



Traffic sign detection based on improved faster R-CNN for autonomous driving

Xiaomei Li^{1,2} · Zhijiang Xie^{1,2} · Xiong Deng^{1,2} · Yanxue Wu^{1,2} · Yangjun Pi^{1,2} 

Accepted: 25 November 2021 / Published online: 6 January 2022

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

The timely and accurate identification of traffic signs plays a significant role in realizing the autonomous driving of vehicles. However, the size of traffic signs accounts for a low proportion of the input picture, which increases the difficulty of detection. This paper proposes an improved faster R-CNN traffic sign detection method. ResNet50-D feature extractor, attention-guided context feature pyramid network (ACFPN), and AutoAugment technology are designed for the faster R-CNN model. ResNet50-D is selected as the backbone network to obtain more characteristic information. ACFPN is performed to decrease the loss of contextual information. And data augmentation and transfer learning are adopted to make model training more convenient and time-saving. To prove the availability of the proposed method, we compare it with mainstream approaches (SSD, YOLOv3, RetinaNet, cascade R-CNN, FCOS, and CornerNet-Squeeze) and state-of-the-art methods. Experimental results on the CCTSDB dataset show that the improved faster R-CNN achieves the frames per second of 29.8 and the mean average precision of 99.5%, which is superior to the state-of-the-art methods and more suitable for traffic sign detection. Moreover, the proposed model is extended to the Tsinghua-Tencent 100 K (TT100K) dataset and also achieves a competitive detection result.

Keywords Faster R-CNN · ResNet50-D · ACFPN · AutoAugment · Autonomous driving · Traffic sign detection

✉ Yangjun Pi
cqpp@cqu.edu.cn

¹ State Key Laboratory of Mechanical Transmissions, Chongqing University, No. 174 Shazheng Street, Shapingba District, Chongqing City 400044, China

² College of Mechanical and Vehicle Engineering, Chongqing University, No. 174 Shazheng Street, Shapingba District, Chongqing City 400044, China

1 Introduction

In the past decade, the rise of deep learning has made human life more and more intelligent. One of the applications of deep learning is the detection and identification of traffic signs in autonomous driving environment perception. The level of environmental perception is a key reason to ensure that the vehicle can safely reach its destination. Traffic sign detection is crucial to the application of self-driving vehicles on the ground. Although many well-designed traffic signs are installed around the roads, thousands of people are killed or seriously injured in traffic accidents in the country every year. Therefore, traffic sign detection is a challenging practical problem.

The identification accuracy and speed of traffic signs are some of the important factors affecting the safety of autonomous driving, which has been researched by many scholars. Traditional traffic sign detection mainly extracts a large number of candidate areas through simple and effective features such as symbols, texts, colors, and shapes. Luo et al. [1] proposed an approach to recognize both symbol-based traffic signs and text-based traffic signs. Yang et al. [2] defined a detection module of traffic sign extraction and classification based on the color probability model and color features, which can obtain classification results from the convolutional neural network with high speed. Gudigar et al. [3] utilized multiple thresholding to extract region of interest (RoI) to detect traffic signs. Xu et al. [4] made use of the shape symmetry hypothesis algorithm to extract the RoI of the traffic sign. Among the approaches mentioned above, the color feature is one of the most conspicuous features of traffic signs and is used in many kinds of literature. However, these kinds of ways are weak in adaptability under complex environmental conditions.

In recent years, object detection algorithms on the basis of deep learning have been broadly utilized. Single shot detector (SSD) [5], faster region-based convolutional neural network (faster R-CNN) [6], you only look once (YOLO) series [7, 8], RetinaNet [9], cascade R-CNN [10], fully convolutional one-stage object detector (FCOS) [11], and CornerNet-Squeeze [12] are the current mainstream framework. For example, Yao et al. [13] focused on using YOLOv3 combined with DenseNet to detect foreign bodies, and He et al. [14] utilized popular object detection networks (SSD, faster R-CNN) together with different feature extractors (visual geometry group network (VGG), ResNet) to build foreign body detection models. Analogously, deep learning algorithms have been applied in industrial quality inspection [15], obstacle detection [16], remote sensing image detection [17], unmanned patrol inspection [18], and other aspects [19].

The research of deep learning is popular in the transportation field [20–26], especially in traffic sign detection. Traffic sign detection has been continuously improved in detection accuracy and speed and has made satisfactory progress. Lee et al. [27] utilized CNN to detect traffic signs by predicting the exact boundary and location of traffic signs simultaneously, which was more effective than other boundary estimation methods. Yang et al. [28] put an attention network (AN) in faster R-CNN to find the RoI and adopted fine RPN to generate the

terminal region proposal. Arcos-García et al. [29] analyzed the detection performance of several frameworks (faster R-CNN, R-FCN, SSD, and YOLOv2) combined with different backbone networks (ResNet50, Inception V2, Darknet19, and so on), discussed the characteristics of these models in the field of traffic sign detection through the method of transfer learning, and summarized the most appropriate model for different demands.

Song et al. [30] defined an efficient CNN with few parameters and fast network operation speed. Under the consistent model size, the detection accuracy was better than faster R-CNN. Li et al. [31] designed and implemented a traffic sign detector modeled on the framework of faster R-CNN and MobileNet. The performance of the detector was optimized by the location refinement approach according to color and shape information. A traffic sign detection method based on repeated attention was proposed by Tian et al. [32]. Their method effectively improved the use of context in images.

Zhang et al. [33] put forward two new lightweight networks, teacher network and student network, which obtained higher traffic sign detection accuracy while ensuring fewer trainable parameters. Wan et al. [34] presented a novel small traffic sign detection approach based on YOLOv3 architecture. Tabernik et al. [35] employed an improved mask R-CNN to realize the detection and recognition of traffic signs through end-to-end automatic learning. Gamez Serna et al. [36] designed a mask R-CNN method to test and refine the German Traffic Sign Detection Benchmark (GTSDB) and adopted the CNN architecture to achieve classification.

In 2021, scholars continued to study traffic sign detection to get more satisfactory results. The thin and deep convolutional neural network structure proposed by Haque et al. [37] detected traffic signs rapidly and accurately. In [38], a cascade cascade network structure with class hierarchy was proposed for traffic sign detection. Their method has good performance and can meet the real-time requirements of autonomous driving. Tang et al. [39] presented a neck network IFA-FPN to solve the problem of difficult detection caused by the imbalance of the size and category distribution of traffic signs. Experimental results performed on three mainstream datasets showed the superiority of the algorithm. Shen et al. [40] proposed a new group multi-scale attention pyramid network to detect small traffic signs, which achieved high accuracy. Liu et al. [41] presented a new network TSingNet, which used AbFPN to learn scale-aware features and ARFF blocks to derive more context information. Many experimental results showed that TSingNet was superior to the state-of-the-art approaches.

