Multimedia Tools and Applications (2020) 79:18201–18219
https://doi.org/10.1007/s11042-020-08722-y

Real-time traffic sign detection and classification towards real traffic scene

Yiqiang Wu[1,2] · Zhiyong Li[1,2] · Ying Chen[1,2] · Ke Nai[1,2] · Jin Yuan[1,2]

Received: 7 May 2019 / Revised: 4 December 2019 / Accepted: 31 January 2020 /
Published online: 2 March 2020

Abstract

In this paper we propose a real-time traffic sign recognition algorithm which is robust to the small-sized objects and can identify all traffic sign categories. Specifically, we present a two-level detection framework which consists of the region proposal module(RPM) which is responsible for locating the objects and the classification module(CM) which aims to classify the located objects. In addition, to solve the problem of insufficient samples, we present an effective data augmentation method based on traffic sign logo to generate enough training data. The experiments are conducted in TT100k, and the results show the superiority of our method.

Keywords Traffic sign recognition · Small object detection · Data augmentation · Image synthesis

1 Introduction

Traffic sign recognition is an important sub-task in the advanced driver assistant systems and autonomous driving systems. In general, there are two stages in a traffic sign recognition system: finding the locations of the traffic signs in real traffic scenes (traffic sign detection) and classifying the detected traffic signs into their specific sub-classes (traffic sign classification). Traffic sign detection faces many difficulties in real traffic scenes due to illumination changes, partial occlusion, cluttered background, and small size, as shown in Fig. 1. And in the second stage, there are also many problems, such as missing samples, unbalanced sample categories, the rare sample numbers etc. TT100k [37] is a Chinese traffic sign dataset and contains over 150 categories, which is the most widely covered traffic sign dataset in the current. However, many rather rare traffic signs are still not included. And not only that, the existing categories in the dataset are extremely imbalanced, let alone

✉ Zhiyong Li
zhiyong.li@hnu.edu.cn

[1] College of Computer Science and Electronic Engineering Hunan University, Changsha, China

[2] Key Laboratory for Embedded and Network Computing of Hunan Province, Changsha, China

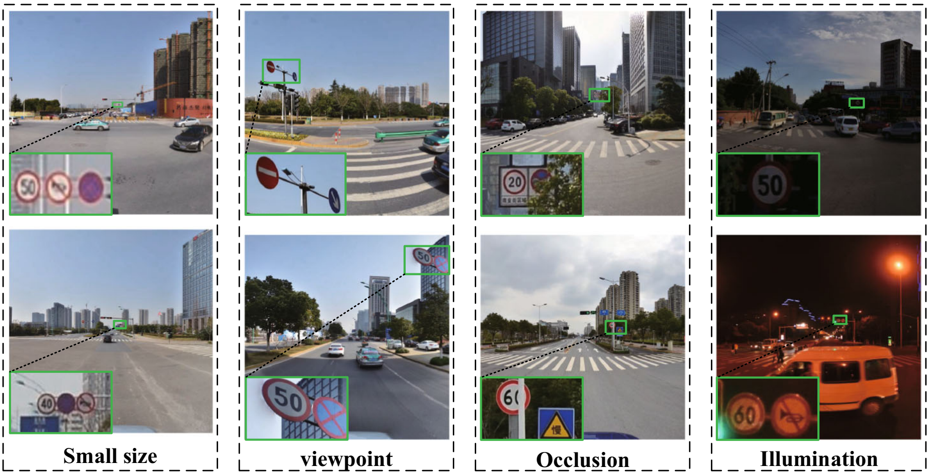

Fig. 1 The difficulties in traffic sign detection. In real traffic scenes, traffic sign detection faces many difficulties including small size, viewpoint, occlusion, illumination and so on

the majorities are less than 100 which is obviously not enough for training the detection model, as shown in Fig. 2.

On the one hand, current object detection methods [12, 21, 25–28] are either not robust to small-sized traffic signs or difficult to meet the real-time performance. On the other hand, limited by traffic sign data sets, most approaches can only classify several super-classes such as just 3 classes of *Mandatory*, *Danger* and *Prohibitory* [34] or a limited number of sub-classes such as 45 classes [20, 22, 37]. In this paper, oriented to the real traffic scene, we proposed a novel two-level detection architecture to address the aforementioned challenges. The contributions of this paper are summarized as follows:

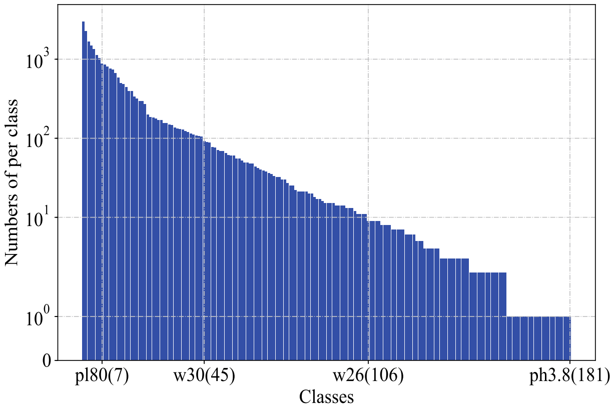

Fig. 2 The number of the traffic signs in TT100k. The number of the existing samples is extremely unbalanced, the number of categories starting with *w30* is less than 100. the number of categories starting with *w26* is less than 10

1) We propose a two-level detection architecture to deal with the problem of missing samples and imbalanced samples. We present a revised YOLOv3 network for traffic sign detection and improve the performance of small objects detection.

2) We present an effective data augmentation method based on traffic sign logo to generate enough training data and achieve unlimited traffic sign recognition. The reasonable experiment is designed to prove the effectiveness of the method.

3) The approach achieves good trade-offs in terms of completeness, real-time and accuracy, it can be applied widely to many fields such as advanced driver assistant systems and autonomous driving systems.

