightness and details of the image are improved, but the tone distortion is not removed well. (e) For the result of CIECAM02 model processing, the tone distortion of the image is improved to a certain extent, but the image is blurred, and the contrast and image details are low. (f) In order to improve the hue distortion, the image contrast and details are well preserved. (g) As a result of Sigmoid-iCAM processing in this paper, the tone distortion is the smallest, and the detail is rich, and the image is the clearest.


(a) Standard image (b) Original image (c) NASA′s Retinex

（d）CIECAM02 algorithm （e）iCAM algorithm （f）Our method Sigmoid-iCAM

Fig. 1. Comparisons of toy standard diagram algorithms

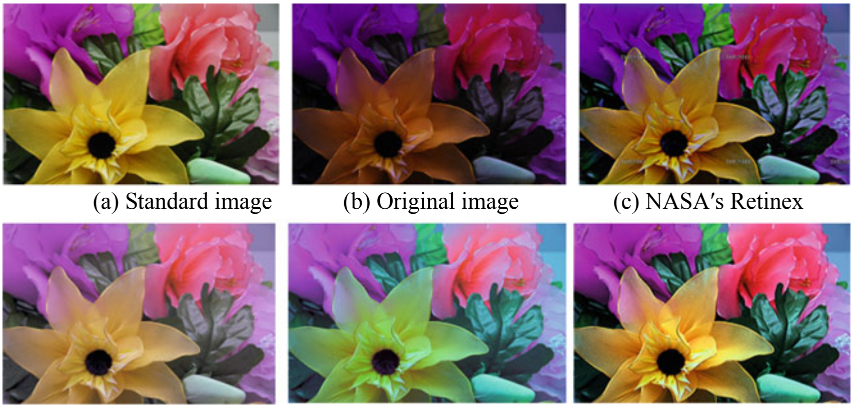
Six parameters, $\bar{I}$, standard deviation SD, entropy H, average gradient G, $\Delta E^*_{\rm ab}$ and ΔE_{00} are used to evaluate the image processed by various algorithms. $\bar{I}$ reflects the overall brightness of the image. The average brightness of the image with better visual effect should be between 120 and 150. Standard deviation SD, entropy H and average gradient G are the evaluation indexes of image contrast, information and image details, respectively. The larger the value, the better the image quality is. According to the formula given by CIEDE2000, we can calculate the color difference $\Delta E^*_{\rm ab}$ and ΔE_{00} [13]. By calculating the color difference between the output image and the input image, we can evaluate the comprehensive distortion of the brightness and color of the image and the standard image. The smaller the value, the smallest the color distortion of the image is.

The objective evaluation results of Fig. 1 are shown in Table 1. It can be seen that $\Delta E^*_{\rm ab}$ and ΔE_{00} of the original image are the largest, that is, the image distortion is serious. $\Delta E^*_{\rm ab}$ and ΔE_{00} of histogram equalization algorithm are small, and SD is the largest, which shows that the algorithm can improve the overall tone of the image and has better contrast ratio. The reconstruction method based on CIECAM02 and iCAM color model also improves the overall tone of the image, but the image value $\bar{I}$ is too high and the SD and G values are low, which indicates that the image is too bright, and the contrast and details are not improved. The SD, H and G values of the image processed by this algorithm are the highest, that is, the contrast, information and details of the image are the best. $\Delta E^*_{\rm ab}$ and ΔE_{00} are the smallest, that is, the image tone distortion is the smallest.

Figures 2 and 3 show the processing effect of other commonly used color distortion images. Generally speaking, the image reproduction algorithm based on Sigmoid-iCAM color model can better reproduce the true color of tone-distorted image.

