J. Shanghai Jiao Tong Univ. (Sci.), 2024
https://doi.org/10.1007/s12204-024-2751-y

YOLO-VSF: An Improved YOLO Model by Incorporating Attention Mechanism for Object Detection in Traffic Scenes

MIAO Jun[1*] (苗　军), *GONG Shaocui*[1] (龚少翠), *DENG Yongqiang*[2] (邓永强), *LIANG Hao*[2] (梁　浩), *LI Juanjuan*[2] (李娟娟), *QI Honggang*[3] (齐洪钢), *ZHANG Maoxuan*[1] (张茂炫)
(1. School of Computer Science, Beijing Information Science and Technology University, Beijing 100101, China; 2. Beijing VanJee Technology Co., Ltd., Beijing 100193, China; 3. School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Intelligent transportation and autonomous driving systems have made urgent demands on the techniques with high performance on object detection in traffic scenes. This paper proposes an improved object detection model YOLO-VSF over the YOLOv4 model, which is a representative work with excellent performance among YOLO series of object detection models. The main improvement measures include: The backbone feature extraction network CSPDarknet53 of YOLOv4 is replaced with VGG16 to improve the feature extraction capability; SENet attention mechanism is incorporated to improve the salient and correlation feature representation capability; Focal Loss is integrated into the loss function to overcome the sample imbalance problem. In addition, the detection performance of small targets is improved by increasing the resolution of input images. Experimental results show that on the VanJee traffic image dataset provided by Beijing VanJee Technology Co., Ltd., the proposed YOLO-VSF model achieves an average mean accuracy (mAP) of 92.21 percentage points, which improves the mAP by 3.04 percentage points compared with the YOLOv4 model while maintaining the detection speed of the original model. On the UA-DETRAC dataset, the average accuracy of YOLO-VSF is close to that of the latest YOLOv7 model with the number of parameters reduced by 1.329×10^7. The proposed method can provide a support for object detection in traffic scenes.
Keywords: object detection, traffic scenes, backbone network, attention mechanism, Focal Loss
CLC number: TP391　　**Document code:** A

0 Introduction

With the rapid development of artificial intelligence technology, intelligent transportation systems and autonomous driving have received extensive attention from researchers. In autonomous driving and traffic scenes, as an important module of front-end information collection, object detection technology plays an important role in detecting objects, such as vehicles and pedestrians, and transmits the detected information to the driving control system. Therefore, the design of efficient and accurate object detection algorithms for traffic scenes is of great significance for the development of autonomous driving.

The task of object detection in traffic scenes is to locate the target's position from roads or intersections and classify the type of the target. So far, object detection based on computer vision is generally divided into two categories: traditional object detection and deep learning-based detection. The traditional object detection process is shown in Fig. 1, which first uses sliding windows with different sizes to generate a large number of candidate regions in a measured area, then designs and applies artificial features for candidates' representation, and finally uses a classifier to recognize targets in each candidate region.

The commonly used detection features are scale invariant feature transform (SIFT), histogram of oriented gradients (HOG), and Harr[1-3], and the classifiers are usually support vector machine (SVM) and Adaboost[4-5]. For example, the detection modes could be Harr+Adaboost, HOG+SVM, or deformable parts model (DPM) operation method[6]. The traditional detection algorithms select features relying too much on prior knowledge; however, there are many interference

Received: 2023-08-03　**Accepted:** 2023-11-16
Foundation item: the National Natural Science Foundation of China (No. 62271466), the Beijing Natural Science Foundation (No. 4202025), the Beijing VanJee Technology Co., Ltd. - Beijing Municipal Science and Technology Project (No. Z201100003920003), the Tianjin IoT Technology Enterprise Key Laboratory Research Project (No. VTJ-OT20230209-2), and the Guizhou Provincial Sci-Tech Project (No. zk[2022]-012)
∗**E-mail:** jmiao@bistu.edu.cn

Fig. 1 Traditional object detection process.

factors in the actual application, such as lighting and occlusion, so it is difficult for the traditional vehicle and pedestrian detection algorithms to achieve the accuracy and robustness necessary for practical autonomous driving applications.

In recent years, with the continuous expansion of data volume and updates of hardware devices, the performance of deep learning-based object detection algorithms has been improved dramatically. Due to the different focuses on real-time or accurate performances, deep learning object detection algorithms have gradually developed two routes: one is the two-stage object detection model that focuses on improving detection accuracy as much as possible, and the other is the one-stage object detection model that focuses on further improving detection speed.

The two-stage object detection model first generates a number of candidate regions that are likely to contain a target to be detected, and then followed by some subsequent measures to distinguish which targets are contained in each candidate region. For example, the Region-CNN (RCNN)[7] uses the selective search algorithm[8] to get the candidate regions, and then performs further classification within these regions to get the class of objects. Fast RCNN[9] optimized the RCNN model for the shortcomings of slow detection by integrating the feature extraction into the RCNN sub-network to run detection as a whole module, thus significantly improving the detection speed by sharing computational operations. As a representative algorithm of the current two-stage detection methods, Faster RCNN[10] uses region proposal networks (RPNs) instead of selective search algorithm to complete the object detection task by training a complete model, which effectively avoids the problem of too much repeated computation that needs to train three different branches in RCNN, thus making Faster RCNN the best in terms of accuracy at present.

Single-stage object detection models remove the region search part in the two-stage models, and obtain the prediction results directly from the feature map after down-sampling the original image. For example, the models of you only look once (YOLO) series[11-14] directly turn the task into a uniform regression problem, processing the image once to obtain the target class, confidence and location information from the regression of each pixel in the feature map. Compared with the two-stage object detection algorithm, YOLO has a faster detection speed, but the detection accuracy of the YOLO model is lower due to the lack of a region search and binary classification procedure. Among YOLO series, YOLOv4[14] is a representative object detection algorithm with excellent performance in terms of accuracy and speed, which optimized and improved the backbone feature extraction network, activation function and loss function of the YOLOv3[13].

Inspired by the improvement measures taken in YOLOv4, this paper proposes a VGG SENet focal (YOLO-VSF) detection model which makes the improvement over the YOLOv4 model. The main improvement measures are taken in three aspects: Replacing the backbone feature extraction network of the original model to improve the feature extraction capability; incorporating an channel attention mechanism into the feature extraction network to improve the salient and correlation feature representation capability; integrating a new loss function into the objective function of the original model to overcome the sample imbalance problem.

1 Related Work: YOLOv4 Model

As a representative work of YOLO object detection model series, the YOLOv4 model is intensively introduced in this section for its excellent performance. YOLOv4 combines a large number of previous research techniques with appropriate innovations, and performs well in terms of the balance of object detection speed and accuracy. YOLOv4 is composed of three parts: ① feature extraction part: including a backbone network CSPDarknet53[15] for convolutional feature extraction and a spatial pyramid pooling (SPP) module[16] for multi-scale feature extraction; ② feature fusion part: a PANet (Path Aggregation Network) module[17] for fusion of convolutional features and multi-scale features from the CSPDarknet53 and SPP modules; ③ prediction part: three YOLO Head structures for outputting the predicted positions and classes of objects to be detected. Its total network structure is shown in Fig. 2.

