Traffic Sign Recognition Using Guided Image Filtering

Jiawei Xing and Wei Qi Yan(✉)

Auckland University of Technology, Auckland, New Zealand
wyan@aut.ac.nz

Abstract. In challenging lighting conditions, such as haze, rain, and weak lighting condition, the accuracy of traffic sign recognition is not very high due to missed detection or incorrect positioning. In this paper, we propose a traffic sign recognition algorithm based on Faster R-CNN and YOLOv5. Firstly, we conduct image preprocessing by using guided image filtering for the input image to remove noises. The processed images are imported into the neural networks for training and testing. The outcomes of the traffic sign recognition are promising.

Keywords: Traffic sign recognition · Faster R-CNN · GTSDB dataset · FRIDA database

1 Introduction

Traffic signs are everywhere to assist our driving in real traffic scenes, a rich assortment of traffic signs have been set on roadside as shown in Fig. 1. But only using our human visual systems is tough to eye these signs due to fast moving or weather conditions. Therefore, advanced driver assistance systems have become the focus of our attention [1–3].

At present, traffic sign detection algorithms have emerged and achieved satisfactory results [4, 5], but these algorithms mainly aim at digital images of traffic signs acquired under ideal weather conditions. Owing to the rapid development of road construction and transportations in recent years, the smoggy weather has been increased year after year. The collected images of traffic signs are blurred, the colors are faded, which in turn slash the recognition accuracy of these algorithms. In response to this issue, an accurate locating and recognition algorithm for traffic signs in haze weather is proposed in this paper.

Traffic sign recognition (TSR) was developed in the early 1980s and has taken a great step in the field of autonomous vehicles in 1987 [6]. It mainly targets at speed limit signs and takes use of classic algorithms based on image segmentation and template matching. The recognition process takes around 0.5 s on average. Due to hardware being developed at that time, the systems are not working in real time, the images are relatively small and cannot be integrated into real applications.

M. Nguyen et al. (Eds.): ISGV 2021, CCIS 1386, pp. 85–99, 2021.
https://doi.org/10.1007/978-3-030-72073-5_7

Fig. 1. Foggy traffic signs in various scenarios.

After the 1990s, with the continuous improvement of the hardware and its computing capability, advanced technology in digital imaging and computer vision has been emerged to take effects on discovering the principle of TSR. A variety of solutions have been proposed, such as edge extraction, color segmentation, feature vector extraction, artificial neural network, etc. In recent years, with the successful applications of deep learning [7, 8], such as semantic segmentation, etc., deep learning methods have been gradually brought into TSR.

The existing TSR algorithms generally have two key steps: Positioning and recognition. Because of the swift development of deep learning, in this paper, we propose a method for traffic sign recognition based on Faster R-CNN model.

The remaining part of the paper is organized as follows: The existing work is critically reviewed in Sect. 2. The proposed methods of this paper will be detailed in Sect. 3. The experimental results will be showcased and analyzed in Sect. 4. Our conclusion and future work will be presented in Sect. 5.

2 Literature Review

Traffic sign recognition has become a hot topic of current research. With the gradual completion of hardware design and implementations, two types of methods are taken into consideration. One is to extract visual features in digital image for pattern classification by using deep learning methods, the other is to extract visual features of objects for classification by using CNNs and RNNs.

A comprehensive scheme [9] was proffered for traffic sign recognition. Firstly, a cascade of trained classifiers is employed to scan the background quickly so as to locate a region of interest (ROI), then Hough transform is applied to shape detection. This method is evaluated based on an image database including 135 traffic signs. The average recognition speed is 25 frames per second (FPS), the recognition accuracy is 93%. In [10], edge detection was accomplished by using a combination of color filtering and closed curves. Through a neural network, the extracted features are applied to classify the targets. The average recognition rate is up to 94.9%. The nearest neighbors are applied to classify and recognize traffic signs by calculating Euclidean distance between a traffic sign and its standard template, then the image is classified according to the minimum distance.