Table 1. Objective evaluation results

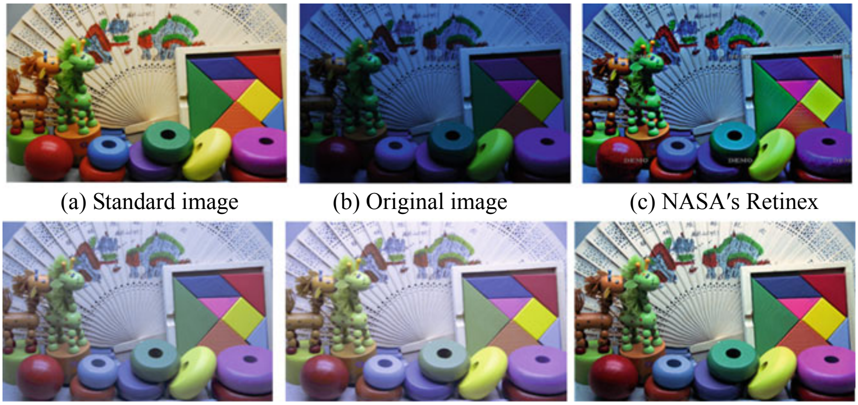
Evaluation algorithm	$\bar{I}$	SD	H	G	$\Delta E^{*}_{\mathrm{ab}}$	ΔE_{00}
Original image	121.2223	65.8646	7.6693	5.6494	101.8264	62.9329
RGB histogram equalization	146.7192	66.5011	7.3195	6.2391	51.5210	42.3950
NASA’s retinex	122.9594	57.4037	7.6588	7.1803	90.7774	53.6448
CIECAM02 algorithm	181.8675	49.6454	7.2931	4.3072	67.9677	33.5983
iCAM algorithm	172.7423	59.1945	7.4771	5.1545	68.0334	33.5307
Our Sigmoid-iCAM algorithm	146.0339	66.7513	7.6715	9.0170	50.6383	22.9777



(a) Standard image (b) Original image (c) NASA′s Retinex

（d）CIECAM02 algorithm （e）iCAM algorithm （f）Our method Sigmoid-iCAM

Fig. 2. Comparisons of flower standard diagram algorithms



(a) Standard image (b) Original image (c) NASA′s Retinex

（d）CIECAM02 algorithm （e）iCAM algorithm （f）Our method Sigmoid-iCAM

Fig. 3. Comparisons of sector standard graph algorithms

4 Conclusion

In order to take account of both the color restoration performance of the color model and the local adaptability of the visual system, this paper introduces the nonlinear Sigmoid function of the artificial neural network on the basis of the iCAM color model, proposes a real image reproduction algorithm based on the Sigmoid-iCAM color model and gives the statistical rules of the control parameters of the model for processing different types of images. Compared with histogram equalization, NASA's Retinex, CIECAM02 color model and iCAM color model-based image reproduction algorithm, the standard deviation, entropy and average gradient of the image processed by this algorithm are the highest, while the color difference based on CIIELAB and CIED2000 is the lowest, which shows that the contrast, information quantity and detail information of the image are the highest, and the color difference between the image and the standard image is the smallest. The color restoration of color image with tone distortion is achieved, and the color invariance of vision system is well realized.

References

1. Hunt RWG, Luo MR (1994) Evaluation of a model of colour vision by magnitude scalings discussion of collected results. Color Res Appl 19(1):27–33
2. Nayatani Y, Hashimoto K et al (1987) A nonlinear color appearance model using estevez-hunt-pointer primaries. Color Res Appl 12(5):231–242
3. Fairchild MD (1994) Visual evaluation and evolution of RLAB color space. In: Proceeding of the IS&T and SID's 2nd color imaging conference. Color systems and applications. Springfield, USA, pp 9–13
4. Luo MR, LoM C, Kuo WG (1996) The LLAB colour model. Color Res Appl 21(6):412–429
5. Fairchild MD (2001) A revision of CIECAM97s for practical applications. Color Res Appl 26(6):418–427
6. Morney N, Fairchild MD, Hunt RWG (2002) The CIECAM02 color appearance model. In: Proceeding of the tenth color imaging conference color science systems and applications, vol 21, 23–27
7. Fairchild MD, Jobnson GM (2002) iCAM: a next generation color appearance model. In: IS&T/SID 10th color imaging conference, Scottsale Spring Berlin, pp 33–38
8. Kuang J, Johnson GM, Fairchild MD (2007) iCAM06: a refined image appearance model for HDR image rendering. J Vis Commun Image Represent 18(5):406–414
9. Kunkel T, Reinhard E (2009) A neurophysiology-inspired steady-state color appearance model. J Opt Soc Am A 26(4):776–782
10. Wan X, Xie D (2009) Color vision based high dynamic range images rendering. World Congr Comput Sci Inform Eng 2009:583–587
11. Naka KL, Rushton WAH (1966) S-potentials from color units in the retina of fish. J Physiol 185(3):536–555
12. Moon P, Spencer D (1945) The visual effect of non-uniform surrounds. J Opt Soc Amer 35 (3):233–248
13. International Commission on Illumination (2001) CIE 142-2001 report: improvement to industrial colour-difference evaluation. Vienna