The function of each composed part in the YOLOv4 is described as follows:

(1) Feature extraction part. It consists of two modules including the CSPDarknet53[15] and the SPP[16] modules. CSPDarknet53 is an improved backbone network over that of YOLOv3 for convolutional feature extraction, which splits the original stack of

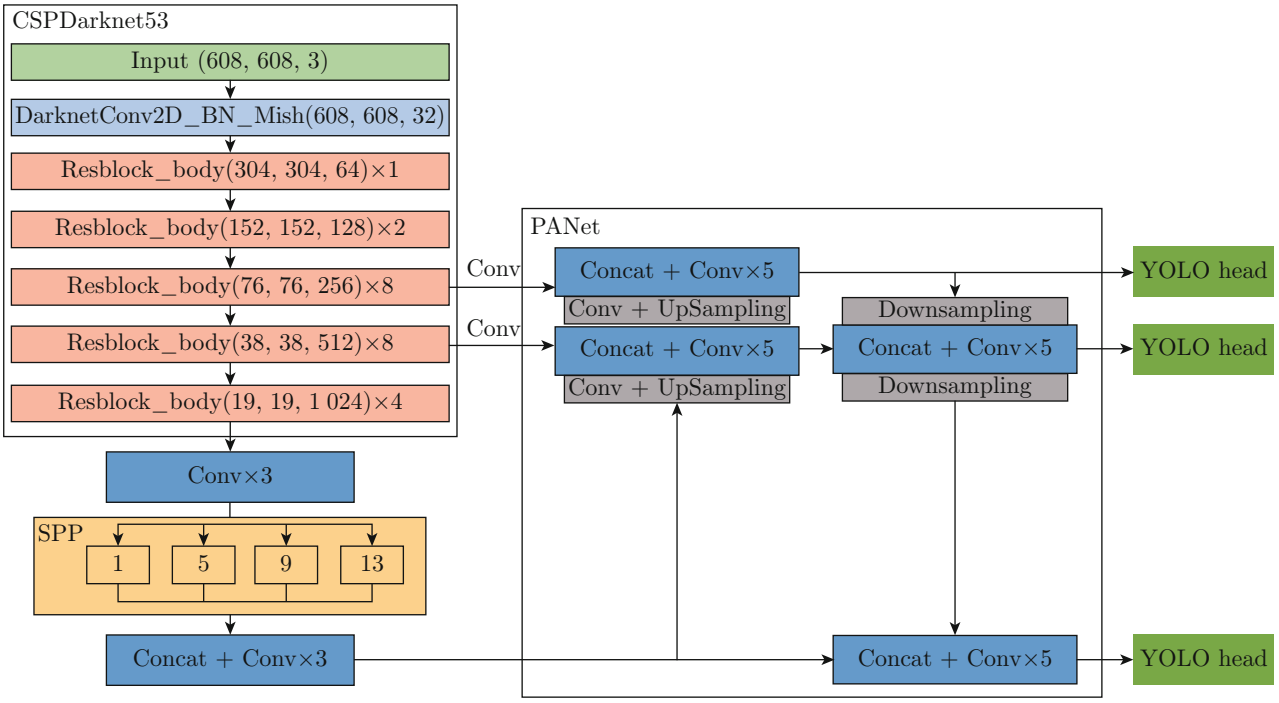

Fig. 2 YOLOv4 network structure.

residual blocks into a backbone path and a residual path by adding a cross stage partial (CSP)[15] structure to the backbone network Darknet53[13] of YOLOv3. This structure achieves a cross-stage feature extraction, which can maintain good detection accuracy while reducing the computational cost. After three convolutions on the last layer of CSPdarknet53, the SPP (Spatial Pyramid Pooling) module is followed for maximum pooling at four different scales with kernel sizes of 13×13, 9×9, 5×5, and 1×1 respectively, which serves to increase the perceptive fields and separate out significant contextual features. In addition, the activation function in YOLO4 is modified from Leaky ReLU to Mish, which is calculated as

$$\text{Mish} = x \tanh \ln(1 + \mathrm{e}^x), \tag{1}$$

where x is the input of the activation function.

(2) Feature fusion part. It is operated by a path aggregation network (PANet)[17]. The PANet receives the convolutional features and the multi-scale features from the CSPDarknet53 and SPP modules, and fuses them into three groups of features to the prediction part by concatenating, convolution, up-sampling and down-sampling the features horizontally and vertically.

(3) Prediction part. It consists of three YOLO heads. They output prediction results of objects to be detected, including the relative position coordinates of the prediction box with respect to the upper left corner point, the width and the height, and the confidence level of the presence of targets in the prediction box, and the probabilities of object classes that the prediction box belongs to.

The working procedure and the training loss functions of the YOLOv4 are introduced as follows:

The original image size is adjusted to 608×608 pixels and images are input to the CSPDarknet53 for feature convolutional feature extraction, then the three feature layers from the CSPDarknet53 and the SPP module are input to the PANet for feature fusion, and finally three scales of fused feature layers with sizes of 19×19, 38×38, and 76×76 are input to three YOLO detection heads to predict positions and classes of small, medium and large targets, respectively.

The loss function of YOLOv4 consists of three components: regression loss (L_{reg}), target confidence loss (L_{conf}) and classification loss (L_{cls}), which are defined as

$$L_{\text{object}} = L_{\text{reg}} + L_{\text{conf}} + L_{\text{cls}}. \tag{2}$$

The regression loss is defined as

$$L_{\text{reg}} = \lambda_{\text{coord}} \sum_{i=0}^{K \times K} \sum_{j=0}^{M} I_{ij}^{\text{obj}} (2 - w_i h_i)(1 - \text{CIOU}), \tag{3}$$

where λ_{coord} is the positive sample weight coefficient, I_{ij}^{obj} determines whether it is a positive sample, and $2 - w_i h_i$ is the penalty term. Additionally, CIOU (complete intersection over union) takes into account the scale information of overlap, center distance and aspect

ratio of the border on the basis of IOU (intersection over union), whose formula is defined as

$$\mathrm{CIOU} = \mathrm{IOU} - \frac{\rho^2(b, b^{\mathrm{gt}})}{a^2} - \beta\nu, \tag{4}$$

where $\rho^2(b, b^{\mathrm{gt}})$ represents the Euclidean distance between the center points of the prediction box and the real box, a represents the diagonal distance of the smallest closed area that can contain both the prediction box and the real box, β is a parameter measuring the consistency of the aspect ratio, and ν is a trade-off parameter.

The confidence loss adopts the cross entropy loss function, which is divided into two parts: obj and noobj, where the loss of the noobj part is multiplied with a weight coefficient λ_{noobj}. In addition, $\hat{C}_i$ is the sample value, and C_i is the prediction value. The corresponding formula is defined as

$$\begin{aligned} L_{\mathrm{conf}} = &- \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\mathrm{obj}}[\hat{C}_i \log C_i + \\ &(1-\hat{C}_i)\log(1-C_i)] - \lambda_{\mathrm{noobj}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\mathrm{noobj}} \times \\ &[\hat{C}_i \log C_i + (1-\hat{C}_i)\log(1-C_i)]. \end{aligned} \tag{5}$$

The classification loss uses a cross-entropy loss function, with c being the class number, as defined in

$$\begin{aligned} L_{\mathrm{cls}} = &- \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\mathrm{obj}} \sum_{c\in\mathrm{classes}} [\hat{p}_i(c)\log p_i(c) + \\ &(1-\hat{p}_i(c))\log(1-p_i(c))]. \end{aligned} \tag{6}$$

2 Methodology

Although YOLOv4 is an excellent object detection model, it still has some spaces to be improved.