2 Related work

2.1 Traffic sign detection

Traditional traffic sign detection approaches contain a wide variety of algorithms and thoughts [23, 24]. Escalera et al. [19] use color and shape features to detect road traffic signs, while Garcia-Garrido et al. [36] employ the Hough transform to get the information from the edges in the image. To improve the detection speed, Bahlmann et al. [1] detect traffic signs by using a set of Haar wavelet features obtained from AdaBoost training. Salti et al. [29] use HOG features and SVM classifier to detect traffic signs. Recently, Berkaya et al. [3] extended this approach by combining features including HOG, local binary patterns (LBP) and Gabor features within a SVM classification framework.

In addition to the traditional methods, CNN-based methods have developed rapidly in recent years. Jin et al. [18] use a hinge loss stochastic gradient descent (HLSGD) method to train a detection network. Zhu et al. [36] employ fully convolution network (FCN) to guide traffic sign proposals and deep convolutional neural network (CNN) for object classification. Meng et al. [22] detect traffic signs based on SSD by using image pyramid. Li et al. [20] improve the performance of small object detection by using Generative Adversarial Networks(GAN). All these approaches are able to detect some traffic signs. However, limited by data sets and the small size of the traffic signs, these methods are difficult to make good trade-offs in terms of completeness, real-time and accuracy. To achieve unlimited traffic sign detection, some researchers put their attention to data augmentation due to the specific templates of the traffic signs. In reference [4], the authors present a pipeline-based approach to image augmentation including z-stack augmentation, randomized elastic distortions etc. And these image augmentation methods are open to the public. Zhun et al. [35] propose Random Erasing methods to randomly select a rectangle region in an image and erase its pixels with random values, which can improve the robustness of the model against occlusion and prevent the occurrence of over-fitting to some extent. Similar to [35], in [9], the authors show an approach of randomly masking out square regions of input images and prove the effectiveness of the method by sufficient experiments.

2.2 Object detection

As early as 2001, Viola-Jones et al. [33] used the strategy of sliding window and multi-scale haar feature to realize real-time face detection in the first time. In 2005, Hog feature [8] was proposed to detect pedestrians and achieved robust results. In 2008, the DPM method

[10] was proposed and achieved the best results at the time. Before the deep learning, the traditional object detection methods are roughly divided into three parts: regional selection (sliding window, ROI, etc.), feature extraction (SIFT, HOG, etc.) and classifier (SVM, Adaboost, etc.). However, the traditional object detection algorithms have many shortcomings. On the one hand, the sliding window selection strategy is time-consuming and redundant, on the other hand, the hand-designed features are less robust.

Recently, the CNN-based approaches have achieved great success in many fields [5, 6]. Generally speaking, the CNN-based methods in the field of object detection can be divided into two categories: Region-Proposed methods and End-to-End methods. Among Region-Proposed methods, the Overfeat [30] is an early work using CNNs to do object detection . The main idea is to use multi-scale sliding windows for classification, positioning and detection. Unlike the Overfeat, which uses sliding windows for region-propose, R-CNN [13] uses selective search[1] to propose ROI region and makes final predictions using an SVM. Due to different size of input images, SPP-Net [15] introduces a Spatial Pyramid Pooling Layer (SPP) to reduce the adverse effects of deformation and cutting. Fast R-CNN [12] enables the network to be trained end-to-end. Consider the huge time cost of selective search in Fast R-CNN, Faster R-CNN [28] proposes a region proposal CNN and integrates it with Fast-RCNN by sharing convolutional layers, which further improves the object detection performance in terms of detection speed and accuracy. In R-FCN [7], the final fully connection layer is replaced by a location-sensitive convolutional network, which greatly improves the detection rate while maintaining high positioning accuracy.

Among End-to-End methods, YOLO [27] can be viewed as an originator which uses a fully connection layer to directly produce the class of object as well as the bounding box. Subsequently, the single shot multi-box detector(SSD) [21] introduces the default boxes inspired by [28] and multi-scale feature mapping layers to raise the precision of the bounding box. DSOD [31] designs an efficient framework and a set of principles to learn object detectors from scratch, following the network structure of SSD. Based on [27], YOLO9000 [25] and YOLOv3 [26] further improve the performance and speed by a large margin by merging many tricks including multi-scale training, anchor boxes, new classification network design and so on.

3 Base network

Our method is based on the YOLOv3 structure. Compared with YOLO and YOLO9000, YOLOv3 uses a new feature extracting network with some shortcut connections: darknet53, and does predictions across 3 scales. Instead of using softmax loss, the authors use logistic loss to deal with more complex domains like the Open Images Dataset.

3.1 Network structure

The YOLOv3 network is organized as follows: the classifier network darknet53 with many Residual blocks achieves similar performance to ResNet-152 but 2× speed. The darknet53 executes down-sampling operation using a convolution layer with stride 2 instead of using pooling layer. After each down-sampling operation, the Residual block is employed for in that scale. The darknet53 contains 5 down-sampling operations and results in 32-fold scale-zooming. Based on the darknet53, YOLOv3 adds 3 prediction blocks from different scales respectively. Specifically, for each prediction block, the up-sampling operation and router operation are employed to get higher resolution feature maps. The YOLOv3 structure is illustrated in the Fig. 3.

3.2 Loss function

The YOLOv3 predicts 3 boxes at every scale and outputs the tensors with the $N \times N \times [3 \times (4 + 1 + C)]$ dimensions for the 4 bounding boxes offsets, 1 objectness prediction and C classes predictions. The authors use k-means clustering to acquire 9 preset bounding boxes and every scale contains 3 preset bounding boxes. The loss function is as follows:

$$
\begin{aligned}
loss = {} & \lambda_{coord}[\sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{obj}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{obj}[(\sqrt{w}_i - \sqrt{\hat{x}}_i)^2 + (\hat{h}_i - \hat{\hat{y}}_i)^2]] \\
& + \lambda_{obj}\sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{obj}(C_i - \hat{C}_i)^2 \\
& + \lambda_{obj}\sum_{i=1}^{s^2} 1_{i}^{obj}\sum_{c}(p_i(c) - \hat{p}_i(c))^2 \\
& + \lambda_{noobj}\sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{noobj}(C_i - \hat{C}_i)^2 \qquad (1)
\end{aligned}
$$