Game Traffic Classification Based on DNS Domain Name Resolution

Xingchen Xu[1], Mei Nian[2](✉), and Bingcai Chen[1]

[1] College of Computer Science and Technology, Xinjiang Normal University,
830051 Urumqi, Xinjiang, China

[2] College of Computer Science and Technology, Xinjiang Normal University,
No. 102, xinyi road, Shaybak District, Urumqi, Xinjiang 830054, China
2468830639@qq.com
http://www.xjnu.edu.cn/

Abstract. The accurate classification of network game traffic is the technical basis for the campus network resources and the refined management of student learning behavior. This article first uses *DNS reverse resolution* to capture the domain name characteristics of the captured network game traffic, uses the domain name feature to mark the game traffic, and builds the network game feature data set. Then, the network game data set is used to train the decision tree CART to optimize the classification model parameters. Finally, the optimized classification model is tested with Moore standard data set and this data set, respectively. The results show that the accuracy of network *game traffic classification* based on DNS domain name resolution can reach 94%, and its classification performance is better than *Moore data set*.

Keywords: Game traffic classification · Moore data set · CART decision tree · DNS reverse resolution

1 Introduction

Network traffic classification refers to classifying mixed traffic of various applications according to application protocols. Network traffic classification is not only the basis of high-performance network protocol design, but also the basis of network operation management and network development planning, and also an important means of network attack and malicious code detection [1]. At present, some students in colleges and universities are addicted to network games. Playing games and occupying campus network resources also affect students' learning. Fine management needs to adjust the proportion of resources occupied by network games and the impact on students' learning. To this end, it is necessary to classify the campus network traffic data, accurately obtain the bandwidth

Q. Liang et al. (Eds.): *Artificial Intelligence in China*, LNEE 572, pp. 659–667, 2020.
https://doi.org/10.1007/978-981-15-0187-6_79

status of various campus network application traffic, the time spent by students playing games, etc., and then adjust the campus network resources in a targeted manner, alert students who play games. Although the current research on network traffic classification has produced a lot of results, the rapid growth of game size and game traffic has not attracted the attention of traditional academia and industry. The research focus in the field of network traffic classification is video traffic, P2P traffic, and abnormal traffic, and the research results of network game traffic classification are very few.

Although network game traffic is a type of network traffic, it is quite different from other network applications' traffic. Some network game traffic adopts tunnel encapsulation technology, which is concealed, and it has a small proportion in the public traffic data set. The existing network traffic classification technology cannot achieve accurate and comprehensive classification. Secondly, the existing network traffic standard data set, Moore [2], includes only 8 game traffic data, which results in low accuracy of network game traffic classification and fails to meet the requirements of the application.

2 Related Work

Along with the phenomenon that college students are obsessed with the network, the fine management of the campus network requires higher requirements for the school network management. In addition, the university student management department needs to understand the students' learning and entertainment status and prevent students from indulging in the adverse consequences of hanging games and repeating grades caused by online games. The premise of the above tasks requires an accurate understanding of the online behavior of students on campus networks, whether the allocation of campus network resources is reasonable, and whether students' online behavior is healthy. To this end, it is necessary to analyze the behavior of campus online games and the traffic of campus network games. The identification provides a data foundation for the use of campus network resources and student online behavior analysis.

2.1 Network Traffic Classification

Currently, network traffic classification methods mainly include port-based, deep packet inspection, and machine learning. In the early days, TCP-based ports were mainly used for classification. However, with the emergence of P2P traffic and the widespread use of dynamic port numbers, the method of classifying traffic using port numbers is no longer effective [3].

Subsequently, a deep packet inspection (DPI) technology was proposed, which has a very high accuracy by detecting load characteristics and classifying traffic. Therefore, the traffic classification tools currently in large use are products based on DPI technology, such as PACE [4], OpenDPI [5], NDPI [6], L7-filter, Libprotoident [7], and so on. However, the deep packet inspection needs to satisfy two conditions: Firstly, part of the content of the data packet must be visible; secondly, the DPI classifies the specific application by matching the payload code of the network data packet and requires that the payload code of the network data packet is known. The network game traffic cannot meet the above conditions and is not suitable for classification by deep packet parsing.