The traffic sign detection based on deep learning has good performance, but there are still some problems that need further improvement. Traffic signs are small and multi-scale, which increases the difficulty of feature extraction. During the training process, the network undergoes multiple convolutions and sums. After the pooling operation, many feature map details will be lost, and the pixel-level deviation of the feature map will often cause errors in detection, greatly reducing the accuracy of network detection, so it is more difficult to identify small-size traffic signs. In the real driving environment, traffic signs are very easy to be confused with billboards or other similar signs. For convolutional neural networks, illumination changes, partial occlusion, complex background, etc., also make traffic sign detection more

challenging. Therefore, in terms of detection accuracy and speed, the general object detection approaches cannot better meet the requirements.

Aiming at the above traffic sign detection problems, we choose the faster R-CNN model, because it is more sensitive to small target traffic signs and is an end-to-end detection network. In addition, it shares parameters in the training process, which reduces the training time and improves model quality. Faster R-CNN is used as the baseline plan and improved on this basis to achieve better model performance.

The main contributions of this study are as follows:

1. An improved faster R-CNN model is proposed, which can effectively improve the accuracy of traffic sign detection. Experiments on the CCTSDB dataset show that the proposed method is competitive for traffic sign detection, and the mAP is up to 99.5%.
2. ResNet50-D with fewer parameters is adopted as the backbone network to realize feature extraction. Compared with ResNet50, the accuracy is improved by 0.5%.
3. ACFPN is introduced into the traffic sign detection, which focuses on context information and only slightly increases the model complexity and calculation, but the detection accuracy is further improved by 0.7%.
4. The data augmentation methods such as random flip, resize, and AutoAugment are adopted to enhance the generalization ability of the model. Transfer learning can significantly reduce training time and speed up network convergence.
5. Our method can be extended to detect other traffic sign datasets. The adaptability experiment on the TT100K dataset further proves its usability.

The rest of the paper is arranged as follows: Sect. 2 presents the proposed method. In Sect. 3, the experimental process is described, containing the training ways, data processing methods, transfer learning approach, and so on. Section 4 gives the experimental results and the discussion. Conclusions and future work are presented in Sect. 5.

2 Proposed method

2.1 Improved faster R-CNN

In this study, the traffic sign detection system uses the original faster R-CNN detector, which is a general method developed for the detection and recognition of general objects. To adapt it to the specific field of traffic sign detection, some improvements are proposed. Firstly, the traffic sign images are processed by using the data augmentation approaches, then the characteristics are obtained by inputting ResNet50-D, and the multi-level characteristics are output. Secondly, the multi-level features are input into the ACFPN for feature fusion to generate a multi-scale characteristic pyramid with rich context information. Thirdly, the feature pyramid is input to the RPN to produce region proposals. Fourthly, the feature pyramid and region proposals are transported to the RoI align layer to produce proposal feature maps. Finally,

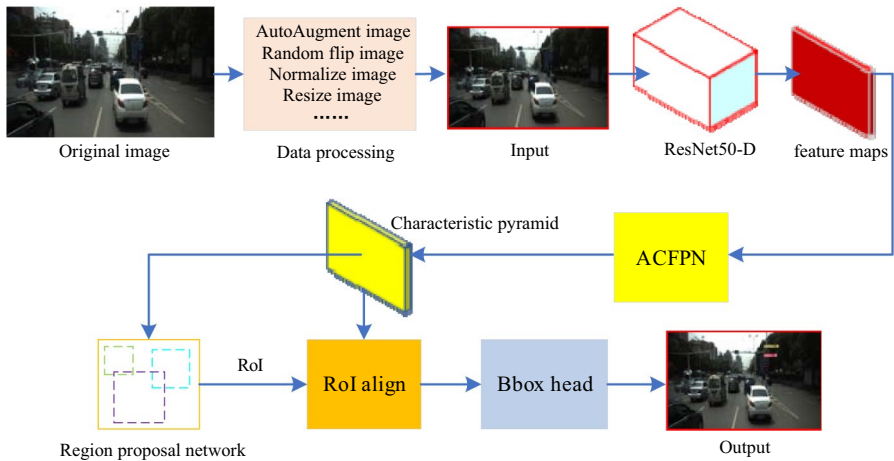


Fig. 1 The overall structure

the proposal feature maps are sent to the Bbox head to achieve object classification and obtain the precise position of the object bounding box. The overall structure is shown in Fig. 1.

Compared with faster R-CNN, the proposed method has the following advantages: (1) A series of data augmentation methods, such as AutoAugment, random flip, and resize, are adopted to enhance the generalization capacity and detection performance; (2) ResNet50-D is conducted as the backbone network for characteristic extraction, which improves the capability of characteristic extraction; (3) the introduction of ACFPN can reduce the loss of context information, so ACFPN is used to perform feature fusion on the multi-level feature output of ResNet50-D, and a feature map with rich semantic and location information is obtained.

2.2 Faster R-CNN

Region-based networks mostly contain R-CNN, fast R-CNN, and faster R-CNN. Both R-CNN and fast R-CNN adopt selective search (SS) to extract 2000 candidate area frames, which spend too much time and are difficult to meet the real-time requirements. Therefore, based on R-CNN and fast R-CNN, the typical two-stage model faster R-CNN came into being. Its core idea is to use RPN (region proposal network) to replace SS to extract candidate regions.

Here, we briefly describe faster R-CNN and see [6] for a more explicit illustration. The faster R-CNN uses anchor to replace SS to extract candidate areas. The anchor containing the target object is screened and employed in RoI pooling to extract the feature map of fixed size. Then, the candidate regions are classified using proposal feature maps and the target object position is predicted using bounding box regression. The faster R-CNN architecture is shown in Fig. 2.

In faster R-CNN, RPN is a fully connected network for the generation of candidate regions.