Firstly, the backbone net CSPDarknet53[15] in the YOLOv4 is not strong enough. CSPDarknet53 is a deep network which is designed to achieve a high speed running on GPU by adopting small convolutional kernels with sizes of 1×1 and 3×3 to reduce model complexity. Although it wins the performance on time, it had to sacrifice some detection accuracy as the cost. In contrast, the deep neural network of VGG16[18] adopts the convolutional kernels with size of 3×3 for feature extraction and representation. Although VGG16 could spend more computational cost than that of CSPDarknet53, its higher quantity of model parameters may lead to a relatively higher ability for detection accuracy.

Secondly, there are no enough attention mechanisms for correlation feature representation in YOLOv4 model. YOLOv4 modifies the attention module selective attention model (SAM)[13] of YOLOv3 from spatial-wise attention to point-wise attention. However, SAM is only a spatial attention mechanism, and there are still other types of attention mechanisms could be considered. For example, the squeeze-and-excitation (SENet)[19] mechanism, a kind of channel attention mechanism for correlation feature representation, may be considered to be incorporated into the model.

Thirdly, on the task of object detection, only a small portion of candidate regions contain positive targets and the rest are the negative backgrounds, which leads to the imbalance problem of positive/negative samples. Duo to the limit of the objective function of YOLOv4, it cannot handle this problem.

In order to resolve the above problems, this paper proposes an improved model over the YOLOv4. The main improvements are: using more complex model of VGG16 instead of CSPDarknet53 as feature extraction network; incorporating SENet attention mechanism in the feature extraction networks; integrating Focal Loss[20] in the loss function of the original model to balance positive and negative samples during training. Therefore, we name the model proposed in this paper as YOLO-VSF model.

The total YOLO-VSF network structure is shown in Fig. 3. It is composed of a new VGG16 backbone network, three new SENet attention modules, an SPP module, a path aggregation network (PANet) and three YOLO Heads with new loss function of integrated Focal Loss.

The working procedure of the proposed model is as follows: First, the input image is subjected to 3×3 convolution, ReLU activation function, and maximum pooling operations by the VGG16 backbone feature extraction network, and a total of five such groups of operations are performed to obtain three sets of output features in different sizes. Then two of them are processed by two SENet attention modules between VGG16 and PANet and one of them is processed by an SENet attention module between VGG16 and SPP to produce three fused features at different scales which focus more on targets of interest. Finally three YOLO Heads are fed with three scale features and prediction results are output at three detection scales respectively, which include the relative coordinates of the center of the prediction box with respect to the upper left corner of the prediction box, the width and height of the prediction box, the confidence level of the presence of the target in the prediction box, and the probability corresponding to multiple target classes.

The main improvements of the proposed model are elaborated in the next subsections.

2.1 Improving the Feature Extraction by Replacing the CSPDarknet53 with VGG16

With reference to Fig. 2, the backbone network CSPDarknet53 of YOLOv4 is a combination of Resblock_body modules, which is composed of a single down-sampling model and multiple residual structures stacked with 3×3 or 1×1 convolutional kernels. The

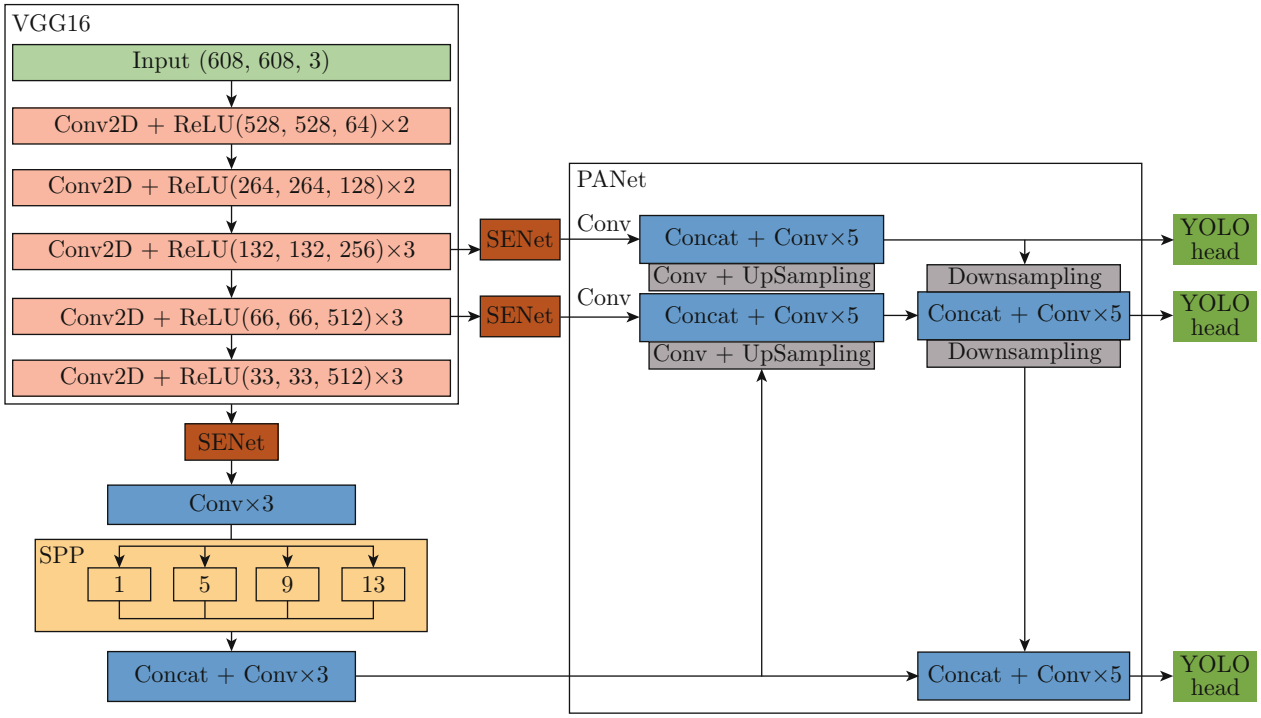

Fig. 3 YOLO-VSF network structure.

CSPDarknet53 has no full connection layers. One of the improvements in this paper is to replace the CSPDarknet53 network structure with VGG16 network[18]. VGG16 uses 3×3 convolution kernels, which has the advantage that using two 3×3 convolution kernels has the same perceptive field as using a 5 × 5 convolution, and using three 3 × 3 convolution kernels has the same perceptive field as using a 7 × 7 convolution. The VGG16 has 13 convolutional layers and three full connection layers. By abstracting layer by layer, VGG16 network is able to continuously learn the features from low to high layers and has stronger nonlinear expression capability to fit more complex features. In addition, as the network deepens, the number of convolutional kernels increases from 64, to 128, 256, and 512, giving it a larger network width and allowing the network layers to learn richer features such as color and texture. These enhance the feature representation ability of the model. The model structure with this improvement measure is shown in Fig. 4.