The network traffic classification hotspot technology is machine learning. The effective features and statistical information of the network flow are extracted by preprocessing the network traffic, the extracted feature information is discovered through machine learning to discover its regularity, and the common features of each type of traffic are counted. Based on the design of the classifier, the network stream is effectively classified and identified. The traffic classification algorithm based on machine learning is not affected by factors such as dynamic port, encryption, and incomplete packet characteristics. The accuracy and reliability of classification are greatly improved compared with port-based and deep packet detection methods. The traffic classification algorithm for machine learning is divided into supervised and unsupervised [8]. For supervised algorithms, a training set is first needed, the training model is used to train the classification model, and then the classification model is used to test the classification performance of the traffic. The accuracy of the supervised algorithm depends on the selection of the feature vector and the construction of the standard data set. The unsupervised algorithm directly mines the data samples to achieve clustering, but cannot accurately determine the type of traffic application.

2.2 Network Game Traffic

The game is mainly divided into three types: platform PC network game, non-platform PC stand-alone game, and browser-based Web game. The traffic of non-platform PC stand-alone games is not transmitted from the network and is not included in this study. Browser-based Web games can use deep packet inspection to achieve accurate classification. This article studies the difficulty in the classification of game traffic, which is the classification of platform PC network game traffic.

Machine learning algorithm overcomes the shortcomings of standard port matching and deep packet detection methods in traditional traffic classification, and the classification based on the network traffic statistics feature does not involve the payload information of the message, effectively avoiding the legal problems caused by user privacy violations [9]. To this end, this article uses machine learning algorithms to classify network game traffic.

3 Game Traffic Classification Based on DNS Domain Name Resolution

The machine learning traffic classification algorithm firstly extracts the features of each piece of data from a large amount of data through feature engineering to form a labeled data set. Then, the machine learning algorithm is used to build the model, and the feature data is trained and learned to form a classifier.

Finally, the test data is classified using a classifier model. The performance of the traffic classifier implemented by machine learning depends on a good tag data set, and to build an excellent classifier, there must be a data set that is sufficiently accurately labeled for training. The difficulty with current game traffic classification is the lack of tagged network game data sets. To this end, the first problem solved in this article is how to construct a good tag data set, and on this basis, the parameter training and optimization of the machine learning classifier use the optimized classifier to accurately classify the network game traffic. Through reverse domain name resolution of game traffic, we can find that the real-time traffic generated by the anti-cheat mechanism of platform PC network games has very distinct domain name characteristics. So we speculate that the traffic generated by the platform PC network game also has very obvious domain name characteristics. It can be used to mark the network game traffic training data set to solve the network game traffic data set annotation and construct a network game traffic data set for training.

3.1 Network Game Traffic Data Set Construction Based on DNS Domain Name Resolution

The statistical results of the platform PC network game rankings and related questionnaires show that the mainstream PC game platforms in domestic universities include Steam, WeGame, and NetEase games. This article selects four mainstream games such as League of Legends, CrossFire, CSGO, Hearthstone to collect information and build a network game traffic data set.

This article first uses Wireshark software to capture network game traffic distributedly and uses tshark tool to segment and filter traffic packets, leaving only data streams such as TCP, UDP, TLS, and DNS and extracting information features of data packets through tshark command. The command is as follows:

$$tshark{-}rinput.pcap{-}Tfields{-}eip.src{-}eip.dst{-}etcp.srcport{-}etcp.dstport{-} eudp.srcport{-}eudp.dstport{-}eip.hdr_len{-}eip.len{-}eip.proto{-}eip.ttl{-}edata.data{-} eframe.time_delta{-}eframe.len{-}etcp.flags.fin{-}etcp.flags.syn{-}etcp.flags.reset -etcp.flags.ack-etcp.flags.push-eip.version-Eheader=y-Eseparator= /t-Equote=d-Eoccurrence=f>output.csv$$

After the captured network traffic data is processed, it only contains the feature data set in the traffic and has not been marked by the traffic type, so it cannot be used for classifier training. Among the features of game traffic, only source IP and destination IP can represent the type of traffic, but because the game traffic has high concealment, the IP address in the traffic does not contain the distinctive platform game features, so the network game traffic data set cannot be marked by IP address. To this end, this article extracts the source IP and destination IP of all traffic data, the DNS reverse domain name resolution is performed according to the extracted IP list, and the platform domain name with obvious feature of the game platform anti-cheat mechanism is selected from the domain name address. On this basis, use the domain name of the game's official

Web site to mark game traffic. Finally, the non-game traffic data set is marked to construct a network traffic data set.