Girshick et al. [11] proposed a rich feature hierarchical structure for precise target detection and semantic segmentation, i.e., R-CNN, using selective search (SS) [12] instead of traditional. The sliding window method extracts 2,000 target candidate regions on the image, then takes use of a deep convolutional network to classify the target candidate regions. However, because it carries out convolution operations on each candidate area instead of sharing calculations, the detection speed is slow, but with 47.9% segmentation accuracy. He et al. [13] proposed the spatial pyramid pooling network (SPPnets), which improves speed by sharing convolutional feature maps. Fast R-CNN [14] extracts convolutional feature maps, the training improves the detection accuracy and speed [15].

SSD [16] was set forth to detect traffic signs by using Inception v3 network instead of VGG16. For SSD, a random center point with a prior designed strategy was proposed. Douville, et al. [17] firstly normalized the image of traffic signs, then extracted Gabor features, finally a three-layer perceptron was employed to classify and recognize the traffic sign. A perceptual confrontation network was put forwarded for highway traffic sign detection [18], which combined Faster R-CNN with a generative confrontation network. The residual network is applied to learn the differences between the feature maps of small visual objects and large target objects so as to uplift the rates of highway traffic sign detection and recognition. The detection results have been achieved based on the Tsinghua-Tencent 100K dataset.

3 Network Design

Our idea for traffic sign recognition in this paper is depicted in Fig. 2. We firstly employ image processing to cope with foggy images for artifacts removal [31, 32], then import the preprocessed images into a neural network for object detection and classification.

Fig. 2. The pipeline for traffic sign detection and recognition

3.1 Guided Image Filtering

Image defogging is an important preprocess for haze removal, which enhances visual effects such as edges and contours. There are generally two types of image defogging algorithms, one is histogram equalization, which simply enhances the contrast of the image. The other is an image restoration-based defogging algorithm [19], which takes use of original images to compare with the foggy images so as to reconstruct the new image. The dehazing effect is salient, but it is difficult to achieve the quality of original image.

Image filtering can solve the drawbacks of the two dehazing algorithms. The algorithm adopts an image to guide and filter the target image so that the final output image roughly resembles to the target image, the texture is akin to the guiding image. The guiding or reference image can be either a different one or the same one as the input image. If the guiding image is equivalent to the input image, the filtering becomes an edge-preserving operation, which is able to be used for image reconstruction. By using visual features of the guided image filtering, haze image preprocessing for traffic signs achieved the results of image denoising, image smoothing, and fog removal. Therefore, we define the original image as p_i, l_i as the guiding image, and q_i as the output image. The relationship is linear as

$$q_i = a_k I_k + b_k \; i \in \omega_k \tag{1}$$

where a_k and b_k are specific factors, ω_k is a square window with a center point k, $i \in \omega_k$ guarantees that a_k is not too big. In order to ensure the guided image filtering has the best outcome, the difference between the original image and the output image needs to be minimized. Therefore, the cost function $E\ (a_k, b_k)$ is defined as

$$E(a_k, b_k) = \sum_{i,k \in \omega_k} \left(\left| (q_i - p_i)^2 - \varepsilon a_k^2 \right| \right) \tag{2}$$

The output is the best one if $E(a_k, b_k)$ is the smallest. We take use of the least square method to find a_k and b_k,

$$a_k = \frac{\frac{1}{|\omega|} \sum_{i \in \omega_k} I_i q_i - u_k \overline{p_k}}{\sigma_k^2 + \varepsilon}, b_{k=} \overline{p_k} - a_k u_k \tag{3}$$

where u is the mean of I in $\boldsymbol{W}$, σ is the variance of I in W, w is the number of pixels in the window. We input a_k and b_k into Eq. (1) and obtain,

$$q_i = \frac{1}{|\omega|} \sum_{i \in \omega_k} (a_k I_k + b_k) = \overline{a_i} I_i + \overline{b_i} \tag{4}$$

3.2 Improved Faster R-CNN

Convolutional networks usually include a convolutional layer and a pooling layer, where the convolutional layer is usually used to extract features from the target. The feature extraction network in Faster R-CNN is based on a convolutional neural network, which uses convolutional network and ReLU (Rectified Linear Unit) activation function to extract the features from the target image, then the extracted features are input into the RPN layer and ROI pooling layer, respectively.