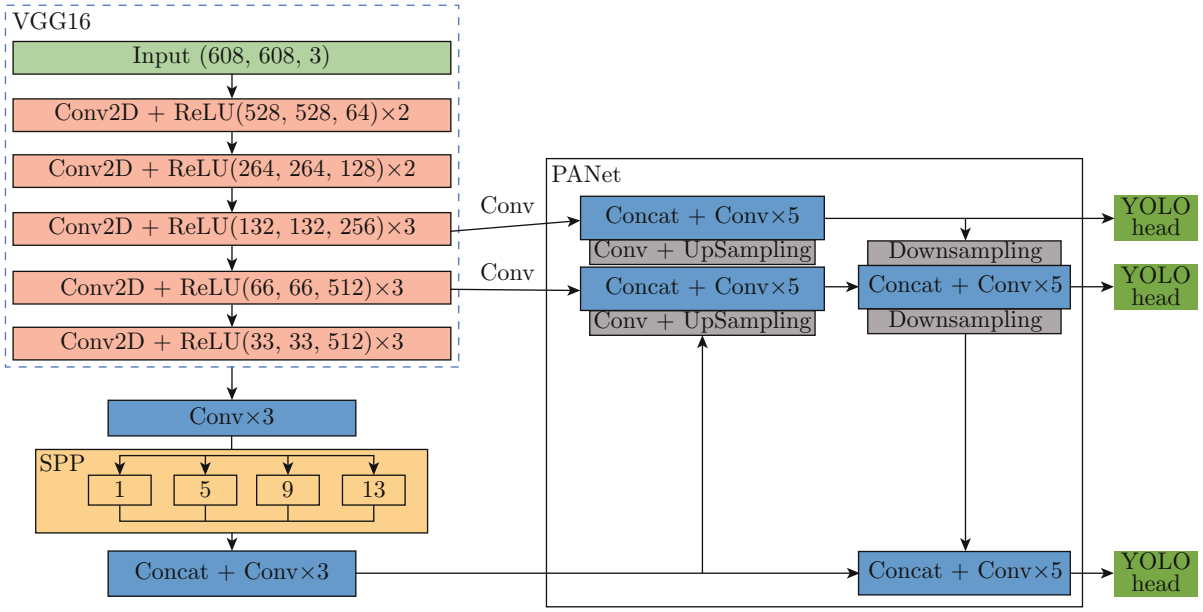

Fig. 4 Replacing the backbone network CSPDarknet53 with VGG16 for stronger feature extraction.

2.2 Improving the Salient and Correlation Feature Representation by Incorporating Channel Attention Mechanism

Since the convolution operation assigns the same weight to each channel of different feature maps, YOLOv4 lacks the ability to describe salient or correlation features between channels. As the representative model of channel attention mechanism, SENet[19] can optimize the channels by introducing an weighting mechanism so that the model can better describe the salient and correlation features. The SENet works through an SE operation, which is divided into 3 main steps, as shown in Fig. 5.

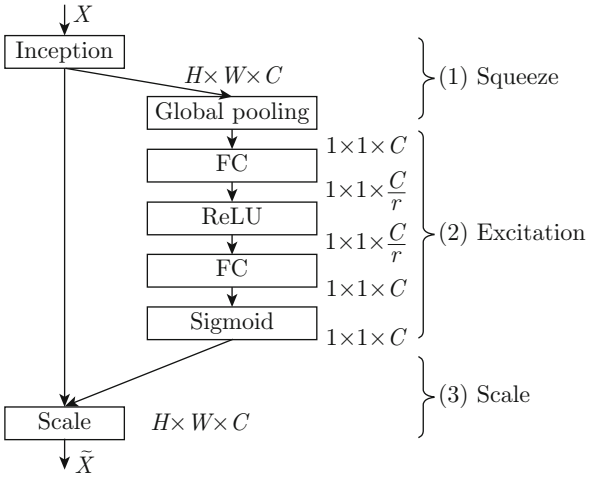

Fig. 5 Structure of SENet for channel weighted attention.

The first step Squeeze is a global average pooling operation that is performed on a group of feature maps with size of $H \times W \times C$, and a squeezed feature map with size of $1 \times 1 \times C$ will be obtained after the operation, which has a global receptive field. The second step Excitation is an operation to perform nonlinear transformation on the result of the squeeze operation by first reducing the dimension and then increasing the dimension of the two full connection layers, and it generates weights for each feature channel to represent the correlation between the feature channels. This operation can not only reduce the complexity of the model, but also better fit the complex correlation between channels and improve the generalization ability of the model. The third step Scale is an operation that applies the weights obtained in the Excitation operation to the original feature maps by channel through multiplication to complete the weighting of the original feature in the channel dimension. In short, the principle of the SENet module is to enhance the important features and weaken the unimportant features.

As shown in Fig. 6, this paper incorporates two SENets between the VGG16 and the PANet as well as one SENet between the VGG16 and the SPP module to represent more salient features.

2.3 Handling Imbalance Problem of Class Samples by Integrating the Focal Loss into the Loss Function

On the object detection task, each image may generate tens of thousands of target candidates, but only a small portion of them contain targets, and the rest are image background, which results in the class sample

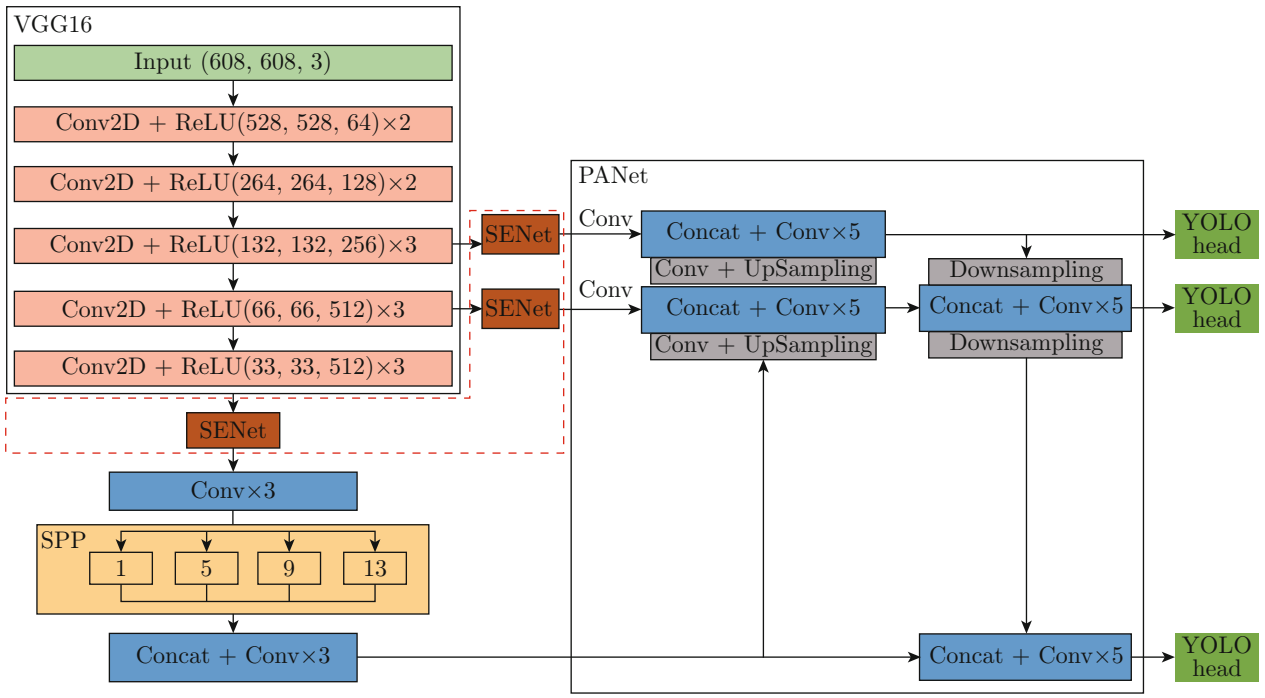

Fig. 6 Incorporating the channel attention mechanism with three SENets.

imbalance problem. The loss function of YOLOv4 consists of three components: regression loss (L_{reg}), target confidence loss (L_{conf}) and classification loss (L_{cls}). These losses are not able to solve the sample imbalance problem. In contrast, the Focus Loss function[20] can decrease the weights of samples easy to classify and focus more on samples difficult to classify during training, which is beneficial to overcome the imbalance problem. In the following paragraphs, we discuss how to improve the original model by integrating the Focal Loss into the loss function of the YOLOv4 to address the imbalance effect of positive and negative samples.