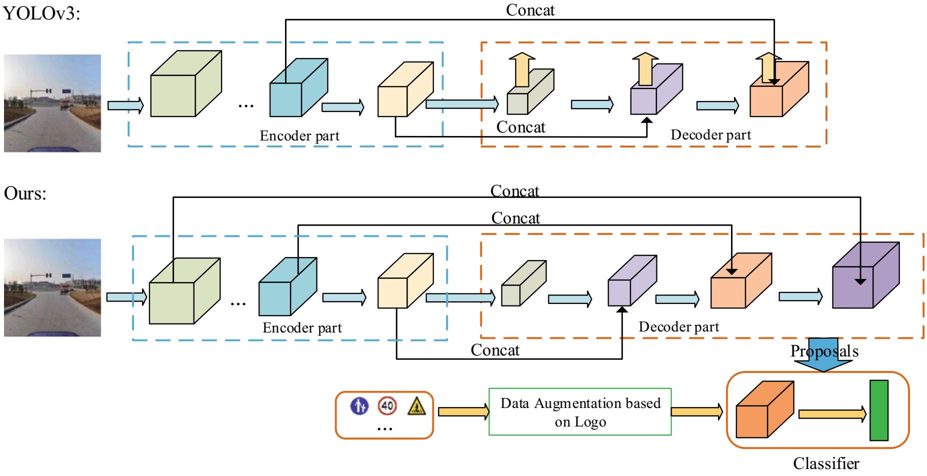

Fig. 3 The architecture of our method. The RPM consists of the feature extraction network and the location prediction network. The CM accepts the proposals from the RPM for further classification. In this work, we select darknet19 as our classifier network

where $1_i^{obj} \in \{0, 1\}$ denotes that the grid cell i which is responsible for predicting the object. $1_{ij}^{obj} \in \{0, 1\}$ represents the jth default bounding box of the grid cell i. $1_{ij}^{noobj} \in \{0, 1\}$ denotes the jth default bounding box of the grid cell which is not responsible for any objects. $p_i(c)$ is a 1-D vector indicating the classes of the object. s^2 denotes the area of the feature maps. k denotes the number of preset bounding boxes, which is equal to 3 in YOLOv3.

4 Our methodology

Our goal is to design an ultra-efficient traffic-sign object detection network towards real scenes, therefore in this work we mainly focus on the small object detection and unlimited traffic sign classification while ensuring the real-time performance. However, the general detection framework does not work since many classes are not included in the dataset. To solve this problem we proposed a novel two-level network architecture, which is formed by two modules, i.e., the region proposal module(RPM) and the classifier module(CM). The RPM aims to regress the locations of the object whereas the CM aims to predict multi-class labels based on the locations, as shown in Fig. 3.

4.1 Two-level detection architecture

For the unlimited traffic sign detection, we design a two-level detection architecture, as shown in Fig. 3. In the first stage, we focus on regressing the locations of the traffic sign by using the RPM. And in the second stage, we target at classifying the specific categories of the traffic signs by the CM. We view all traffic signs as one class and use the RPM to propose the location of the traffic signs, then we add an extra classifier module to acquire the labels of the objects.

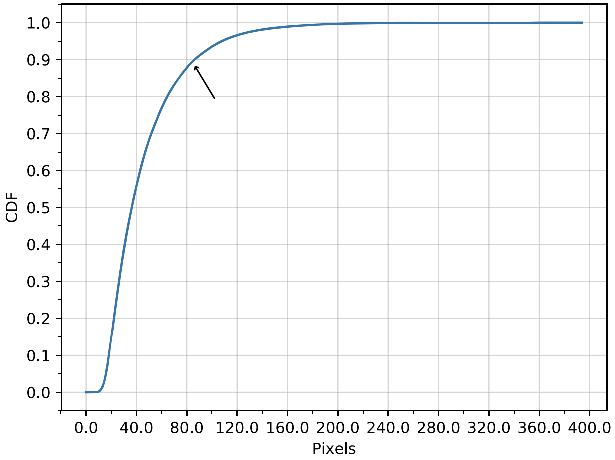

Fig. 4 The Cumulative Distribution Function(CDF) of the size of the traffic signs in TT100k. The sizes of near 90% boxes are less than 80 pixels, which is rather than small compared with the original 2048 pixels

4.2 Improved YOLOv3 for the RPM

The RPM is based on the YOLOv3. Although YOLOv3 achieves better performance in small object detection by doing predictions across scales, there still are much room for improvement. For traffic signs in TT100k, the sizes of near 90% boxes are less than 80 pixels while the size of the whole image is 2048 pixels, as shown in Fig. 4.

Our solutions are inspired by the facts: the larger feature maps tend to encode more location information, which is proven crucial [2, 11] for small object detection. In our network, features are extracted from the tail of each stage in the encoder part. To make use of the low-level features, one straight-forward approach is to add more low-level layers to the decoder part. As shown in Table 1, YOLOv3 [26] adds the layers of {36,67,83} to the decoder network. In this work, we reassign the layers of {11,36,67,83} to the decoder network to acquire 4 scales for region proposals and adjust the number of channels to ensure the same overall computational complexity. The experiment shows that this simple skill can make the performance of detecting small and medium objects increases dramatically while keeping the same detection speed, shown in Table 4, which implies the low-level and large feature maps can provide more location information for the small and medium objects.