Traditional methods may use sliding windows or selective search to generate detection windows. Faster R-CNN chooses RPN to generate the detection window. The network takes advantage of softmax function to determine the properties of anchor points (foreground or background). Then regression is employed to correct it. Finally, accurate proposals will be obtained.

(a) original picture (b) after the guided filtering

Fig. 3. Fog removal from digital images, (a) is the original picture from FROSI databases, (b) is the picture obtained by removing the fog after the guided filtering method

In Fig. 3, the RPN structure is framed by dotted lines. After 3 × 3 convolution, the feature map flows into two different channels, respectively. The upper one is classified by using the softmax layer to obtain foreground and background, the detection target is foreground. In order to obtain a relatively accurate proposal, the feature passes through the channel to calculate the offset of the regression. Finally, while removing the proposal that exceeds the boundary and the value is too small, the previous information is integrated to obtain a new proposal. With the network structure, the RPN layer basically completes the operation of locating the object.

The input of the ROI pooling layer is the proposals with various sizes. However, the input and output sizes of a convolutional neural network after training are fixed, which resize the proposals to the same.

In Faster R-CNN, we have fine-tuned our parameters, set the learning rate as 0.01, the momentum as 0.9, the batch size as 24, and the epoch as 200. The input features contain the proposal of classification network which is composed of fully connected layer and softmax activation function so as to obtain the predicted probability of each class the traffic sign belongs to. Faster R-CNN is shown in Eq. (5).

$$L((f_i),(l_i)) = \lambda \frac{1}{N_{reg}} \sum_i f_i^* L_{reg}\left(l_i, l_i^*\right) + \frac{1}{N} \sum_i L_{cls}\left(f_i, f_i^*\right) \tag{5}$$

where i represents the anchor index, f_i stands for the output probability of the softmax layer of positive samples, f^* means the corresponding prediction probability, l refers to the predicted bounding box, l^* denotes the GT box corresponding to the positive anchor.

Taken into account the advantages of Faster R-CNN, this paper adopts Faster R-CNN model to detect traffic signs. Faster R-CNN is use of VGGNet [20] as the backbone of the net. However, as the performance of the basic network improves, in this paper, we use GoogLeNet [21] for the feature extraction in our experiments. The network parameters are shown in Table 1.

After experimental verification, GoogLeNet has achieved the best results in terms of time-consuming and model performance based on the given dataset. During convolution, the kernels with various sizes were taken for the convolutional operations, the output feature maps are connected together.