For the general binary classification problem, the cross-entropy loss function for a single sample (p, y) is defined as

$$\text{CE}(p, y) = \begin{cases} -\log p, & y = 1 \\ -\log(1-p), & \text{otherwise} \end{cases}, \tag{7}$$

where p is the probability that the predicted sample belongs to the positive sample.

For the classification problem of unbalanced data set, in order to reduce the influence of negative samples, the weights α and $1-\alpha$ can be applied to each of positive and negative samples, and the balance of positive and negative samples can be achieved by setting the value of α. In this way, the cross-entropy loss function can be improved, as shown in

$$\text{CE}(p, y, \alpha) = \begin{cases} -\alpha \log p, & y = 1 \\ -(1-\alpha)\log(1-p), & y = 0 \end{cases}. \tag{8}$$

The Focal Loss function combines the above two weighting methods to achieve both the weighting of positive and negative samples and the weighting of samples easy and hard to classify, which is defined as

$$\text{FocalLoss}(p, y, \alpha) = \begin{cases} -\alpha(1-p)^{\gamma} \log p, & y = 1 \\ -(1-\alpha)p^{\gamma}\log(1-p), & \text{otherwise} \end{cases}, \tag{9}$$

where γ is an adjustment coefficient. Comparing with the cross-entropy loss function, we can see that Focal Loss can dynamically adjust the weights of samples easy and hard to classify in the loss function during the training process.

Finally, the Focal Loss is integrated into the target confidence loss (L_{conf}) of Eq. (5), which is shown in

$$\begin{aligned} L_{\text{conf}} = &- \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{obj}} \alpha(1-p)^{\gamma} \times \\ &[\hat{C}_i \log C_i + (1-\hat{C}_i)\log(1-C_i)] - \\ &- \lambda_{\text{noobj}} \sum_{i=0}^{K\times K}\sum_{j=0}^{M} I_{ij}^{\text{noobj}} (1-\alpha) p^{\gamma} \times \\ &[\hat{C}_i \log C_i + (1-\hat{C}_i)\log(1-C_i)], \end{aligned} \tag{10}$$

where α is set to 0.25 and γ is set to 2[20].

The above three subsections are the main improvements proposed in this paper. In order to verify the effect of the improved method in this paper, we carried out comparison experiments on the dataset provided by Beijing VanJee Technology Co., Ltd. and the UA-DETRAC[21] dataset. The experiments show that the mean average precision (mAP) of the proposed model in this paper is improved compared with YOLOv4 and can accurately complete the detection of vehicle and pedestrians, and the details of the experiments are shown in Section 3.

3 Experiments

This section provides a detailed description about the experimental environment, data sets, training parameters, evaluation metrics, and experimental comparisons of with and without improved parts to illustrate the effectiveness of the model proposed in this paper. In addition, experiments are conducted for images in different resolutions, which show that increasing the resolution of images can significantly improve the overall average accuracy of the detection of small objects.

3.1 Experimental Environment

The hardware configuration for these experiments is: CPU: Intel(R) Core(TM) i7-7820X CPU@3.60 GHz; GPU: NVIDIA GeForce RTX 2080 Ti. The software environment is: Linux operating system, Torch 1.11.0 deep learning framework, Python 3.7 programming environment, and PyCharm development tools; NVIDIA CUDA10.1 and cuDNN 7.6.3 are configured to accelerate GPU computing, and a series of third-party libraries, such as NumPy, are installed to support the running of codes. To improve the convergence speed of the network, stochastic gradient descent (SGD) is used to learn and update the network parameters during the training of the network model.

3.2 Object Detection Datasets in Traffic Scenes

The goal of this paper is oriented to the task of vehicle-pedestrian detection in traffic scenes for vehicle infrastructure cooperation. The existing large publicly available datasets, such as COCO[22] and VOC[23], are not applicable to our task. Therefore, in this paper two object detection datasets in traffic scenes are chosen for experiments.

The first object detection dataset in traffic scenes used in the experiment is provided by Beijing VanJee Technology Co., Ltd. who captured moving objects at city crossroads with fixed cameras. It includes a total

of 3 032 images of 1 920 × 1 080 pixels with object annotation information, the objects of which are divided into 7 categories: pedestrian, bicycle, motorbike, car, van, truck, and bus. The object number of each category is shown in Table 1. The image examples from the Beijing VanJee dataset are shown in Fig. 7.

Table 1 Beijing VanJee dataset

Dataset category	Number
Pedestrian	1 514
Bicycle	8 293
Motorbike	826
Car	18 273
Van	1 670
Truck	502
Bus	414

Fig. 7 Image examples of the Beijing VanJee dataset.

The data set is divided into training set, validation set, and test set according to a ratio of 81 : 9 : 1. The training techniques in YOLOv4, such as Mosaic data enhancement, Label Smoothing, CIOU, and learning rate cosine annealing decay, are adopted for the proposed model.

The second object detection dataset in traffic scenes is UA-DETRAC dataset[21]. It is an open vehicle object detection dataset with rich traffic scenes, which contains 60 image sequences, totally 83 791 frames of images with size of 960 × 540 pixels. We divide the objects of vehicles in the dataset into four categories, namely: car, bus, van, and others. The dataset is also divided into training set, validation set, and test set with the same ratio as the Beijing VanJee dataset. The image number of each category is shown in Table 2. The image examples from the UA-DETRAC dataset are shown in Fig. 8.

Table 2 UA-DETRAC dataset

Dataset category	Number
Car	453 230
Bus	30 299
Van	51 285
Other	3 358

Fig. 8 Image examples of the UA-DETRAC dataset.

3.3 Parameters Setting for Model Training

For the first dataset of Beijing VanJee dataset, all the experimental training number is limited to 300 epochs. For the second dataset of UA-DETRAC, the training number is limited to 100 epochs. Other parameters for the training on the above two datasets are set the same as follows: the batch size is set to 8, the initial learning rate is 0.01, the momentum factor is 0.937, the weight decay factor is set to 0.000 5, and the IOU threshold is set to 0.5.

3.4 Evaluation Indicators

The experimental performance of tested models is evaluated in terms of Precision (P), Recall (R), F1-Score, Average Precision (AP), and mean Accuracy Precision (mAP).

The calculation formulas of Precision rate, Recall rate and F1-Score are defined in

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{11}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{12}$$

$$\text{F1-Score} = \frac{2 \times \text{Precison} \times \text{Recall}}{\text{Precison} + \text{Recall}}, \tag{13}$$

where, TP (true positives) is the number of positive samples predicted to be positive samples; FN (false negatives) is the number of positive samples predicted to be negative samples; FP (false positives) is the number of negative samples predicted to be positive samples.

The average precision (AP) and the mean average precision (mAP) can be used to measure the quality of the model, which needs to be calculated according to the Recall and Precision of the model. Establish coordinate system with P (Precision) and R (Recall) as coordinate axis respectively, and AP is the area enclosed by P-R curve. The calculation formulas are shown in

$$\mathrm{AP} = \int_0^1 P(R)\mathrm{d}R, \tag{14}$$

$$\mathrm{mAP} = \frac{1}{N}\sum_{i=1}^{N}\mathrm{AP}_i. \tag{15}$$

3.5 Optimal Experiments for Choosing Best Improved Modules and Image Resolution

For YOLOv4, its entire network can be divided into three parts: feature extraction part including a backbone network CSPDarknet53 for convolutional feature extraction and an SPP module for multi-scale feature extraction, the feature fusion part of PANet, and the prediction part of YOLO Heads.