In addition, the RPM just aims to acquire the proposals of the object without need for the specific categories of the object, therefore we modify the loss function by getting rid of the section of classifying loss. We rewrite the loss function as follows:

$$
\begin{aligned}
loss = {} & \lambda_{coord}[\sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{obj}[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{obj}[(\sqrt{w}_i - \sqrt{\hat{x}}_i)^2 + (\hat{h}_i - \hat{\hat{y}}_i)^2]] \\
& + \lambda_{obj}\sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{obj}(C_i - \hat{C}_i)^2 \\
& + \lambda_{noobj}\sum_{i=1}^{s^2}\sum_{j=1}^{k} 1_{ij}^{noobj}(C_i - \hat{C}_i)^2
\end{aligned}
\tag{2}
$$

4.3 Data augmentation based on logo for the RM

There are many datasets for traffic sign detection and classification, such as TT100k [37], GTSDB [17], GTSRB [32] and so on. However, the traffic signs distributes seriously unevenly no matter which country it is. For example, in TT100k, *No Parking*, *No Entry* and *Speed Limit* signs are very common with the numbers of more than 1000, while the signs such as *Mountain danger*, *Falling rocks* are very rare. The current situation of non-uniform distribution of traffic signs makes the recognition of traffic signs a difficult problem.

Through the careful observation of the existing data sets, we find that the differences in the same traffic signs are mainly reflected in the following aspects: viewpoint, background, size, color, illumination, contrast, occlusion, and pollution. To get enough and evenly distributed images and simulate as many situations as possible in the real world, a specific data

Table 1 Our revised network based on YOLOv3, the input resolution is set to 512*512

	Encoder part				Decoder part			
YOLOV3	Conv1 block	Conv2 block	Conv3 block	Conv4 block	Conv5 block	Deconv1 block	Deconv2 block	
	256*256	128*128	64*64	32*32	16*16	32*32	64*64	–
Our network	Conv1 block	Conv2 block	Conv3 block	Conv4 block	Conv5 block	Deconv1 block	Deconv2 block	Dconv3 block
	256*256	128*128	64*64	32*32	16*16	32*32	64*64	128*128

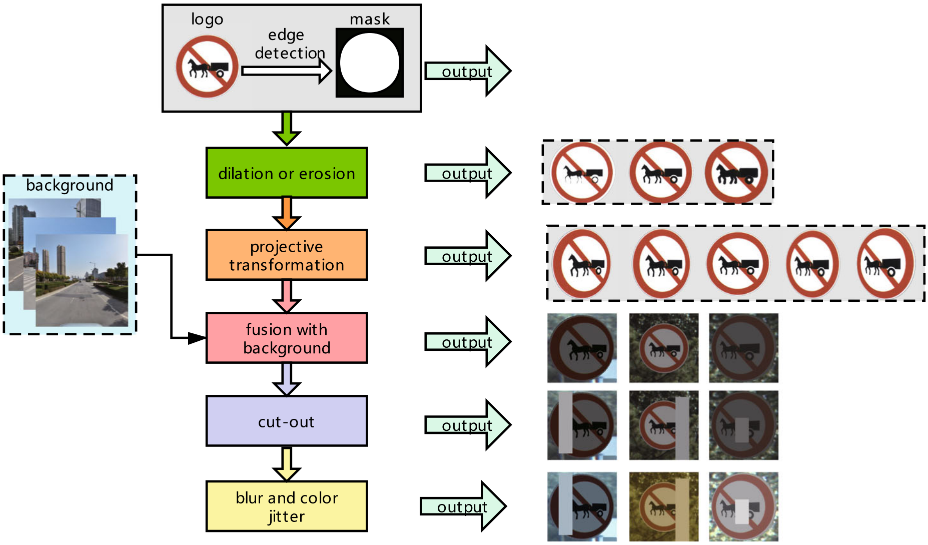

Fig. 5 The pipeline of the data augmentation based on logo

augmentation technology based on traffic sign logo is proposed. The pipeline of the data augmentation based on logo is shown in Fig. 5.

Firstly, the logo of traffic sign is acquired mutually and the mask of the sign is formed automatically by Canny operator. To simulate the change of viewpoint in the real scene, the perspective transformation is applied to the traffic sign logo. A 2-d image is transformed into another plane through perspective transformation. This process can be expressed as:

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} .x' = \frac{a}{c}, y' = \frac{b}{c}$$

M_{ij} is the transformation matrix including 9 parameters,(x, y) is the coordinates before the transformation, (x', y') is the coordinates after transformation.

In real traffic scenarios, the shapes of traffic signs vary in thickness due to the influence of illumination, contamination, manufacturing technology and so on. The erosion and dilation are the primary morphological operation of the image. These two operations have completely opposite effects, which can further increase the variations of the samples.

The next step is the fusion of the sign and the background. We select randomly an image from the real traffic scenes as the background, and crop out a sub-image with the same size of the sign. Then the fusion operation of the sign and the sub-image can described as follows:

$$I_o = mask \odot I_s + (1 - mask) \odot I_b \tag{3}$$

where I_s denotes the traffic sign logo, $mask$ denotes the mask of the sign and I_b denotes the image cropped out from the real traffic scenes.

Table 2 The performance of our classifiers trained by our generated data

Classifier	VGG-16	ResNet50	darknet19
Avg(%)	86.4	91.2	91.3

Finally, color jitter is employed to the fused image. In addition, we add Gaussian noise with different kernel size including $\{1 \times 1, 3 \times 3, 5 \times 5, 7 \times 7\}$ to the image for the blurring and the cut-out [9, 35] is applied to the synthetic image which enables the network to have more generalization performance. The reasonable experiments are conducted to valid the effectiveness of our data augmentation method, shown in Table 2. Some synthetic images are shown in Fig. 6.

5 Experiment

In this section, we compare the performance in terms of accuracy and speed to other approaches in traffic sign detection in TT100k [37] and GTSDB [17]. TT100k is a Chinese traffic sign data set, which is composed of 9176 images containing 143 classes of traffic signs. The resolution of the image is 2048 × 2048 pixels. The images are collected under real world conditions with large illumination variations and weather differences. A typical traffic sign is about 80 × 80 pixels in a 2048 × 2048 pixels image, or just 0.2% area of the image. GTSDB is a German traffic sign data set, which contains 3 categories namely *mandatory*, *prohibitory* and *danger*. GTSDB data set is split into a training set of 600 images and a test set of 300 images and covers natural traffic scenes of various roads (road, rural, urban) recorded during the day and dusk. Due to GTSDB data set just contains 3 classes, we only evaluate the performance of our RPM on this data set.