3.2 Classifier

This article mainly solves the problem of accurately dividing network traffic into "game traffic" and "non-game traffic." It is essentially a two-category problem. According to the result of the algorithm, it is "0" or "1" to determine "yes" or "no." In the machine learning algorithm, the C4.5 and ID3 algorithms are based on the information theory entropy model, which uses the information gain ratio selection feature; these classifiers involve a large number of logarithmic operations and a large amount of computation. But the CART classification tree algorithm uses the Gini coefficient instead of the information gain ratio to simplify the model without losing the advantages of the entropy model. This article chooses the CART algorithm as the classification algorithm of game traffic. The Gini coefficient in the CART classification tree indicates the impureness of the model. The smaller the Gini coefficient, the lower the impurity, the better the characteristics, and the better the classification performance.

The CART algorithm assumes that the decision tree is a binary tree, equivalent to recursively dividing each feature, and can divide the input feature space into finite units and determine the predicted probability distribution, i.e., Gini coefficient, on these units.

In the CART algorithm, assuming K categories, the probability of the kth category is p_k, and then, the Gini coefficient is calculated as shown in Eq. (1):

$$Gini(p) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{1}$$

where the K value is 2, the probability that the sample data belongs to the first type of sample is p, and the Gini coefficient is calculated as shown in Eq. (2):

$$Gini(p) = 2p(1 - p) \tag{2}$$

The algorithm consists of two steps. Firstly, the decision tree is generated based on the training data set. Secondly, the generated tree is pruned and the optimal subtree is selected by the verification data set. The minimum loss function is used as the standard of pruning [10]. The generation of a CART decision tree is the process of constructing a binary tree recursively. CART uses the Gini coefficient minimization criterion for feature selection to generate a binary tree. The CART spanning tree algorithm is as follows:

Input: Training data set D, the conditions for stopping the calculation

Output: CART decision tree.

Step 1: Calculate the Gini coefficient of the existing feature pair data set D.

Step 2: Among all possible features A and all their possible segmentation points a, select the feature with the smallest Gini coefficient and its corresponding segmentation point as the optimal feature and the optimal segmentation point [11]. According to the optimal feature and the optimal segmentation point, two child nodes are generated from the node, and the training data set is allocated to the two child nodes according to the feature.

Step 3: Recursively call Step 1–Step 2 on the two child nodes until the stop condition is met.

Step 4: Generate a CART decision tree.

3.3 Classifier Training

The constructed network traffic data set is preprocessed and divided into a training set and a test set according to a ratio of 8:2. In the classifier training, in order to prevent over-fitting of the decision tree, this article limits the minimum Gini coefficient of node division from 0 to 0.2, increments by 0.002, and classifies the minimum Gini coefficient of different nodes. The training and, at the same time, the training data set are subjected to a fivefold cross-validation process, and the maximum cross-validation value is used to determine the optimal hyperparameter of the node partition.

The experimental data of the parameter adjustment process is shown in Fig. 1. The abscissa is the minimum Gini coefficient of the increasing segmentation point, and the ordinate is the cross-validation value of the classifier under the Gini coefficient. According to the trend of the parameters, it can be found that when the Gini coefficient is selected as 0.00202 and the cross-validation value is the largest, it is 0.8504. According to the principle of the CART algorithm, the optimal super-parameter is determined to be 0.00202.

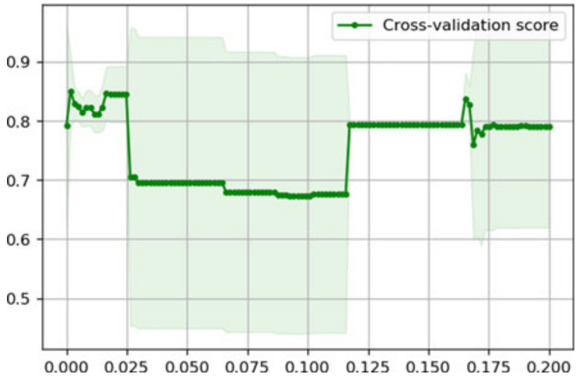


Fig. 1. Trend of cross-validation value with minimum Gini coefficient of segmentation point

4 Experimental Results and Analysis

In order to verify the data set availability and classifier feasibility of this article, the optimized classification model is tested with Moore standard data set and the data set constructed in this article, respectively. In view of the small number of network game traffic samples in Moore standard data, this article firstly adds game traffic data from all entries of Moore standard data to the sixth entry traffic data set for the Moore standard data set as a Moore test set, and the corresponding feature is extracted to form a test data set suitable for the classifier constructed in this paper and then classified and tested with the network game traffic test data set constructed in this article. Finally, compare the results of the two.