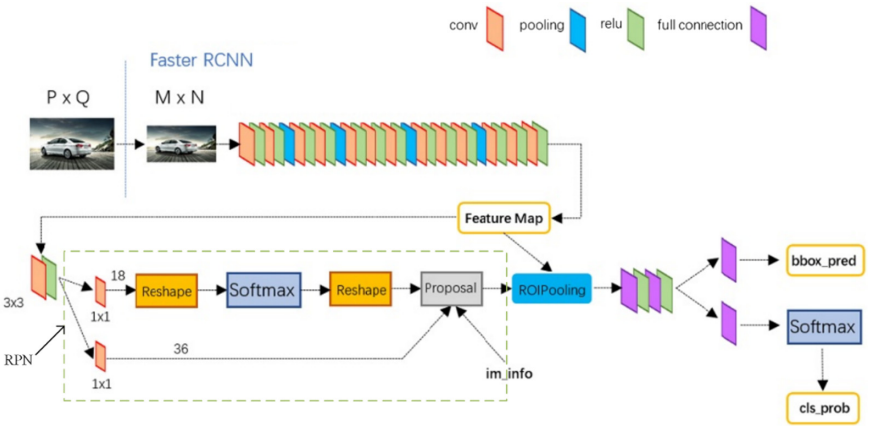

Fig. 4. The structure of faster R-CNN

Table 1. The parameters of GoogLeNet

Layer	Type	Size	Stride
1	Conv	(7, 7)	2
2	Max pooling	(3, 3)	2
3	Conv	(3, 3)	1
4	Max pooling	(3, 3)	2
5	Inception (a)		
6	Inception (b)		
7	Max pooling	(3, 3)	2
8	Inception (a)		
9	Inception (b)		
10	Inception (c)		
11	Inception (d)		
12	Inception (e)		
13	Max pooling	(3, 3)	2
14	Inception (a)		
15	Inception (b)		

Because the traffic sign will show multiple scales in the given image, after feature extraction, the signs with different scales are represented as features. We use cross-layer connection to improve the performance of multiscale target detection.

The detection net that we designed for the cross-layer connection is shown in Fig. 4. Pertaining to an image, CNN is accommodated to extract the features of the entire image. The RPN network is applied to extract a series of candidate regions based on the feature map. The change lies in the feature composition of the candidate region. This feature is no longer extracted by using only a single convolution layer, but is a fusion of features extracted from multiple convolution layers. The fused features contain not only semantic information but also local information.

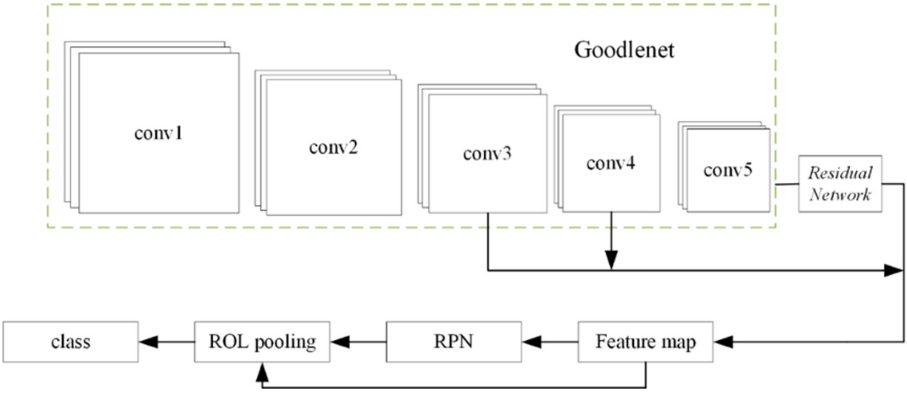

Fig. 5. The improved framework

In the given dataset, we often find a plethora of objects that resemble highway traffic signs. This will generate false detections. In order to achieve the purpose of reducing false detection, we use sample mining [22]. Firstly, the model is used to test on the training set. If there are negative samples with a score 0.8 or more in the obtained test results, they will be classified into a new sample class. In this way, the training set contains two classes: Traffic signs and traffic-like objects. The training set is used to obtain by mining negative samples so as to retrain a new detection model. Traffic signs are classified into the negative class and added to the training set, so that the model has increased the distinction between the two classes during the training time. This solves the problem that the model cannot classify background objects with minor differences between the positive class and the positive class if the amount of data is insufficient, thereby we obtain a satisfy outcome.

3.3 Improved YOLOv5

YOLO is a fast and compact open-source object detection model. Compared with other networks, it has stronger performance at the same size and has very good stability. The YOLO framework regards target detection as a regression problem, it is the first end-to-end neural network that predicts the class and bounding box of the object. At present, YOLOv5 has faster recognition speed and smaller network size than YOLOv4. For model training with different datasets, YOLOv3 and YOLOv4 need to calculate the initial anchor box, YOLOv5 embeds this function into the code to automatically

calculate the best anchor box. However, in practice, it was found that the clustering is deviated from the statistical results of the samples. Therefore, in this paper, the anchor box clustering algorithm is optimized and random correction processing is added:

$$\mathrm{W}_b = \mathrm{O}_2^3(\mathrm{random}[\mathrm{v1};\ \mathrm{v2}] \times \mathrm{w}_b) \tag{6}$$

where $O_2^3(\bullet)$ means that two of three cluster centers are randomly selected for correction, w_b is the width of the prior anchor box before correction, W_b is the width after correction. The numbers indicate the width and height of the anchor box respectively. It is observed that the minimum aspect ratio of the clustering results is 0.527 and the maximum is 0.714. For the data set in this paper, we find that 70% of the sample aspect ratios are between 0.72 and 1.00, 20% of the samples are between 0.6 and 0.7, 10% of the samples are between 0.6 and 0.7. From the analysis, we see that there is a deviation between the clustering results and the statistical results.

In traffic scenes, compared with pedestrians and vehicles, the physical size of traffic signs is smaller and there are at most three traffic signs in most samples. Because the ratio of foreground to background is severely unbalanced, most of the bounding boxes do not contain the target when the one-stage target detector is applied. Because the confidence error of these untargeted bounding boxes is relatively large, the loss of the foreground is submerged in the loss of the background. Therefore, in this paper, we optimize the original loss function. The main idea of optimization is to adaptively balance the loss of foreground and background. The loss function includes two parts, namely, regression loss and classification loss.

$$\begin{aligned} loss = & \sum_{i=0}^{S}\sum_{j=0}^{S}\sum_{k=0}^{B} E_{ijk}^{obj}\{\omega_{coord}[(x_{gt}-x_p)^2+(y_{gt}-y_p)^2+(\sqrt{w_{gt}}-\sqrt{w_p})^2+(\sqrt{h_{gt}}-\sqrt{h_p})^2]\} \\ & + \sum_{i=0}^{S}\sum_{j=0}^{S}\sum_{k=0}^{B}\{[\omega_{obj}E_{ijk}^{obj}(C_{gt}-C_p)^2]+[\omega_{noobj}E_{ijk}^{noobj}C_p(C_{gt}-C_p)^2]\}+\sum_{i=0}^{S}\sum_{j=0}^{S}\sum_{k=0}^{B}(P_{gt}-P_p)^2 \end{aligned} \tag{7}$$

where S is the width and height of the feature map. There are three sizes of the feature map in this paper: 52 × 52, 26 × 26, 13 × 13, B is the number of a priori boxes at each anchor box; E_{ijk}^{obj} represents the anchor point whether the box is responsible for predicting the target, E_{ijk}^{noobj} means it is irrelevant for predicting the target; $x_{gt,}$ $y_{gt,}$ w_{gt}, h_{gt} are ground truths, x_p, y_p, w_p, h_p are predicted values, indicating the coordinates of the target, width and height (in pixels); C_{gt} and C_p represent true confidence and prediction confidence, respectively; P_{gt} and P_p represent classification real probability and classification prediction probability, respectively; ω shows the weight coefficient of each loss part, for weight. The value is set in this paper as $\omega_{coord} = 5$, $\omega_{obj} = 1$, $\omega_{noobj} = 0.5$, the purpose of this setting is to reduce the loss of non-target regions and increase the loss of target ones in order to further avoid the loss of background values to confidence. In this paper, C_p is also used as a part of the weight to adjust the loss value of the background frame adaptively.

4 Results

4.1 Improved Faster R-CNN

In this paper, dataset GTSDB contains 900 images with a total of 1,206 traffic signs. There are four types of traffic signs: Mandatory, prohibitory, danger, and other. As there are not many foggy scenes in GTSDB, we also take use of FRIDA, FRIDA2, and FROSI databases. The dataset FRIDA consists of 90 composite images of 18 urban road scenes, meanwhile FRIDA2 consists of 330 composite images of 66 road scenes. They have the same driver's viewpoint, with different types of fogs added to each original image: Uniform fog, heterogeneous fog, cloudy fog, and cloudy heterogeneous fog, give way, watch out for pedestrians, etc. The FROSI dataset contains fog with the visibility ranging from 50 to 400 meters, including 1,620 traffic signs placed at various locations. By using these datasets, it is possible to train our YOLOv5 model comprehensively. In this paper, we combine two datasets for model training and testing. Among them, 60% samples are used for training, 20% for verification, and 20% for testing. In this paper, the evaluation index for traffic sign detection is measured by using mean average precision (mAP), which is popularly employed in object detection.