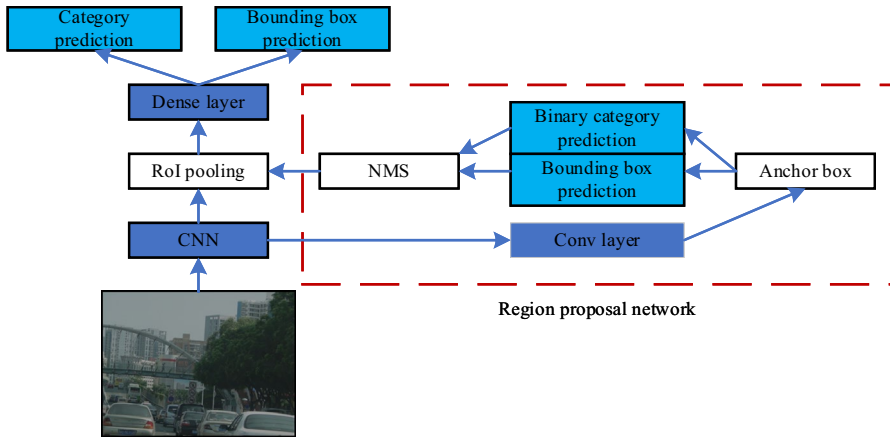


Fig. 2 Architecture of faster R-CNN

RPN loss is defined as follows:

$$L_{rpn} = L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (1)$$

where the classification loss function is

$$L_{cls}(p_i, p_i^*) = -\log [p_i p_i^* + (1 - p_i^*)(1 - p_i)] \quad (2)$$

The regression loss function is

$$L_{reg}(t_i, t_i^*) = R(t_i - t_i^*) \quad (3)$$

R is the robustness loss function:

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2, & \text{if } |x| < 1 \\ |x| - 0.5, & \text{otherwise} \end{cases} \quad (4)$$

$$t_i = \{t_x, t_y, t_w, t_h\} \quad (5)$$

where i is the exponent of the anchor in a minimum sample; P_i is the chance of the anchor to predict the object; P_i^* represents classified supervision information, and its value is 0 (anchor is a negative sample) or 1 (anchor is a positive sample); λ is a constant for balancing the classification loss and regression loss; t_i expresses the four parameterized coordinates of the predicted candidate box; t_i^* represents the coordinate vector of the corresponding candidate box when the anchor generation box is positive; N_{cls} and N_{reg} represent the classification loss and regression loss after normalization treatment, respectively.

2.3 ResNet50-D

In general, the more the network layers are, the richer features are extracted and the more semantic information is obtained. But simply increasing the depth of the network will lead to gradient explosion. ResNet network using residual connection can be a good way to raise neural network depth when the gradient disappears. ResNet according to the number of different network layers mainly has ResNet18, ResNet34, ResNet50, ResNet101, ResNet152, and so on. ResNet50 is the most commonly used. However, in the process of ResNet50 downsampling, the convolution of 1×1 and the stride of 2 will cause that three-quarters of the input feature map will not be utilized. Therefore, the backbone network selected for our method is ResNet50-D. As shown in Fig. 3, ResNet50-D moves the downsampling of path A to the following 3×3 convolution for execution, so all information is used. In the downsampling of path B, 2×2 average pooling layer with a stride of 2 is added before the convolution, so the loss of a large number of feature information is avoided. Compared with ResNet50, ResNet50-D uses an average pooling method to reduce the dimension during the skip connection. The advantages of average pooling are that it can retain image background information well, extract features better, reduce the number of parameters, and save hardware resources.

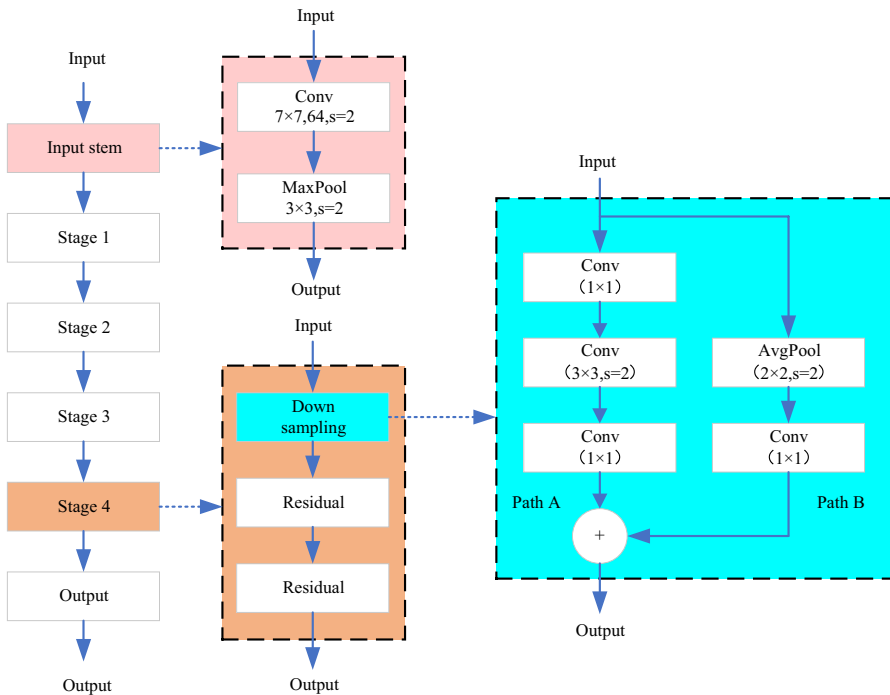


Fig. 3 The architecture of ResNet50-D

2.4 RoI align

The core of RoI pooling is that candidate boxes share features of the feature map and keep the output size consistent. Nevertheless, when RoI pooling is approximated by two rounds, it causes a mismatch between the detection information and the extracted features. To enhance the identification correctness of traffic signs, RoI pooling is converted into RoI align in faster R-CNN.

RoI align can eliminate errors generated in RoI pooling. Firstly, RoI align takes N points uniformly in the region and finds the four points closest to each point in the feature map. Then, the export value of the points is obtained by bilinear interpolation, as shown in Fig. 4. Finally, the output value of the region is obtained by averaging the N points. Consequently, the whole characteristic aggregation procedure is turned into a consecutive manipulation to acquire better performance and retain the features of the original region as much as possible.

The results from the RPN are carried to the RoI align layer along with the feature pyramid, and the final result is obtained after the two fully connected layers. In practical terms, the chance of the traffic sign in the bounding box can be obtained by the classification layer, and the position of the traffic sign can be obtained through the regression layer.

2.5 Attention-guided context feature pyramid network (ACFPN)

To take advantage of the intrinsic multi-scale characteristic representation of deep convolutional networks, feature pyramid network (FPN) is proposed. Unfortunately, there are two problems in FPN: (1) insufficient communication between multi-scale receptive fields; (2) the contradiction between the resolution of the feature map and the receptive field for high-resolution input pictures. To address the defects mentioned above, a new ACFPN is proposed. The structure is shown in Fig. 5.