We design our experiments in 3 parts. In the first part, we conduct the experiment to verify the effectiveness of our data augmentation based on logo. In the second section, the performance of our revised network is compared to the base network YOLOv3. In the final part, we compare our approach to other state-of-the-art methods in terms of speed, accuracy and the number of recognizable traffic signs. All experiments are conducted in a same workstation with Intel Core i7-6700 3.4GHz CPU and a single GeForce GTX1080 Ti Graphics. And the operating system is ubuntu 16.04 LTS. We use the Darknet neural network framework for training and testing.

5.1 Implementation details

For our Region Proposal Module (RPM), k-means cluster method is firstly employed to generate the default anchor boxes. And we use the generated anchor boxes to replace the

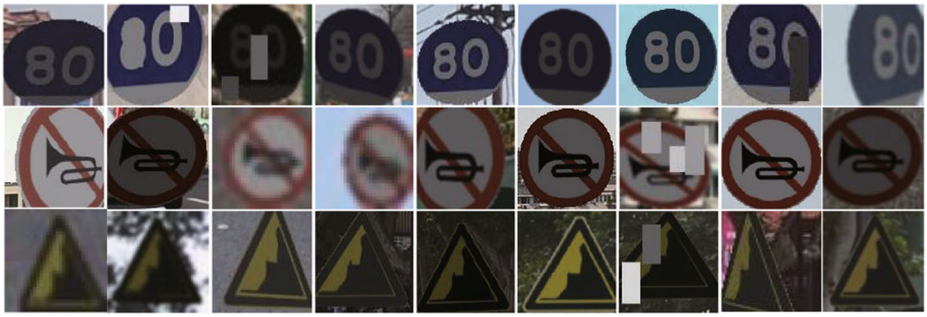

Fig. 6 Some synthetic images

original ones in cfg file. We set the parameters of learning rate, max batches, momentum and decay to 0.0005, 100000, 0.9, 0.001 respectively. And then we start the training process and keep an eye on training losses.

For our Classification Module (CM), first, our data augmentation method is used to generate the training data. And then the parameters of batch, subdivisions, height, width in the cfg file are set to 1280, 2, 72, 72 respectively, other configures remain unchanged. Because the training of classifiers in the original Darknet framework turns on random clipping by default, and random clipping will seriously affect the appearance of traffic signs, so we modify the code to close it.

After getting the detector model and the classifier model, we combine these two models to make predictions in a pipeline, one for predicting the location of the object, and another for classifying the object. For the detection model, we set a relatively low threshold of 0.2 to ensure high recall rate. The threshold of nms is set to 0.35. And for the classifier model, we set a relatively high threshold of 0.75 to ensure high accuracy.

5.2 The experiments of the CM

5.2.1 The effectiveness analysis of the data augmentation based on logo

In this section, we will conduct the experiment to evaluate the performance of our data augmentation method based on traffic sign logo. The signs in TT100k are cropped to form our testing data. The formed testing data includes 143 categories and 19546 images and the *io*, *wo*, *po* are deprecated. The training data are generated by our data augmentation method. Specifically, we manually produce 143 standard logos, and then generate 5,000 synthetic images for each logo through our data augmentation method. Eventually, the training data contains 143 categories and 715000 images. Some synthetic images are shown in Fig. 6. We train our 3 different classifiers by feeding our training data and test in our testing data. The test results are shown in Table 2. For darknet19, ResNet50 and VGG-16, they achieve 91.3%, 91.2% and 86.4% average precision separately in our testing data set. The model trained by the data generated by our data augmentation method still has good robustness and generalization ability for traffic signs in real scenes and it can be used as a baseline for future comparison in data augmentation of the traffic signs. Overall, our studies thus offer a new strategy to treat the unbalanced samples problem in traffic sign classification.

5.2.2 Ablation study of the data augmentation based on logo

To verify the effectiveness of each component of our data augmentation method for traffic signs, we conduct several experiments by removing the specific components of the method.

Table 3 The ablation study of the data augmentation method for traffic signs

Fusion	√	√	√	√	√	√	√	√
Perspective transformation	√		√	√			√	
Erosion and dilation	√	√		√		√		
Cut-out	√	√	√		√			
Avg(%)	91.3	84.2	89.9	90.2	79.3	80.3	87.8	77.2

Table 4 The comparison on the TT100K data set

Methods	mean time(ms)	classes	mAP(%)
zhu.al	4081	45	81.6
Faster RCNN	231	45	73.4
YOLOv3	80	45	71.24
ours(1024*1024)	86	45	**82.6**
ours(1024*1024)	86	**200**	74.28
ours(512*512)	**24**	45	79.38
ours(512*512)	**24**	**200**	70.91

Bold fonts indicate best results

Specifically, several additional experiments are designed, in which perspective transformation, erosion and dilation and cut-out were removed respectively from the method. For a fair comparison, we use the exact same network structure, training images and super-parameters (such as epochs, learning rate, etc.). Specifically, The darknet19 is used as our training network with the learning rate of 0.1 and the 300 epochs. We select the final weights as our testing model. The results in Table 3 suggest that these components are more or less helpful to the model. Although some important data augmentation methods are revealed by the paper, whether there are other means of data augmentation is an open question.

5.3 The experiments of the RPM

In this section, we conduct the experiments to compare our revised network with the base network YOLOv3 in TT100k data set and GTSDB data set.