In order to evaluate the individual contributions of improved modules to the overall performance of the improved model and choose the optimal improvement, this paper adopts an incremental approach to gradually modify related modules and image resolutions in three parts of YOLOv4 and observe whether they can improve the model performance individually.

The experiments are conducted on the Beijing VanJee dataset. To accelerate training, the images in the Beijing VanJee dataset are normalized to a middle level of resolution of 608×608 pixels with 3 channels of RGB colors for the experiment in Subsection 3.5.1. To find an optimal image resolution with which the model could perform better for small objects, the images in the Beijing VanJee dataset are normalized to the low, middle and high levels of resolutions of 416×416, 608×608, and $1\,056 \times 1\,056$ pixels respectively for the experiment in Subsection 3.5.2. As a result of finding that images with resolution of $1\,056 \times 1\,056$ pixels perform better for small objects, all the images in the Beijing VanJee dataset are normalized to a size of $1\,056 \times 1\,056$ pixels for the experiments in Subsections 3.5.3 and 3.5.4.

3.5.1 *Evaluation of Improvements on the Backbone Feature Extraction Network*

The function of the backbone network CSPDarknet53 is to perform convolutional feature extraction. As we discussed in Section 2, the CSPDarknet53 could be replaced by other more powerful convolutional networks. In order to choose the best alternative network to improve the backbone network, a series of convolutional networks are tested, which include lightweight networks such as mobilenetV1[24], mobilenetV2[25], mobilenetV3[26] and GhostNet[27], as well as heavyweight networks, such as Resnet-50[28], Densenet121[29] and VGG16[18].

To accelerate training, the images in the Beijing VanJee dataset are normalized to a middle level of resolution of 608×608 pixels with 3 channels of RGB colors for the experiment in this subsection. The experimental results are shown in Table 3, where YOLOv4_net represents that the YOLOv4 backbone network is replaced with some net model; YOLOv4_Mobilenetv1 means that the YOLOv4 backbone network is replaced with a Mobilenetv1 network.

Table 3 Performance comparison with different backbone networks in the Beijing VanJee dataset (image resolution at middle level: 608×608 pixels)

Model	AP/%							mAP/%
	Bicycle	Bus	Car	Motorbike	Van	Pedestrian	Truck	
YOLOv4	86.01	95.42	96.42	50.56	59.81	28.45	94.96	73.09
YOLOv4_Mobilenetv1	80.74	94.64	93.4	16.45	55.35	2.1	97.2	62.84
YOLOv4_Mobilenetv2	75.63	94.51	93.62	5.87	57.17	1.28	92.97	60.15
YOLOv4_Mobilenetv3	72.18	89.09	91.99	26.99	45.15	0.2	89.85	59.35
YOLOv4_Ghostnet	53.36	92.93	92.44	28.51	48.4	0.24	94.52	58.63
YOLOv4_Resnet50	90.97	99.29	97.37	10.66	90.32	39.64	97.27	75.07
YOLOv4_Densenet121	90.14	97.92	96.97	31.76	83.14	33.3	99.78	76.15
YOLOv4_VGG16	**91.49**	**95.95**	**97.12**	**71.96**	**88.82**	**39.19**	**99.3**	**83.29**

From Table 3, it can be learned that in terms of detection accuracy, after the backbone network is replaced by lightweight networks such as Mobilenet series and Ghostnet, the mAP is 10—13 percentage points lower than YOLOv4's, and the detection of small object objects, such as pedestrians, is poorer. It can be also seen that lightweight networks' detection accuracies for small targets (e.g., pedestrian) are lower. In contrast, the mAP performance of YOLOv4 could be improved by 1.98 percentage points and 3.06 percentage points by

YOLOv4_Resnet50 and YOLOv4_Densenet121, respectively. Especially, the mAP could be improved by 10.2 percentage points when using the YOLOv4_VGG16 model compared to the YOLOv4 model. The above results show that replacing the feature extraction network of YOLOv4 with Resnet50, Densenet121 and VGG16 can effectively improve the object detection accuracy, among which the replacement with VGG16 has the most obvious improvement.

However, we also find that the detection results of all the models for small objects, i.e., pedestrians, are not as good as that for other big objects. An intuitive idea to solve this problem is to improve image resolution, as all the feature extraction parts of the aforementioned models can extract more discriminative features from enlarged small objects. In the next subsection, we will conduct experiments on the basic YOLOv4 with different image resolutions to verify our idea.

3.5.2 *Evaluation of the Improvement on Increasing Image Resolution*

In the object detection task, small objects carry less information due to their low resolution and blurred images. The resulting feature representation is weak, which means that very few features can be extracted for a stable detection of small targets. A simple approach to this problem is to increase the image resolution for extracting more features, e.g., setting cameras to capture moving objects in the crossroads with higher resolution than the ordinary resolution.

To find an optimal image resolution with which the model could perform better for small objects, the images in the Beijing VanJee dataset are normalized to the low, middle and high levels of resolutions of 416×416, 608×608, and $1\,056 \times 1\,056$ pixels respectively for the experiment.

The experiment is carried out on the basic model of YOLOv4 to compare the detection results for pedestrians with different image resolutions, which are shown in Table 4. The Precision-Recall (P-R) curves of detection for pedestrians are shown in Fig. 9.

Table 4 Performance comparison of different image resolutions for detection of the small targets of pedestrians (evaluated by AP) and all the objects (evaluated by mAP)

Model	Image resolution/pixels	AP/%	mAP/%
YOLOv4	416×416	3.71	68.10
	608×608	28.45	73.09
	$1\,056 \times 1\,056$	**68.03**	**89.17**

With reference to Table 4 and Fig. 9, it can be learned that increasing the resolution of the input image can obviously improve the detection performance of small targets as well as that of all the objects. For example, when the resolution is increased from the middle level of 608×608 pixels to the high level of $1\,056 \times 1\,056$ pixels, the detection AP for pedestrians reached 68.03% which is 39.58 percentage points higher than that for the middle level of resolution, and simultaneously the detection AP for all the objects reached 89.17% which is 16.08 percentage points higher than that for the resolution of 608×608 pixels. On the contrary, if we decrease the image resolution from the middle level of 608×608 pixels to the low level of 416×416 pixels, the detection AP for pedestrians is 3.71% which is 24.74 percentage points lower than that for the middle level of resolution, and simultaneously the detection AP for all the objects decreased to 68.10% which is 4.99 percentage points lower than that for the middle level of resolution.

As a result of the optimal resolution experiment in this subsection, the succeeding experiments in Subsections 3.5.3 and 3.5.4 adopt $1\,056 \times 1\,056$ pixels as the optimal resolution of the input images.

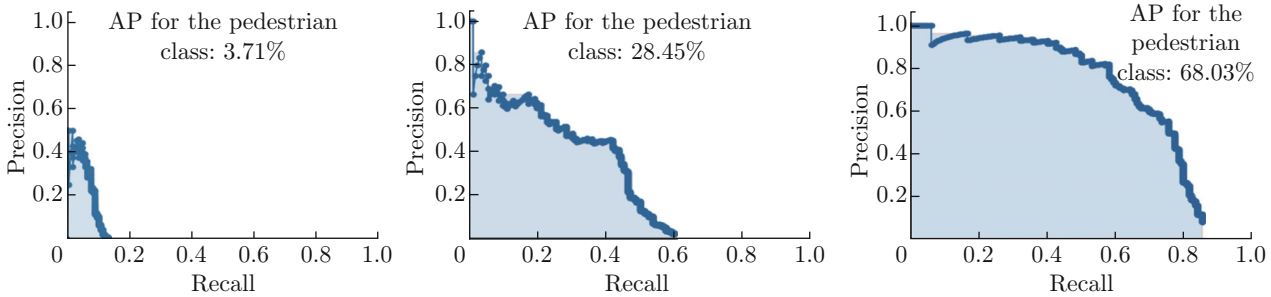

Fig. 9 P-R curves and AP values of different resolutions for detection of small targets (pedestrian).