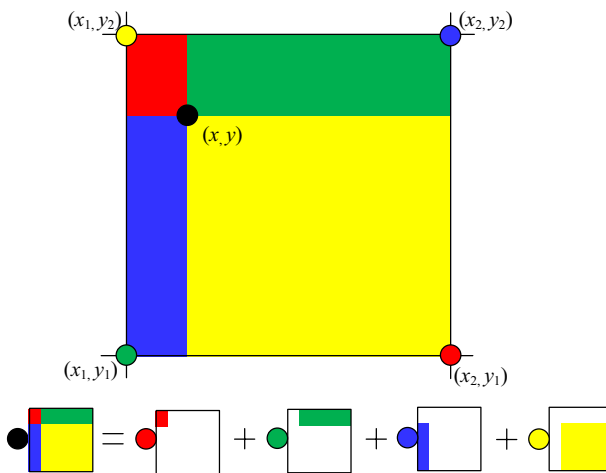


Fig. 4 Bilinear interpolation diagram

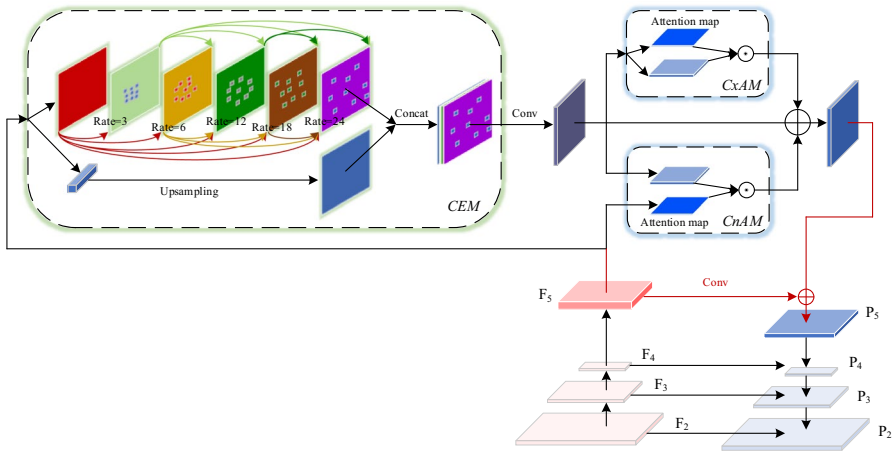


Fig. 5 The structure of ACFPN network

Compared with the original FPN, ACFPN uses dilated convolution on F_5 and sub-sample P_5 and maintains the same step size as FPN through max pooling. ACFPN [42] is composed of two modules, namely context extraction module (CEM) and attention-guided module (AM), which solve the contradiction between feature map resolution and receptive field on high-resolution input. CEM searches extensive contextual information from multiple receiving domains. The AM is composed of two sub-modules: contextual attention module (CxAM) and content attention module (CnAM), which are dedicated to seizing and distinguishing semantics and accurate positioning.

3 Experiment

3.1 Training

In this study, the benchmark datasets are the CCTSDB traffic sign dataset released by Changsha University of Science and Technology [43] and the TT100K dataset [44]. The CCTSDB dataset contains three categories, namely warning, prohibitory, and mandatory. There are 11,062 pictures for training and 2767 pictures for testing, both of which contain all the categories of traffic signs without intersections. The TT100K dataset contains 232 categories, 6105 pictures for training and 3071 pictures for testing.

All methods are trained and tested using the PaddlePaddle framework, the operating environment is Python, and the cloud hardware is configured with a 4-core CPU, 32 GB memory, and a NVIDIA Tesla V100, 16G video memory.

Stochastic gradient descent (SGD) is employed to train the whole network, and the initial learning rate is 0.01. When iterating 80,000 and 100,000, the learning rate is divided by 10, respectively. The weight decay is set to 0.0001. The momentum

is set to 0.9. The Batch_size is set to 2. The training network is initialized by using the pre-trained model of ImageNet classification of object detection, and the initial parameters and weight of the traffic sign detection training model are obtained. The iterative training is 120,000 times. During the training, the model is evaluated every 10,000 times, and the evaluation results are saved until the end of the training.

3.2 Data processing

The CCTSDB dataset has different sizes, such as 1280*720, 1024*768, and 1000*350. The image resolution of the TT100K dataset is 2048*2048. The target size of the faster R-CNN network is modified to 800. All models adopt data augmentation methods such as random flip, normalization, and resize, mainly to increase the number of pictures in the training set and diversify them to enhance the generalization ability of the model. Moreover, the model trained by data augmentation has better detection performance.

In addition to the data augmentation mentioned above, the proposed method also uses AutoAugment technology. Here, we briefly describe AutoAugment strategies and refer to [45] for a more detailed description. The idea is to establish a search space for data augmentation strategies. A strategy in the search space includes many sub-strategies. Each sub-strategy consists of two image-processing functions and function parameters (probability and amplitude). The sub-strategy will be randomly assigned to each picture in each mini-batch. The search algorithm is adopted to search for the best strategy to ensure higher accuracy.

3.3 Increase the number of anchor frames

To adapt to traffic sign detection and detect small targets more successfully, besides the default anchor size, 32*32 and 64*64 parameters are also added, so that each anchor point corresponds to 15 candidate windows, namely anchor_sizes {32*32, 64*64, 128*128, 256*256, 512*512} and aspect_ratios {1:2, 1:1, 2:1}.

3.4 Transfer learning

At present, object detection algorithms based on big datasets and deep CNN models show prominent capability. However, in practical application, when the amount of training data is small, the model has fallen into overfitting. Therefore, transfer learning can solve this problem well. In this study, the ImageNet classification pre-training model is employed to initialize the detection model. We use our dataset in transfer learning and convert it into scripts for COCO format. In transfer learning, the variables of the pre-training model are different from those of the actual model, so there is a mismatch problem. Therefore, it is necessary to selectively load the pre-training model and directly load the pre-training weight, so that the parameters with different shapes of corresponding parameters in the model and the pre-training model will be automatically ignored. Using transfer learning can significantly reduce training time.

3.5 Performance evaluation index

To compare the performance of the method presented in this study with other methods, the following evaluation indexes are utilized.

(1) mAP

AP indicates the performance of each class in the detection model, and mAP is the average of AP values under all categories. mAP with intersection over union (IoU)=0.5 and area=all is used as the evaluation index in the validation set, and COCO metrics are adopted. The mAP calculates the mean value of AP of each category when IoU_threshold=0.5. The mAP represents the detection performance of the detection model.

(2) FPS

FPS expresses the number of frames to fill the image per second. By listing the FPS of different methods, the real-time capability of different methods is revealed.

(3) Precision, Recall_rate, Missing_rate and F1-measure

Precision represents the proportion of accurate prediction samples in all prediction samples. Recall_rate shows the probability that the predicted accurate sample accounts for all positive samples. Missing_rate is the opposite of Recall_rate. F1-measure is the index for comprehensive evaluation of precision and Recall_rate. True positive (TP), true negative (TN), false positive (FP), and false negative (FN) denote four cases of detection results, respectively. These indicators are defined as follows:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

$$Recall_rate = \frac{TP}{TP + FN} \quad (7)$$

$$Missing_rate = 1 - \frac{TP}{TP + FN} \quad (8)$$

$$F1 - measure = 2 \times \frac{Precision \times Recall_rate}{Precision + Recall_rate} \quad (9)$$

4 Results and discussion

4.1 Experimental results and discussion

Figure 6 shows some detection results on the CCTSDB dataset. These traffic signs are photographed on different roads. The shape, color, and size of traffic signs are also different.