The test results include four categories: *TP*, *FP*, *FN*, *TN*. Precision is the probability that the positive sample is predicted correctly. Recall is for the primary positive samples, which indicates how many of the positive samples are predicted correctly. Therefore, the precision and recall are calculated as:

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

Firstly, we test various backbone networks. The performance of the network depends heavily on the ability of the network. Therefore, the part of feature extraction that directly affects network performance requires much effort. This paper offers classic networks as the feature extraction network of Faster R-CNN to compare the impact of different networks based on classification performance. Table 2 shows the experimental results of multiple networks. We see that the backbone networks have effects on the implemention. GoogLeNet and ResNet both have an improvement about 5.1% compared to VGGNET, whilst the running time of GoogLeNet is similar to that of VGG. Therefore, considered mAP and running time, Faster R-CNN as an object detector, GoogleNet is used as the backbone network.

Next, we tackle digital images with guided filtering and input the preprocessed image into the designed network to classify traffic signs. We compare the basic Faster R-CNN network. Table 3 shows the specific performance of our proposed method based on the given dataset. The PR curve of the experiments is shown in Fig. 5 (Fig. 6).

Table 2. The influence of feature extraction networks based on the detection results

Networks	Recall (%)	Precision (%)	mAP (%)	fps
VGGNet	88.2	89.1	90.2	16
GoogLeNet	88.7	93.2	95.3	17
ResNet	92.8	91.2	95.2	16

Table 3. Contrast experiments with the basic Faster R-CNN network

Methods	Recall (%)	Precision (%)	mAP (%)
Faster R-CNN	90.6	91.3	80.3
Our method	92.6	93.4	95.3

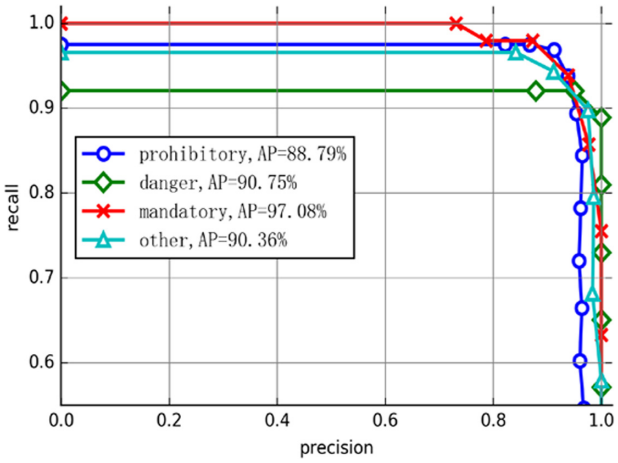

Fig. 6. Average precision (AP) of our experimental results

In Table 2, we compare the accuracy, recall and precision of the three feature extraction networks. We see that under the current scale of data training, the GoogLeNet is better than the VGG in recall and accuracy, but the running time will be relatively slower than the VGG network. Compared with ResNet, our recall rate is relatively low, other indicators are relatively better, but it takes a bit longer time.

In Table 3, the recall and accuracy of Faster R-CNN are relatively high. The reason is that there are a large number of traffic signs in reality. Accordingly, we took use of the guided image filtering for processing the image data. The feature fusion method based on the GoogLeNet is proposed in this paper for model training. Although the recall of target detection has not been changed too much, the accuracy has increased by 15%, which is explained as that by adding difficult negative samples, the net capability has been increased a lot owning to the data augmentation. Figure 5 shows the PR curves for four classifiers. In complex scenes, the general model usually cannot detect the traffic signs very well.