3.5.3 *Evaluation of Incorporating Attention Mechanisms for Salient and Correlation Feature Extraction*

In order to enhance the ability to describe the salient features of input images and to better acquire the correlation features on different channels, we incorporate different attention mechanisms, such as CBAM[30], ECANet[31] and SENet[19], into the places between three feature map layers of the backbone network and the SPP/PANet modules for optimal experiments. A

performance comparison between the original YOLOv4 model and the modified YOLOv4 model that are incorporated with the above attention mechanisms is shown in Table 5.

Table 5 Performance comparison with different incorporated attention mechanisms (image resolution at high level: 1 056 × 1 056 pixels)

Model	mAP/%
YOLOv4	89.17
YOLOv4 with ECANet	89.24
YOLOv4 with CBAM	89.30
YOLOv4 with SENet	**89.45**

As can be seen from the table, compared with YOLOv4, the mAP is improved by 0.07, 0.13 and 0.28 percentage points after the incorporation of ECANet, CBAM and SENet attention mechanisms respectively. The above experimental results show that the optimal attention mechanism to increase the detection accuracy is the SENet, which is considered to be chosen to incorporate into the final improved model.

In addition, to verify the effects of incorporating attention mechanisms, this paper generates several class activation maps from original images, as shown in Fig. 10. From the figure, it can be clearly observed that the proposed model with incorporated attention mechanism mainly focuses on the vehicle areas to be detected in the images.

Fig. 10 Illustration of class activation maps (the 2nd row) from original images (the 1st row).

3.5.4 *Evaluation of the Improvement on the Loss Function*

In order to overcome the sample imbalance problem, this paper integrates the Focal Loss to the target confidence loss function of YOLOv4. To verify the incremental contribution of the improved modules on the detection effect, the YOLOv4 model is gradually changed in order. That is, the performances of the YOLOv4 model with substituted VGG backbone network (YOLO-V), the YOLO-V model with incorporated SENet attention mechanism (YOLO-VS) and the YOLO-VS model with integrated Focal Loss function (YOLO-VSF) are compared. The comparison experimental results are shown in Table 6. In this experiment, the positive and negative sample balance parameter α in Eq. (8) is set to 0.25, and the hard and easy classification sample balance parameter γ in Eq. (10) is set to 2, so that the model focuses more on the hard classification samples.

From Table 6, it can be learned that the proposed method (YOLO-VSF) in this paper improves the mAP by 3.04 percentage points compared to YOLOv4, 0.61 percentage points compared to YOLO-V, and 0.37 percentage points compared to YOLO-VS, respectively. At the same time, the increase of 2.43 percentage points of mAP from YOLO-V to YOLOv4 indicates that VGG16 has better feature extraction capability compared to CSPDarknet53; the increase of 0.24 percentage points of mAP from YOLO-VS to YOLO-V and the increase of 0.37 percentage points of mAP from YOLO-VS to

Table 6 Performance comparison of incremental improvements (image resolution at high level: 1 056 × 1 056 pixels)

Model	mAP/%	FPS
YOLOv4	89.17	8.456 4
YOLO-V	91.60	5.281 8
YOLO-VS	91.84	5.334 4
YOLO-VSF	**92.21**	5.392 4

YOLO-V indicate the effectiveness for the incorporated channel attention mechanism SENet and the integrated Focal Loss function respectively.

3.6 Comparison Experiments with the YOLOv7 Model

YOLOv7[32] is one of the latest models with excellent performance in YOLO model series. In order to further verify the effectiveness of the improved model, the proposed YOLO-VSF is compared with some commonly used object detection models on the Beijing VanJee set dataset and the open UA-DETRAC dataset. The experimental results are shown in Table 7 and Table 8, respectively.

Table 7 Performance comparison on the Beijing VanJee dataset (image resolution at high level: 1 056 × 1 056 pixels)

Model	mAP/%	F1-Score	Number of parameters
SSD[33]	91.11	0.855	$2.414\,6 \times 10^7$
RetinaNet[20]	74.71	0.735	$3.645\,4 \times 10^7$
Faster RCNN[20]	66.56	0.651	$1.368\,12 \times 10^8$
YOLOv4	89.17	0.873	6.44×10^7
YOLOv7	92.56	0.900	3.69×10^7
YOLO-VSF	92.21	0.915	2.361×10^7

Table 8 Performance comparison on the UA-DETRAC dataset (image resolution: 608 × 608 pixels)

Model	mAP/%	F1-Score	Number of parameters
SSD[33]	94.10	0.870	$2.414\,6 \times 10^7$
RetinaNet[20]	91.16	0.890	$3.645\,4 \times 10^7$
YOLOv4	97.26	0.955	6.44×10^7
YOLOv7	99.55	0.980	3.69×10^7
YOLO-VSF	98.24	0.977	2.361×10^7

As can be seen from Tables 7 and 8, the mAPs of the proposed model YOLO-VSF increase by 3.04 percentage points and 0.98 percentage points compared to the YOLOv4, and the numbers of parameters decrease to 2.361×10^7, which accounts for only 36.66 percentage points of the quantity of parameters of the YOLOv4. The mAPs of YOLO-VSF are close to that of YOLOv7, but the number of parameters of YOLO-VSF is lower than that of YOLOv7 by 1.329×10^7. The YOLO-VSF model achieved a high F1-Score while ensuring relatively high mAP and fewer parameters.

Figure 11 shows an effect comparison of the detection examples using the YOLOv4, YOLOv7 and the proposed YOLO-VSF on the UA-DETRAC dataset. Due to lack of channel attention mechanism, both YOLOv4 and YOLOv7 models lose the correlation information between their feature channels which may not provide a powerful support to detect object robustly. For example, it can be seen from Fig. 11 that the same small target of a bus far away from the camera in Figs. 11(a) and 11(b) is not detected by YOLOv4 and YOLOv7 while a handrail and a car are mistakenly detected as a bus and a van respectively; in contrast, our improved model YOLO-VSF can detect the small target of bus and the target of van, as shown in Fig. 11(c).

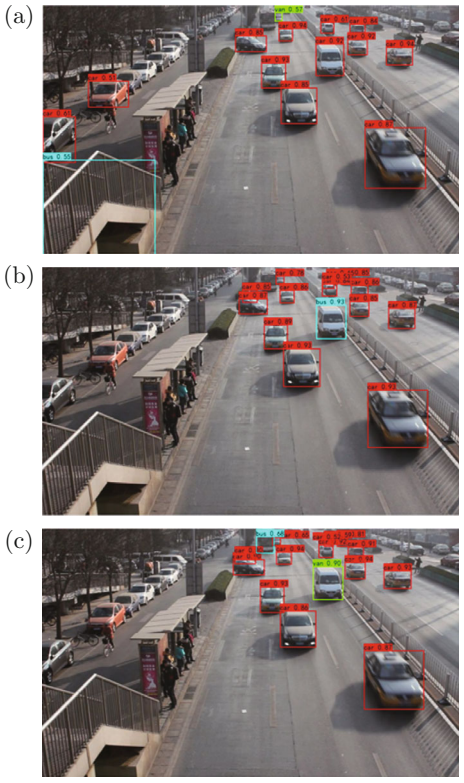

Fig. 11 Effect comparison of detection examples using three models on the UA-DETRAC dataset. (a) YOLOv4; (b) YOLOv7; (c) YOLO-VSF.