Although there are many proposed methods in the field of traffic sign detection, it is difficult to make a reliable comparison with these methods because of different datasets and inconsistent hyperparameters. Therefore, to testify the capability of the proposed method in traffic sign detection, we compare it with some object detection approaches, namely SSD [5], YOLOv3 [8], RetinaNet [9], cascade R-CNN [10],



Fig. 6 Part of the detection results on the CCTSDB dataset

FCOS [11], and CornerNet-Squeeze [12]. These models use the same CCTSDB dataset, set some of the same hyperparameters (momentum, number of training iterations, and so on), measure the performance on the same hardware with a NVIDIA Tesla V100X GPU and a 4-core CPU, 32 GB memory, and compare them on both the mAP and FPS metrics. The experimental results are shown in Table 1. In addition, Fig. 7 shows the more intuitive results.

Obviously, the detection accuracy and speed of the original faster R-CNN model are not good. When ResNet50-D, a better backbone network, is applied to faster R-CNN and ACFPN module and AutoAugment technology, the best detection effect is achieved. In this study, the mAP of the improved faster R-CNN is 99.5%, which is the highest among the eight detection models. FPS is 29.8, although lower than the one-stage detection models SSD and YOLOv3, but close to 30, which can achieve real-time detection of traffic signs. In contrast, the mAP is significantly higher than them. In the case of a significant increase in mAP, the sacrifice of time is worth it, because more attention should be paid to accuracy while meeting the real-time requirements. Compared with the original faster R-CNN model, the mAP and FPS of the improved faster R-CNN model are increased by 1.8% and 24.5%, respectively. And the FPS is significantly improved by at least five times. The improvement in the method in this study is better reflected in the FPS indicator. This indicates that the improved faster R-CNN model has better comprehensive performance than other models.

In the one-stage model, compared with SSD, YOLOv3, and RetinaNet, the improved faster R-CNN enhances 13.7%, 4.9%, and 1.1% in the mAP index, respectively, which is significantly better than the SSD and YOLOv3 models, because the faster R-CNN is a two-stage model consisting of regional recommendation and goal classification. First, RoI align is performed for each proposal extracted. In the second stage, there are two full connection layers to classify the candidate regions and predict the position of the target object, so higher detection accuracy is obtained. However, its inference speed is significantly lower than that of SSD and YOLOv3 models, mainly because faster R-CNN uses RPN to acquire a group of candidate regions and then predicts it. Consequently, the detection speed of faster R-CNN is

Table 1 Experimental results of different object detection methods on the CCTSDB dataset

Method	Backbone	mAP (%)	FPS
SSD [5]	VGG16	85.8	57.7
YOLOv3 [8]	DarkNet53	94.6	47.4
RetinaNet [9]+FPN	ResNet50	98.4	27.3
Cascade R-CNN [10]	ResNet50-D	97.9	23.0
FCOS [11]+FPN	ResNet50-DCN	98.9	17.4
CornerNet-Squeeze [12]+FPN	ResNet50-D-DCN	90.9	28.5
Faster R-CNN [6]	ResNet50	97.7	5.3
Faster R-CNN + ACFPN + AutoAugment (Ours)	ResNet50-D	99.5	29.8

All results are obtained on the same hardware

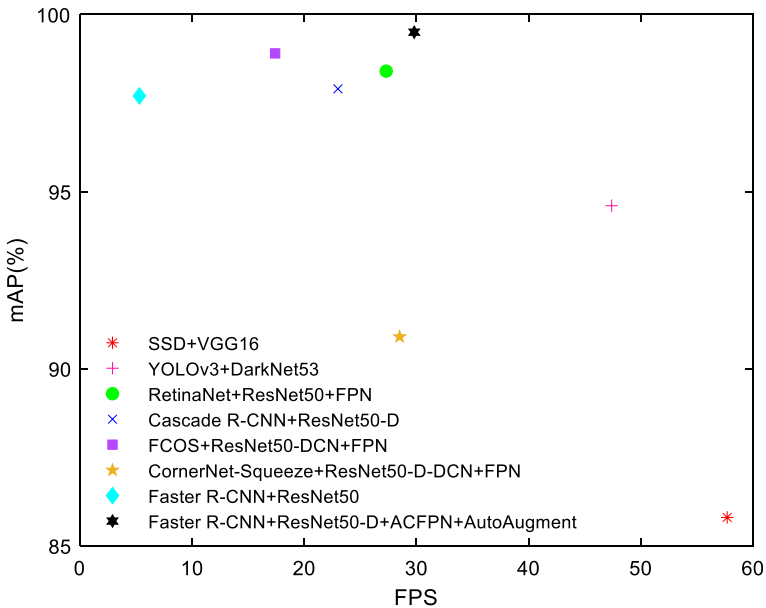


Fig. 7 Performance comparison of each model on the CCTSDB dataset

impeded, while SSD and YOLOv3 are fast, because they are one-stage detection models with fewer layers than other models, whereas this does not impact the overall detection behavior of the proposed method.

In the two-stage model, compared with faster R-CNN and cascade R-CNN, the improved faster R-CNN not only increases 1.6% and 1.8% in the mAP index but also improves 6.8 and 24.5 in the FPS index, respectively. The improvement is noticeable.

In the anchor-free model, compared with FCOS and CornerNet-Squeeze, the improved faster R-CNN not only raises 0.6% and 8.6% in the mAP index but also increases 12.4 and 1.3 in the FPS index, respectively.

An illustration of precision–recall curves for these methods is provided in Fig. 8, which further demonstrates the validity of the proposed model. From Fig. 8, among three types of traffic signs including warning, prohibitory, and mandatory, it can be observed that the best way is faster R-CNN + ResNet50-D + ACFPN + AutoAugment which achieved the highest average precision.

Additionally, the proposed model is compared with the state-of-the-art approaches on the CCTSDB dataset. It can be easily seen from Table 2 that our method is superior to Shen et al. [40] on the Recall_rate, Missing_rate, and F1-measure indicators. Compared with Liu et al. [38], the proposed model acquires over 9.0% improvement on mAP. Therefore, these results further confirm the availability of the proposed method.