4.2 Improved YOLOv5

In this paper, we modify the YOLOv5 framework as the basis of the detection algorithm, and train two networks separately, one of which is the standard YOLOv5 network, which is used as a comparison method. The loss curve is shown in Fig. 7.

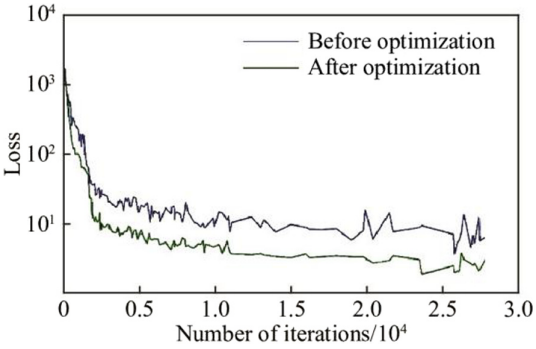

Fig. 7. The loss curves.

4.3 Comparsions of YOLOv5 and Faster R-CNN

These models are trained and evaluated based on a computer equipped with a Core i7-8th CPU, 16 GB of RAM, and NVIDIA RTX2060 GPU by using the same dataset. At first, we compare the training time. Faster R-CNN training took 14 h, while YOLOv5 training ran 11 h because YOLOv5 is a smaller size of network than Faster R-CNN. Secondly, the recognition speed of Faster R-CNN is 17 FPS, while the recognition speed of YOLOv5 is 60 FPS. YOLOv5 is much suitable for real-time traffic sign recognition. Finally, Fig. 8 and Fig. 9 show the results of the two classifications by using the FRIDA dataset.

Fig. 8. The results of traffic sign recognition based on FRIDA dataset by using YOLOv5

4.4 Guided Image Filtering

In this section, we use YOLOv5 as the basic framework to compare the results for dehazing. In Fig. 10(a), there is a traffic sign that is not able to be recognized.

Fig. 9. The results of traffic sign recognition based on FRIDA dataset by using faster R-CNN

(a) YOLOv5 without dehazing (b) YOLOv5 with dehazing

Fig. 10. The results of traffic sign recognition without (a) and with (b) dehazing

5 Conclusion

Our proposed method is based on Faster R-CNN to achieve traffic sign detection and recognition, we compare the results of object detection and recognition with multiple networks. If the overall framework of our experiment is the same, we choose the excellent network as our base net.

We have effectively employed multiresolution feature maps through cross-layer connections to build up the feature maps of traffic signs with multiple scales. We take use of guided image filtering to eliminate the noises from the given images, and further improve the accuracy of our experiments.

There are two aspects in our future work [23–30]. One is to collect more traffic signs as the samples under complicated conditions to form our own dataset. The other is to further optimize the method to form an end-to-end TRS system.