4 Conclusion

In this paper, a YOLO-VSF model is proposed which is improved over the YOLOv4 object detection model. The proposed model provides contributions as follows:

(1) Substituting VGG16 for the backbone network CSPDarknet53 improves the feature extraction ability, which makes a significant increase on the detection accuracy.

(2) Incorporating channel attention mechanism of SENet between the feature extraction network and the SPP/PANet parts makes the model focus more on the targets of interest and the correlation information between feature channels.

(3) Integrating Focal Loss in the loss function makes the model focus more on the hard-to-classify samples,

which improves the detection ability of the model.

The experimental results show that the proposed model achieves an average detection accuracy of 92.21 percent points on the Beijing VanJee dataset. And on the UA-DETRAC dataset, the proposed model improves the average detection accuracy while decreasing the number of parameters by about 4×10^7 compared with YOLOv4, and it reaches an average detection accuracy close to that of YOLOv7 by decreasing about 1.3×10^7 parameters. The model proposed in this paper obtains the improvement in both accuracy and number of parameters, which can provide supports for the applications on practical object detection in traffic scenes. In the future study, as a new direction, models of transformers[34-36] could be considered to be embedded into the model to achieve higher performance.

Conflict of Interest The authors declare that they have no conflict of interest.

References

[1] BAY H, TUYTELAARS T, VAN GOOL L. SURF: Speeded up robust features [M]//Computer Vision – ECCV 2006. Berlin, Heidelberg: Springer, 2006: 404-417.

[2] DALAL N, TRIGGS B. Histograms of oriented gradients for human detection [C]//*2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. San Diego: IEEE, 2005: 886-893.

[3] VIOLA P, JONES M. Rapid object detection using a boosted cascade of simple features [C]//*2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Kauai: IEEE, 2001: 511-518.

[4] SUYKENS J A K, VANDEWALLE J. Least squares support vector machine classifiers [J]. *Neural Processing Letters*, 1999, **9**(3): 293-300.

[5] FREUND Y, SCHAPIRE R E. A desicion-theoretic generalization of on-line learning and an application to boosting [M]//Computational learning theory. Berlin, Heidelberg: Springer, 1995: 23-37.

[6] FELZENSZWALB P, MCALLESTER D, RAMANAN D. A discriminatively trained, multiscale, deformable part model [C]//*2008 IEEE Conference on Computer Vision and Pattern Recognition*. Anchorage: IEEE, 2008: 1-8.

[7] GIRSHICK R, DONAHUE J, DARRELL T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation [C]//*2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus: IEEE, 2014: 580-587.

[8] UIJLINGS J R R, VAN DE SANDE K E A, GEVERS T, et al. Selective search for object recognition [J]. *International Journal of Computer Vision*, 2013, **104**(2): 154-171.

[9] GIRSHICK R. Fast R-CNN [C]//*IEEE International Conference on Computer Vision*. Santiago: IEEE, 2015: 1440-1448.

[10] REN S Q, HE K M, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, **39**(6): 1137-1149.

[11] REDMON J, DIVVALA S, GIRSHICK R, et al. You only look once: Unified, real-time object detection [C]//*2016 IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas: IEEE, 2016: 779-788.

[12] REDMON J, FARHADI A. YOLO9000: Better, faster, stronger [C]//*2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017: 6517-6525.

[13] REDMON J, FARHADI A. YOLOv3: An incremental improvement [DB/OL]. (2018-04-08). http://arxiv.org/abs/1804.02767

[14] BOCHKOVSKIY A, WANG C Y, LIAO H Y M. YOLOv4: Optimal speed and accuracy of object detection [DB/OL]. (2020-04-23). http://arxiv.org/abs/2004.10934

[15] WANG C Y, MARK LIAO H Y, WU Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN [C]//*2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. Seattle: IEEE, 2020: 1571-1580.

[16] HE K M, ZHANG X Y, REN S Q, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, **37**(9): 1904-1916.

[17] LIU S, QI L, QIN H F, et al. Path aggregation network for instance segmentation [C]//*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 8759-8768.

[18] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition [DB/OL]. (2014-09-04). https://arxiv.org/abs/1409.1556

[19] HU J, SHEN L, ALBANIE S, et al. Squeeze-and-excitation networks [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, **42**(8): 2011-2023.

[20] LIN T Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020, **42**(2): 318-327.

[21] The UA-DETRAC dataset [EB/OL]. [2023-08-03]. https://detrac-db.rit.albany.edu/

[22] LIN T Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: Common objects in context [M]//Computer vision – ECCV 2014. Cham: Springer, 2014: 740-755.

[23] EVERINGHAM M, VAN GOOL L, WILLIAMS C K I, et al. The pascal visual object classes (VOC) challenge [J]. *International Journal of Computer Vision*, 2010, **88**(2): 303-338.

[24] HOWARD A G, ZHU M L, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications [DB/OL]. (2017-04-17). http://arxiv.org/abs/1704.04861

[25] SANDLER M, HOWARD A, ZHU M L, et al. MobileNetV2: Inverted residuals and linear bottlenecks [C]//*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City: IEEE, 2018: 4510-4520.

[26] HOWARD A, SANDLER M, CHEN B, et al. Searching for MobileNetV3 [C]//*2019 IEEE/CVF International Conference on Computer Vision*. Seoul: IEEE, 2019: 1314-1324.

[27] HAN K, WANG Y H, TIAN Q, et al. GhostNet: More features from cheap operations [C]//*2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle: IEEE, 2020: 1577-1586.

[28] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C]//*2016 IEEE Conference on Computer Vision and Pattern Recognition*. Las Vegas: IEEE, 2016: 770-778.

[29] HUANG G, LIU Z, VAN DER MAATEN L, et al. Densely connected convolutional networks [C]//*2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu: IEEE, 2017: 2261-2269.

[30] WOO S, PARK J, LEE J Y, et al. CBAM: Convolutional block attention module [M]//Computer vision – ECCV 2018. Cham: Springer, 2018: 3-19.

[31] WANG Q L, WU B G, ZHU P F, et al. ECA-net: Efficient channel attention for deep convolutional neural networks [C]//*2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Seattle: IEEE, 2020: 11531-11539.

[32] WANG C Y, BOCHKOVSKIY A, LIAO H Y M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors [C]//*2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Vancouver: IEEE, 2023: 7464-7475.

[33] LIU W, ANGUELOV D, ERHAN D, et al. SSD: Single shot multibox detector [M]//Computer vision – ECCV 2016. Cham: Springer, 2016: 21-37.

[34] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16×16 words: Transformers for image recognition at scale [DB/OL]. (2020-10-22). https://arxiv.org/abs/2010.11929

[35] LIU Z, LIN Y T, CAO Y, et al. Swin transformer: Hierarchical vision transformer using shifted windows [C]//*2021 IEEE/CVF International Conference on Computer Vision*. Montreal: IEEE, 2021: 9992-10002.

[36] LI Y H, YAO T, PAN Y W, et al. Contextual transformer networks for visual recognition [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023, **45**(2): 1489-1500.