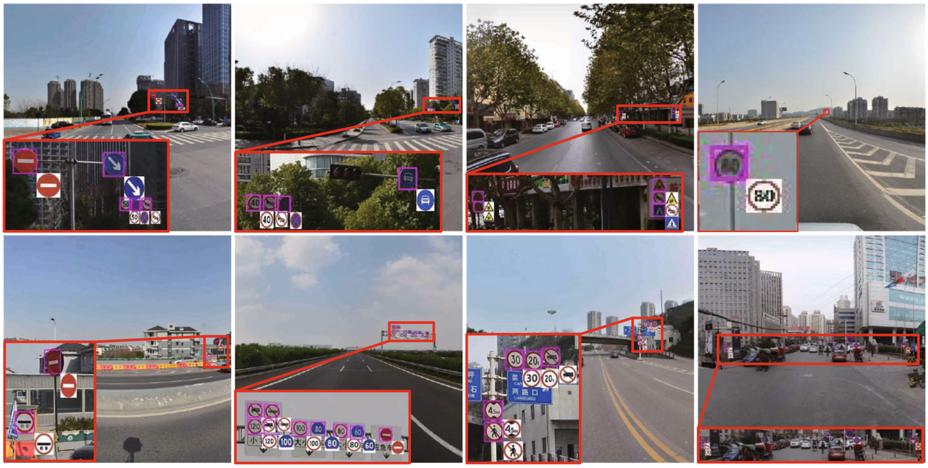

Fig. 7 Some detection results when the input resolution is 512*512. The label appears as an image in the bottom right corner of the box

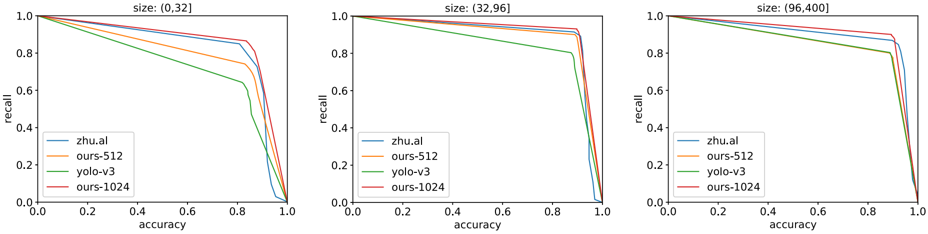

Fig. 8 The detection results of zhu.al [37], YOLOv3(512) [26] and ours(512,1024)

5.3.1 The results in TT100k

Firstly, we use k-means cluster method to generate the default anchor boxes. In YOLOv3, the anchor boxes are set to {4,4, 6,7, 6,12, 9,9, 11,22, 13,13, 18,19, 25,28, 42,42} and in our revised network, the anchor boxes are {3,4, 5,6, 6,11, 7,7, 8,9, 11,12, 11,22, 14,14, 17,18, 22,24, 30,32, 46,45}. The input resolution are 512 × 512. Most of our training strategies follow YOLOv3, including multi-scale training, data augmentation, aspect ratios, learning rate and so on. As shown in Table 4, the average detection time of YOLOv3 is 80 ms and it achieves 71.24% mAP, while our network achieves 79.38% mAP with 86 ms time-spending. For a further comparison, following [37], the AUC curves are drawn in 3 different sizes, that are (0,32], (32,96] and (96,400], as shown in Figs. 7 and 8. Compared with the base network, our revised network mainly improves the performance in the size of (0,32] and (32,96].

5.3.2 The results in GTSDB

Similar to the above steps, the k-means cluster method is used to produce the default anchor boxes. In YOLOv3, the generated anchors are {7,12, 8,14, 10,17, 12,20, 15,24, 17,30, 22,37, 28,47, 40,67}. And in our revised network, they are {6,8, 7,9, 8,11, 9,12, 10,13, 11,14, 12,16, 15,19, 17,24, 22,29, 29,37, 40,52}. As shown in Table 5, YOLOv3 achieves the mAP value with 0.89 and speed with 78 ms, However, our revised network obtains better result with 0.93 mAP value and 83 ms speed. Except for that, from the IoU point of view, compared with YOLOv3, our network achieves more accurate positioning with IoU value of 0.845. This mainly benefits from our larger feature maps and more anchor boxes.

5.4 The experiments of the whole framework

5.4.1 Overall performance comparison to other methods

In this section, we compare our whole algorithm consisting of RPM and CM to other state-of-the-art methods. Figure 8 provides the comparison of our approach with other

Table 5 The comparison on the GTSDB data set, aIoU denotes the average of IoU values of true positive bounding boxes

Methods	mean time(ms)	aIoU(%)	mAP(%)
zhu.al	483	80.36	89.56
Faster RCNN	208	82.93	91.51
YOLOv3	**78**	79.56	89.51
ours	83	**84.52**	**93.60**

Bold fonts indicate best results

Table 6 The analysis of the effectiveness of the RPM and CM. All the input images are resized to the same size of 512 × 512 in all variants

Methods	mean time(ms)	mAP(%)
YOLOv3 without CM	**19**	69.83
YOLOv3 with CM	21	75.21
RPM without CM	21	73.35
RPM with CM	24	**79.38**

Bold fonts indicate best results

state-of-the-art methods in terms of the recall and accuracy in TT100k. It can be observed that our proposed approach obtains comparable results to the previous state-of-the-art methods [26, 37]. Specifically, on the one hand, compared to other methods of limited classes detection, our proposed method achieves the detection of all traffic sign classes due to our two-level detection architecture and the data augmentation based on logo, as shown in Tables 6 and 7. On the other hand, as shown in Table 4, when we set the input sizes of the network to 512 × 512, our method can also achieve the comparable results with the mAP of 79.38% and the speed of 41.67 FPS, which make a great trade-off between the accuracy and the speed. When the input sizes of the network are set to 1024 × 1024, we achieve better results with the mAP of 82.6% and 11.6 FPS, which outperforms other state-of-the-arts in terms of the detection accuracy and speed. In [37], the resolutions of the input images are 2048 × 2048, which far exceeds our input sizes. Some detection results are shown in the Fig. 7.

5.4.2 Ablation studies

To better verify the effectiveness of the RPM and CM in our method, we construct four variants and evaluate them on TT100k, shown in Table 6. For a pair comparison, we resize the input images to the same size of 512 × 512 in all variants. The CM is trained to classify 45 traffic signs using the generated data. The methods without CM imply just original data is used to train the model end by end. We compare the results by mAP value and speed. As shown in Table 6, YOLOv3 with CM achieves great improvement in terms of mAP value of 0.85 with a slightly slower speed. And RPM with CM further improves the performances with a mAP value of 79.38 and the speed of 24 ms. This can be explained by the facts that RPM can improve the detection accuracy of small objects and CM can achieve more accurate classification of rare categories.