To further prove the effectiveness of our method, the faster R-CNN + ResNet50 model is used as the baseline model for a comparative experiment. Table 3 displays the average recall (AR) rate on the CCTSDB dataset. Compared with the benchmark

Fig. 8 Precision–recall curves of traffic sign detection performance on the CCTSDB dataset for warning, prohibitory, and mandatory

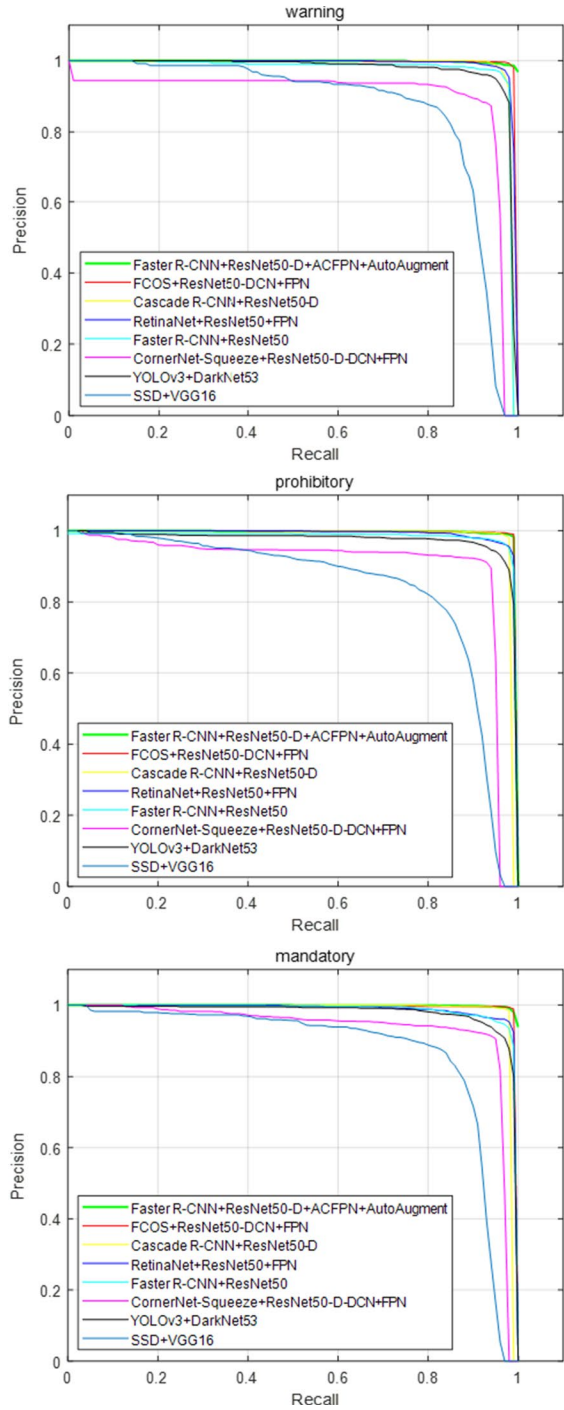


Table 2 Comparison of our method with the state-of-the-art methods on the CCTSDB dataset

Method	Year	mAP	Precision	Recall_rate	Missing_rate	F1-measure
Li et al. [46]	2018	–	86.7%	75.6%	24.4%	80.8%
Zhang et al. [47]	2020	–	99.7%	83.62%	16.38%	90.82%
Shen et al. [40]	2021	–	99.4%	91.8%	8.2%	95.4%
Liu et al. [38]	2021	89.7%	–	–	–	–
Ours	2021	99.5%	98.8%	98.3%	1.7%	98.5%

The results of all methods are derived from the original paper

model, the proposed method increases 6.8% in AR^S , and the effect is significantly improved, which verifies that the proposed method significantly enhances the test capacity of little targets. The performance of the proposed method on AR^L is poor, because ACFPN will pay more attention to small targets after fusing more information, which improves the detection performance of small targets and reduces the attention to large targets.

In summary, the improved faster R-CNN model has higher accuracy compared with the other models and achieves a satisfactory balance between detection accuracy and speed. It shows that the improved faster R-CNN model has superior performance and is effective in the field of traffic sign detection.

4.2 Ablation study

To discuss the influence of ResNet50-D, ACFPN, and AutoAugment technology on the detection capacity of faster R-CNN, we conduct an ablation study on the CCTSDB dataset. First, the backbone network is changed to ResNet50-D to study its impact on the faster R-CNN. Next, we demonstrate the role of the ACFPN in the faster R-CNN detection model. Finally, the AutoAugment technique is added to further evaluate the performance. Table 4 reveals the experimental results of the ablation study.

It can be seen from Table 4 that, by changing the backbone network ResNet50-D, the mAP of faster R-CNN is increased by 0.5%. Faster R-CNN with FPN is superior to the faster R-CNN without FPN, with a difference of 0.6%. The effectiveness of ACFPN is also proved in our proposed model. After replacing FPN with ACFPN, the detection capability is enhanced, and the network effect is further improved. The mAP is improved by 0.1%. Finally, after adding AutoAugment technology, the mAP value increased by 0.6%. These improvements are

Table 3 Average recall rate performance comparison on the CCTSDB dataset

Model	AR^{10} (%)	AR^{100} (%)	AR^S (%)	AR^M (%)	AR^L (%)
Faster R-CNN + ResNet50	77.1	77.1	73.6	81.3	91.7
Faster R-CNN + ResNet50-D + ACFPN + AutoAugment	82.6	82.6	80.4	85.2	91.7

All results are obtained on the same hardware

Table 4 Experimental results of ablation study on the CCTSDB dataset

Model	mAP (%)	FPS
Faster R-CNN + ResNet50	97.7	5.3
Faster R-CNN + ResNet50-D	98.2	5.7
Faster R-CNN + ResNet50-D + FPN	98.8	31.6
Faster R-CNN + ResNet50-D + ACFPN	98.9	29.4
Faster R-CNN + ResNet50-D + ACFPN + AutoAugment	99.5	29.8

All results are obtained on the same hardware

mainly because of the capability of the ResNet50-D network to retain image background information and better extract features. By fusing the features of different receptive fields, ACFPN not only increases the receptive fields of objects, but also gives better classification results by using the context information of objects. AutoAugment technology improves accuracy by automatically searching data augmentation strategies suitable for specific datasets.

4.3 Adaptability experiment

To prove the adaptability of the proposed method, it is applied to the TT100K dataset. The experimental results are shown in Table 5. The improved faster R-CNN acquires a competitive result with 95.8% in mAP, which surpasses some mainstream approaches, such as SSD, YOLOv3, and faster R-CNN. Simultaneously, the improved faster R-CNN is compared with the state-of-the-art approaches, and the results are shown in Table 6. The improved faster R-CNN obtains 2.2%, 0.38%, and 1.08% improvement on mAP, precision, and Recall rate, respectively. Table 6 further illustrates the superiority of the proposed method. This shows that our method has strong adaptability. This means that when the proposed method is applied to a new dataset, it can work well only by fine-tuning.