4.1 Experimental Results

The test data is sent to the trained classifier, and then the precision, accuracy, recall rate, F1 score, and the number of sample of the network game traffic classification test result of the Moore data set and the data set constructed in the article are extracted. As shown in Table 1.

Table 1. Comparison results

	Moore test set	Data set constructed in this article
Precision	0.83	0.94
Accuracy	0	0.89
Recall	1	0.85
F1-Score	0.01	0.87
Number of test set samples	18415	170955
Number of game traffic samples	8	35260

4.2 Analysis of Results

It can be clearly seen from the results in Table 1 that in the classification test results, the data set based on DNS domain name resolution in both precision, accuracy, and F1 score is superior to the Moore test set. In Table 1, although the accuracy of Moore test set is 0 and recall rate of it is 1, it can correctly classify non-game traffic data sets. The reason is that the lack of tagged game traffic data in the Moore data set affects the training of the trainer, making it less effective in game traffic classification. This article better solves the problem of marked network game traffic, greatly expands the game traffic of the training set mark, and thus realizes the training and optimization of the classifier. The classification test result is much higher than the Moore standard data set. It is verified from the results of Table 1 that it is effective to construct a training data set based on DNS reverse domain name resolution for recognition of game traffic.

5 Conclusion

At present, the commonly used Moore standard data set lacks the data of network game traffic, and its accuracy for game traffic classification is very low and has no practical value. This article performs network game traffic marking based on DNS reverse domain name resolution and completes the construction of online game traffic annotation data set. Using the labeled network game traffic data and training of the classifier, the classifier model is optimized. From the test results, it can be found that the accuracy of the network game traffic classification can reach 94%. The construction technology of the network game traffic tag data set provided by this article is low in difficulty; the classifier training and testing are easy to implement and have certain use value.

The performance of the accurate classification algorithm for network game traffic needs to be further improved. The next step will be to study in depth in the following aspects:

1. Combine the marking method of network game DNS domain name resolution with the semi-supervised clustering marking method to improve the stability of the model.

2. Add the statistical characteristics of the network data stream to the model building to improve the applicability of the model.

3. Optimize the feature selection scheme, and filter the redundant features to improve the operational efficiency of the model.

Acknowledgements. This project was supported in part by the Open Research Fund of Key Laboratory of Data Security, Xinjiang Normal University, under Grant XJNUSYS102017B04 and Graduate Research and Innovation Fund, Xinjiang Normal University, under Grant XSY201902005 and University Scientific Research Project, Xinjiang Autonomous Region, under Grant XJEDU2017S032.

References

1. Wang L, Qian L, Wang D et al (2013) Network traffic classification method and practice. People's Posts and Telecommunications Press, BeiJing
2. Network protocol analysis. https://www.cl.cam.ac.uk/research/srg/netos/projects/archive/nprobe/
3. Karagiannis T, Broido A, Faloutsos M et al (2004) Transport layer identification of P2P traffic. In: Proceedings of the 4th ACM SIGCOMM conference on internet measurement, pp 25–27
4. DPI engine-R & SPACE2. http://www.ipoque.com/products/dpi-engine-rsrpace-2
5. OpenDPI. https://sourceforge.net/projects/opendigt/
6. Deri L, Martinelli M, Bujlow T et al (2014) nDPIOpen-sourcehigh-speed deep packet inspection. In: Proceedings of wireless communications and mobile computing conference, pp 617–622
7. Application layer packet classifier for Linux. https://17-filter.soutceforge.net/2009
8. Zhang W (2015) Research on network traffic classification based on multi-classifier. Yangzhou University, JiangSu

9. Jiwei Liu (2018) A method of network traffic feature selection based on statistical sorting. Commun Netw 1:84–87
10. Wu X, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, McLachlan GJ, Ng A, Liu B, Philip SY, Zhou ZH (2008) Top 10 algorithms in data mining. Knowled Inf Syst 14(1):1–37
11. Li H (2012) Statistical learning method. Tsinghua University Press, Beijing