6 Conclusion

In this paper, we present a novel two-level detection architecture, which is composed of the location modules(RPM) and the classification module(CM). The RPM aims to locate the objects and then the CM targets to get the specific labels of the objects. We revised the YOLOv3 network to locate the small size object more precisely. To solve the problem of missing categories, a data augmentation method based on logo is present and a reasonable experiment is designed to prove the effectiveness of the method. In the future, we plan to employ the two-level detection architecture to other specific kinds of objects e.g. traffic light etc. In addition, in the detection task of traffic signs, a wrong detection result or classification result may suddenly appear. However, we cannot effectively explain this phenomenon.

Table 7 The detection results of per class in TT100k, ’-’ means the detection results don’t contain this category

Classes	i1	i10	i11	i12	i13	i14	i15	i2	i3	i4	i5	il100	il110	il50	il60	il70	il80	il90	ip	p10	p11	p12	p13	p14	p15
Zhu.al(%)	-	-	-	-	-	-	-	77.68	-	88.04	93.04	-	50.00	-	86.97	-	85.16	-	82.31	78.72	87.47	86.50	-	-	-
ours(%)	25.00	99.58	0.00	100.0	56.67	100.0	33.33	41.46	45.00	62.00	83.91	89.74	100.0	100.0	79.76	100.0	92.80	100.0	81.13	75.26	80.00	90.36	17.05	94.79	100.0
Classes	p16	p17	p18	p19	p2	p22	p23	p25	p26	p27	p28	p3	p4	p5	p6	p8	p9	pa13	pa14	pb	pg	ph2	ph2.2	ph2.4	ph2.5
Zhu.al(%)	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ours(%)	97.62	80.56	15.60	95.43	68.33	30.00	84.16	91.67	85.58	74.45	16.67	86.83	100.0	92.66	82.11	57.67	84.58	96.12	100.0	75.75	94.34	0.00	100.0	100.0	41.67
Classes	ph3	ph3.2	ph3.5	ph3.8	ph4	ph4.2	ph4.3	ph4.5	ph4.8	ph5	ph5.5	pl10	pl100	pl110	pl120	pl15	pl20	pl25	pl30	pl35	pl40	pl5	pl50	pl60	pl70
Zhu.al(%)	-	-	-	-	71.92	-	-	76.53	-	66.25	-	-	92.48	-	93.92	-	74.53	-	86.86	-	90.23	-	89.38	79.44	87.22
ours(%)	97.62	100.0	62.50	100.0	64.99	77.78	100.0	92.84	100.0	76.22	50.00	79.49	89.67	93.33	89.66	96.66	76.46	100.0	68.64	100.0	90.17	86.55	83.24	91.14	81.41
Classes	pl80	pl90	pm10	pm15	pm20	pm30	pm35	pm40	pm50	pm55	pm8	pn	pne	pr20	pr30	pr40	pr45	pr50	pr60	pr70	pr80	ps	pw3	pw3.2	pw3.5
Zhu.al(%)	87.39	-	-	-	83.96	87.62	-	-	-	79.99	-	89.75	91.16	-	-	87.16	-	-	-	-	-	-	-	-	-
ours(%)	95.16	86.67	100.0	60.0	85.01	55.74	0.00	100.0	61.11	96.22	0.00	76.52	93.78	100.0	50.00	90.39	100.0	100.0	99.09	100.0	100.0	55.77	50.00	60.00	100.0
Classes	pw4	w12	w13	w15	w16	w18	w2	w20	w21	w22	w24	w28	w3	w30	w32	w34	w35	w42	w43	w45	w46	w47	w55	w56	w57
Zhu.al(%)	-	-	77.69	-	-	-	-	-	-	-	-	-	-	-	54.56	-	-	-	-	-	-	-	-	-	-
ours(%)	100.0	100.0	99.80	0.00	0.00	47.88	0.00	100.0	88.89	76.79	66.67	100.0	100.0	92.00	82.05	20.00	100.0	100.0	100.0	85.42	91.67	65.00	92.01	14.29	91.78
Classes	w58	w59	w63																						
Zhu.al(%)	-	-	-																						
ours(%)	70.99	85.96	65.08																						

Therefore, in the future we will explore the upcoming field of explainable AI, e.g. why there is an error classification of a traffic sign. For those truly interested in this field, more details can be found in [14, 16].

Acknowledgements This work was partially supported by the National Natural Science Foundation of China (No.61672215), National Key R&D Program of China (No. 2018YFB1308604) and Hunan Science and Technology Innovation Project (No. 2017XK2102).