To sum up, ResNet50-D, ACFPN, and AutoAugment technology can availablely ameliorate the traffic sign detection performance of the faster R-CNN model.

Table 5 Experimental results of different object detection methods on the TT100K dataset

Method	Backbone	mAP (%)	FPS
SSD [5]	VGG16	87.0	7.4
YOLOv3 [8]	DarkNet53	89.2	11.1
Faster R-CNN [6]	ResNet50	80.8	3.5
Faster R-CNN + ACFPN + Auto-Augment (Ours)	ResNet50-D	95.8	3.4

All results are obtained on the same hardware

Table 6 Comparison of our method with the state-of-the-art methods on the TT100K dataset

Method	Year	mAP	Precision	Recall_rate
Yang et al. [28]	2018	80.31%	–	–
Wang et al. [34]	2020	–	93.80%	92.25%
Qing et al. [39]	2021	93.6%	–	–
Liu et al. [41]	2021	93.3%	–	–
Ours	2021	95.8%	94.18%	93.33%

The results of all methods are derived from the original paper

5 Conclusions

Traffic sign detection is one of the important premises for the realization of autonomous driving environment perception. In the real driving road scene, the target size of traffic signs is small, and the environment changes in real time, which increases the difficulty of traffic sign detection. This paper proposes an improved faster R-CNN traffic sign detection model, which is more suitable for traffic sign detection by selecting a better backbone network (ResNet50-D), RPN optimization (anchor size), hyperparameter, ACFPN, and AutoAugment technology. Compared with the state-of-the-art approaches on the CCTSDB dataset, our method obtains the mAP of 99.5% and the FPS of 29.8. In addition, the adaptability experiment on the TT100K dataset further proves its superiority. Therefore, the proposed method has more advantages and stronger generalization ability, which can provide the idea for the environment perception of autonomous driving and can be extended to the area of small object detection like traffic signs.

The model results show that the ideal performance is still not achieved. This is mainly because there is an apparent disparity between the proposed approach and the one-stage object detection method in real-time detection speed. Future work will focus on this aspect and optimize the detection model to further improve the detection speed. In addition, we also want to test the performance of the proposed traffic sign model in more traffic scenarios.

Acknowledgements This work was supported by the National Key Research and Development Program of China [Grant No. 2018YFC0808004], Fundamental Research Funds for the Central Universities [Project No. 2018CDXYJX0019], and project supported by Graduate Scientific Research and Innovation Foundation of Chongqing, China [Grant No. CYB19009].

Funding Funding was provided by National Key Research and Development Program of China (2018YFC0808004), Fundamental Research Funds for the Central Universities (2018CDXYJX0019), and Graduate Scientific Research and Innovation Foundation of Chongqing, China (CYB19009).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Luo H, Yang Y, Tong B, Wu F, Fan B (2018) Traffic sign recognition using a multi-task convolutional neural network. *IEEE Trans Intell Transp Syst* 19(4):1100–1111. <https://doi.org/10.1109/TITS.2017.2714691>
2. 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. <https://doi.org/10.1109/TITS.2015.2482461>
3. Gudigar A, Chokkadi S, Raghavendra U, Acharya UR (2016) Multiple thresholding and subspace based approach for detection and recognition of traffic sign. *Multimed Tools Appl* 76(5):6973–6991. <https://doi.org/10.1007/s11042-016-3321-6>
4. Xu X, Jin J, Zhang S, Zhang L, Pu S, Chen Z (2019) Smart data driven traffic sign detection method based on adaptive color threshold and shape symmetry. *Futur Gener Comput Syst* 94:381–391. <https://doi.org/10.1016/j.future.2018.11.027>
5. Liu W et al (2016) SSD: single shot multibox detector. In: 14th European Conference on Computer Vision (ECCV), vol 9905, pp 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
6. Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39(6):1137–1149. <https://doi.org/10.1109/tpami.2016.2577031>
7. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 779–788. <https://doi.org/10.1109/CVPR.2016.91>
8. Redmon J, Farhadi A (2018) YOLOv3: an incremental improvement. In: 2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [arxiv:1804.02767](https://arxiv.org/abs/1804.02767)
9. Lin T, Goyal P, Girshick R, He K, Dollár P (2020) Focal loss for dense object detection. *IEEE Trans Pattern Anal Mach Intell* 42(2):318–327. <https://doi.org/10.1109/tpami.2018.2858826>
10. Cai Z, Vasconcelos N (2021) Cascade R-CNN: high quality object detection and instance segmentation. *IEEE Trans Pattern Anal Mach Intell* 43(5):1483–1498. <https://doi.org/10.1109/tpami.2019.2956516>
11. Tian Z, Shen C, Chen H, He T (2020) FCOS: a simple and strong anchor-free object detector. *IEEE Trans Pattern Anal Mach Intell*. <https://doi.org/10.1109/tpami.2020.3032166>
12. Law H, Teng Y, Russakovsky O, Deng J (2019) CornerNet-Lite: efficient keypoint based object detection. In: 2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [arxiv:1904.08900](https://arxiv.org/abs/1904.08900)
13. Yao Z, He D, Chen Y, Liu B, Miao J et al (2020) Inspection of exterior substance on high-speed train bottom based on improved deep learning method. *Measurement*. <https://doi.org/10.1016/j.measurement.2020.108013>
14. He D, Yao Z, Jiang Z, Chen Y, Deng J, Xiang W (2019) Detection of foreign matter on high-speed train underbody based on deep learning. *IEEE Access* 7:183838–183846. <https://doi.org/10.1109/ACCESS.2019.2960439>
15. Zheng X, Zheng S, Kong Y, Chen J (2021) Recent advances in surface defect inspection of industrial products using deep learning techniques. *Int J Adv Manuf Technol* 113(1):35–58. <https://doi.org/10.1007/s00170-021-06592-8>
16. He D, Zou Z, Chen Y, Liu B, Yao X, Shan S (2021) Obstacle detection of rail transit based on deep learning. *Measurement*. <https://doi.org/10.1016/j.measurement.2021.109241>
17. Zhong Y, Zheng Z, Ma A, Lu X, Zhang L (2020) Color: cycling, offline learning, and online representation framework for airport and airplane detection using GF-2 satellite images. *IEEE Trans Geosci Remote Sens* 58(12):8438–8449. <https://doi.org/10.1109/TGRS.2020.2987907>
18. Chen B, Miao X (2020) Distribution line pole detection and counting based on YOLO using UAV inspection line video. *J Electr Eng Technol* 15(1):441–448. <https://doi.org/10.1007/s42835-019-00268-w>
19. Polat Ö, Güngen C (2021) Classification of brain tumors from MR images using deep transfer learning. *J Supercomput* 77(7):7236–7252. <https://doi.org/10.1007/s11227-020-03572-9>
20. Yoneda K, Kuramoto A, Suganuma N, Asaka T, Aldibaja M, Yanase R (2020) Robust traffic light and arrow detection using digital map with spatial prior information for automated driving. *Sensors*. <https://doi.org/10.3390/s20041181>