References

1. Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., Igel, C.: Detection of traffic signs in real-world images: the German traffic sign detection benchmark. In: International Joint Conference on Neural Networks, pp. 1–8 (2013)
2. Yang, Y., Luo, H., Xu, H., Wu, F.: Towards real-time traffic sign detection and classification. IEEE Trans. Intell. Transp. Syst. **17**(7), 2022–2031 (2016)
3. Berkaya, S.K., Gunduz, H., Ozsen, O., Akinlar, C., Gunal, S.: On circular traffic sign detection and recognition. Expert Syst. Appl. **48**, 67–75 (2016)
4. Jie, Y., Xiaomin, C., Pengfei, G., Zhonglong, X.: A new traffic light detection and recognition algorithm for electronic travel aid. In: International Conference on Intelligent Control & Information Processing (2013)
5. Jin, J., Fu, K., Zhang, C.: Traffic sign recognition with hinge loss trained convolutional neural networks. IEEE Trans. Intell. Transp. Syst. **15**(5), 1991–2000 (2014)
6. Priese, L., Klieber, J., Lakmann, R., Rehrmann, V., Schian, R.: New results on traffic sign recognition. In: IEEE Intelligent Vehicles Symposium (2002)
7. Sun, L., Chen, J., Xie, K., Gu, T.: Deep and shallow features fusion based on deep convolutional neural network for speech emotion recognition. Int. J. Speech Technol. **21**(4), 931–940 (2018). https://doi.org/10.1007/s10772-018-9551-4
8. Ren, Y., Yang, J., Zhang, Q., Guo, Z.: Multi-feature fusion with convolutional neural network for ship classification in optical images. Appl. Sci. **9**(20), 4209 (2019)
9. Ruta, A., Li, Y., Liu, X.: Detection, tracking and recognition of traffic signs from video input. In: Intelligent Transportation System, pp. 55–60 (2008)
10. Blancard, M.: Road sign recognition: a study of vision-based decision making for road environment recognition. In: Masaki, I. (ed.) Vision-Based Vehicle Guidance. Springer Series in Perception Engineering, pp. 162–172. Springer, New York (1992). https://doi.org/10.1007/978-1-4612-2778-6_7
11. Girshick, R., Donahue, J., Darrell, T.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE CVPR (2014)
12. Uijlings, R., van de Sande, A., Gevers, T., Smeulders, M.: Selective search for object recognition. Int. J. Comput. Vision **104**(2), 154–171 (2013)
13. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **37**(9), 1904–1916 (2014)
14. Girshick, R.: Fast R-CNN. In: IEEE International Conference on Computer Vision (2015)

15. Inoue, Y., Kohashi, Y., Ishikawa, N., Nakajima, M.: Automatic recognition of road signs. In: International Symposium on Optical Science & Technology (2002)
16. Müller, J., Dietmayer, K.: Detecting traffic lights by single shot detection. In: International Conference on Intelligent Transportation Systems (2018)
17. Douville, P.: Real-time classification of traffic signs. Real-Time Imaging **6**(3), 185–193 (2000)
18. Barnes, N., Zelinsky, A.: Real-time speed sign detection using the radial symmetry detector. IEEE Trans. Intell. Transp. Syst. **9**(2), 322–332 (2016)
19. Illingworth, J., Kittler, J.: A survey of the Hough transform. Comput. Vis. Graph. Image Process. **43**(2), 280 (1988)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. Comput. Sci. (2014)
21. Szegedy, C., et al.: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition (2014)
22. Sung, K.: Learning and example selection for object and pattern detection. Ph.D. thesis, Massachusetts Institute of Technology, United States (1995)
23. Liu, X., Yan, W.: Traffic-light sign recognition using Capsule network. MTAP (2021)
24. Qin, Z., Yan, W.: Traffic-sign recognition using deep learning. In: International Symposium on Geometry and Vision (ISGV) (2021)
25. Liu, X., Yan, W.: Vehicle-related scene segmentation using CapsNets. In: IEEE IVCZN (2020)
26. Pan, C., Yan, W.: Salient object detection based on perception saturation. Multimed. Tools Appl. **79**(27–28), 19925–19944 (2020)
27. Liu, X., Nguyen, M., Yan, W.: Vehicle-related scene understanding using deep learning. In: ACPR Workshop (2019)
28. Pan, C., Yan, W.: A learning-based positive feedback in salient object detection. In: IEEE IVCNZ (2019)
29. Yan, W.: Computational Methods for Deep Learning. Springer, Heidelberg (2021). https://doi.org/10.1007/978-3-030-61081-4
30. Yan, W.: Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-10713-0
31. Yan, W., Kankanhalli, M.: Detection and removal of lighting & shaking artifacts in home videos. In: ACM International Conference on Multimedia, pp. 107–116 (2002)
32. Yan, W., Kankanhalli, M., Wang, J.: Analogies-based video editing. Multimed. Syst. **11**(1), 3–18 (2005)