References

1. Bahlmann C, Zhu Y, Ramesh V, Pellkofer M, Koehler T (2005) A system for traffic sign detection, tracking, and recognition using color, shape, and motion information. IEEE Intell Vehicles Symposium: 255–260
2. Bell S, Zitnick CL, Bala K, Girshick RB (2016) Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks. Compu Vis Pattern Recogn: 2874–2883
3. Berkaya SK, Gunduz H, Ozsen O, Akinlar C, Gunal S (2016) On circular traffic sign detection and recognition. Expert Sys Appl 48:67–75
4. Bloice M, Roth PM, Holzinger A (2019) Biomedical image augmentation using augmentor. Bioinformatics 35(21):4522–4524
5. Chen W, An J, Li R, Fu L, Xie G, Bhuiyan ZA, Li K (2018) A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features. Futur Gener Comput Syst 89:78–88
6. Chen Y, Yuan J, Li Z, Wu Y, Nouioua M, Xie G (2019) Person re-identification based on re-ranking with expanded k-reciprocal nearest neighbors. J Vis Commun Image Represent 58:486–494
7. Dai J, Li Y, He K, Sun J (2016) R-fcn: object detection via region-based fully convolutional networks. Neural Inform Process Sys: 379–387
8. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. Comput Vis Pattern Recogn 1:886–893
9. Devries T, Taylor GW (2017) Improved regularization of convolutional neural networks with cutout. Comput Vis Pattern Recogn
10. Felzenszwalb PF, Girshick RB, Mcallester DA, Ramanan D (2010) Object detection with discriminatively trained part-based models. IEEE Trans Pattern Anal Mach Intell 32(9):1627–1645
11. Gidaris S, Komodakis N (2015) Object detection via a multi-region and semantic segmentation-aware cnn model. Int Conf Comput Vis: 1134–1142
12. Girshick RB (2015) Fast r-cnn. Int Conf Comput Vis: 1440–1448
13. Girshick RB, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. Comput Vis Pattern Recogn: 580–587
14. Goebel R, Chander A, Holzinger K, Lecue F, Akata Z, Stumpf S, Kieseberg P, Holzinger A (2018) Explainable ai: the new 42? In: International cross-domain conference for machine learning and knowledge extraction. Springer, pp 295–303
15. 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 37(9):1904–1916
16. Holzinger A, Kieseberg P, Weippl E, Tjoa AM (2018) Current advances, trends and challenges of machine learning and knowledge extraction: from machine learning to explainable ai. In: International cross-domain conference for machine learning and knowledge extraction. Springer, pp 1–8
17. Houben S, Stallkamp J, Salmen J, Schlipsing M, Igel C (2013) Detection of traffic signs in real-world images: The german traffic sign detection benchmark. Int Joint Conf Neural Netw: 1–8
18. Jin J, Fu K, Zhang C (2014) Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Trans Intell Transp Syst 15(5):1991–2000
19. La Escalera AD, Moreno L, Salichs MA, Armingol JM (1997) Road traffic sign detection and classification. IEEE Trans Ind Electron 44(6):848–859
20. Li J, Liang X, Wei Y, Xu T, Feng J, Yan S (2017) Perceptual generative adversarial networks for small object detection. Comput Vis Pattern Recogn: 1951–1959
21. Liu W, Anguelov D, Erhan D, Szegedy C, Reed SE, Fu C, Berg AC (2016) Ssd: single shot multibox detector. European Conf Comput Vis: 21–37
22. Meng Z, Fan X, Chen X, Chen M, Tong Y (2017) Detecting small signs from large images. Inform Reuse Integrat: 217–224

23. Nai K, Li Z, Li G, Wang S (2018) Robust object tracking via local sparse appearance model. IEEE Trans Image Process 27(10):4958–4970
24. Nai K, Xiao D, Li Z, Jiang S, Gu Y (2019) Multi-pattern correlation tracking. Knowledge Based Systems 104789:181
25. Redmon J, Farhadi A (2017) Yolo9000: better, faster, stronger. Comput Vis Pattern Recogn: 6517–6525
26. Redmon J, Farhadi A (2018) Yolov3: an incremental improvement. Comput Vis Pattern Recogn
27. Redmon J, Divvala SK, Girshick RB, Farhadi A (2016) You only look once: unified, real-time object detection. Comput Vis Pattern Recogn: 779–788
28. Ren S, He K, Girshick RB, Sun J (2017) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149
29. Salti S, Petrelli A, Tombari F, Fioraio N, Stefano LD (2015) Traffic sign detection via interest region extraction. Pattern Recogn 48(4):1039–1049
30. Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, Lecun Y (2014) Overfeat: integrated recognition, localization and detection using convolutional networks. Int Conf Learn Represent
31. Shen Z, Liu Z, Li J, Jiang Y, Chen Y, Xue X (2017) Dsod: learning deeply supervised object detectors from scratch. Int Conf Comput Vis: 1937–1945
32. Stallkamp J, Schlipsing M, Salmen J, Igel C (2012) 2012 special issue: man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. Neural Netw 32:323–332
33. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vis 57(2):137–154
34. Yang Y, Luo H, Xu H, Wu F (2016) Towards real-time traffic sign detection and classification. IEEE Trans Intell Transp Syst 17(7):2022–2031
35. Zhong Z, Zheng L, Kang G, Li S, Yang Y (2017) Random erasing data augmentation. Comput Vis Pattern Recogn
36. Zhu Y, Zhang C, Zhou D, Wang X, Bai X, Liu W (2016) Traffic sign detection and recognition using fully convolutional network guided proposals. Neurocomputing 214:758–766
37. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. Comput Vis Pattern Recogn: 2110–2118

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Yiqiang Wu received bachelor degree in electronic information engineering, China university of petroleum, qingdao, in 2017. Currently he is studying in Hunan university for a master's degree. His research interests include Object detection, Object tracking, image segmentation and point cloud segmentation, etc.

Zhiyong Li (M¡¯14) received the M.Sc. degree in system engineering from the National University of Defense Technology, Changsha, China, in 1996, and the Ph.D. degree in control theory and control engineering from Hunan University, Changsha, in 2004. Since 2004, he has been with the College of Computer Science and Electronic Engineering, Hunan University, where he is currently a Full Professor. He has authored over 100 papers in international journals and conferences. His research interests include intelligent perception and autonomous moving body, machine learning and industrial big data, and intelligent optimization algorithms with applications. He is a member of the China Computer Federation and the Chinese Association for Artificial Intelligence.

Ying Chen was graduated from Hunan Normal University, China, in 2017. She is currently studying in Hunan University for a master’s degree. Her research interests include person re-identification, generative adversarial networks, and transfer learning.

Ke Nai received the M.Sc. degree in computer science and technology from the College of Information Science and Engineering, Center South University, China, in 2014. He is currently pursuing the Ph.D. degree with the College of Information Science and Engineering, Hunan University, China. His current research interests include visual tracking, face recognition, computer vision, pattern recognition, and machine learning.

Jin Yuan received the the Ph.D. degree in the School of Computing, National University of Singapore. Currently, he is an assistant professor in Hunan University of China. His scholarly research has primarily focused on multimedia retrieval. His current research interests include object detection, image and video caption and multimedia search etc.. He has authored more than ten journal and conference papers in these areas, including ACM MM, ICME, TMM, etc.