21. Pon A, Adrienko O, Harakeh A, Waslander SL (2018) A hierarchical deep architecture and mini-batch selection method for joint traffic sign and light detection. In: 2018 15th Conference on Computer and Robot Vision (CRV). <https://doi.org/10.1109/CRV.2018.00024>
22. Wang H, Li Y, Wang S (2020) Fast pedestrian detection with attention-enhanced multi-scale rpn and soft-cascaded decision trees. *IEEE Trans Intell Transp Syst* 21(12):5086–5093. <https://doi.org/10.1109/TITS.2019.2948398>
23. Cai Y, Liu Z, Sun X, Chen L, Wang H, Zhang Y (2017) Vehicle detection based on deep dual-vehicle deformable part models. *J Sens* 2017:1–10. <https://doi.org/10.1155/2017/5627281>
24. Hu X et al (2019) SINet: a scale-insensitive convolutional neural network for fast vehicle detection. *IEEE Trans Intell Transp Syst* 20(3):1010–1019. <https://doi.org/10.1109/TITS.2018.2838132>
25. Tang J, Li S, Liu P (2021) A review of lane detection methods based on deep learning. *Pattern Recogn*. <https://doi.org/10.1016/j.patcog.2020.107623>
26. Kausar A, Jamil A, Nida N, Yousaf MH (2020) Two-wheeled vehicle detection using two-step and single-step deep learning models. *Arab J Sci Eng* 45(12):10755–10773. <https://doi.org/10.1007/s13369-020-04837-4>
27. Lee HS, Kim K (2018) Simultaneous traffic sign detection and boundary estimation using convolutional neural network. *IEEE Trans Intell Transp Syst* 19(5):1652–1663. <https://doi.org/10.1109/tits.2018.2801560>
28. Yang T, Long X, Sangaiah AK, Zheng Z, Tong C (2018) Deep detection network for real-life traffic sign in vehicular networks. *Comput Netw* 136:95–104. <https://doi.org/10.1016/j.comnet.2018.02.026>
29. Arcos-García Á, Álvarez-García JA, Soria-Morillo LM (2018) Evaluation of deep neural networks for traffic sign detection systems. *Neurocomputing* 316:332–344. <https://doi.org/10.1016/j.neucom.2018.08.009>
30. Song S, Que Z, Hou J, Du S, Song Y (2019) An efficient convolutional neural network for small traffic sign detection. *J Syst Archit* 97:269–277. <https://doi.org/10.1016/j.sysarc.2019.01.012>
31. Li J, Wang Z (2019) Real-time traffic sign recognition based on efficient CNNs in the Wild. *IEEE Trans Intell Transp Syst* 20(3):975–984. <https://doi.org/10.1109/TITS.2018.2843815>
32. Tian Y, Gelernter J, Wang X, Li J, Yu Y (2019) Traffic sign detection using a multi-scale recurrent attention network. *IEEE Trans Intell Transp Syst* 20(12):4466–4475. <https://doi.org/10.1109/tits.2018.2886283>
33. Zhang J, Wang W, Lu C, Wang J, Sangaiah AK (2020) Lightweight deep network for traffic sign classification. *Ann Telecommun* 75(7–8):369–379. <https://doi.org/10.1007/s12243-019-00731-9>
34. Wan J et al (2020) An efficient small traffic sign detection method based on YOLOv3. *J Signal Process Syst Signal Image Video Technol*. <https://doi.org/10.1007/s11265-020-01614-2>
35. Tabernik D, Skocaj D (2020) Deep learning for large-scale traffic-sign detection and recognition. *IEEE Trans Intell Transp Syst* 21(4):1427–1440. <https://doi.org/10.1109/TITS.2019.2913588>
36. Gamez Serna C, Ruichek Y (2020) Traffic signs detection and classification for European urban environments. *IEEE Trans Intell Transp Syst* 21(10):4388–4399. <https://doi.org/10.1109/TITS.2019.2941081>
37. Haque WA, Arefin S, Shihavuddin ASM, Hasan MA (2021) DeepThin: a novel lightweight CNN architecture for traffic sign recognition without GPU requirements. *Expert Syst Appl*. <https://doi.org/10.1016/j.eswa.2020.114481>
38. Liu Z, Qi M, Shen C, Fang Y, Zhao X (2021) Cascade saccade machine learning network with hierarchical classes for traffic sign detection. *Sustain Cities Soc*. <https://doi.org/10.1016/j.scs.2020.102700>
39. Tang Q, Cao G, Jo KH (2021) Integrated feature pyramid network with feature aggregation for traffic sign detection. *IEEE Access* 9:117784–117794. <https://doi.org/10.1109/access.2021.3106350>
40. Shen L, You L, Peng B, Zhang C (2021) Group multi-scale attention pyramid network for traffic sign detection. *Neurocomputing* 452:1–14. <https://doi.org/10.1016/j.neucom.2021.04.083>
41. Liu Y, Peng J, Xue J-H, Chen Y, Fu Z-H (2021) TSingNet: Scale-aware and context-rich feature learning for traffic sign detection and recognition in the wild. *Neurocomputing* 447:10–22. <https://doi.org/10.1016/j.neucom.2021.03.049>
42. Cao J, Chen Q, Guo J, Shi R (2020) Attention-guided context feature pyramid network for object detection. [arXiv:2005.11475](https://arxiv.org/abs/2005.11475)

43. Zhang J, Huang M, Jin X, Li X (2017) A real-time Chinese traffic sign detection algorithm based on modified YOLOv2. *Algorithms*. <https://doi.org/10.3390/a10040127>
44. Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S (2016) Traffic-sign detection and classification in the wild. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). <https://doi.org/10.1109/CVPR.2016.232>
45. Cubuk ED, Zoph B, Mane D, Vasudevan V, Le QV (2019) AutoAugment: learning augmentation strategies from data. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp 113–123. <https://doi.org/10.1109/CVPR.2019.00020>
46. Li C, Chen Z, Wu QMJ, Liu C (2019) Deep saliency with channel-wise hierarchical feature responses for traffic sign detection. *IEEE Trans Intell Transp Syst* 20(7):2497–2509. <https://doi.org/10.1109/tits.2018.2867183>
47. Zhang J, Xie Z, Sun J, Zou X, Wang J (2020) A cascaded R-CNN with multiscale attention and imbalanced samples for traffic sign detection. *IEEE Access* 8:29742–29754. <https://doi.org/10.1109/access.2020.2972338>

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