Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(7): 12240–12262.
DOI: 10.3934/mbe.2023545
Received: 06 February 2023
Revised: 10 April 2023
Accepted: 04 May 2023
Published: 18 May 2023

Research article

Research on rainy day traffic sign recognition algorithm based on PMRNet

Jing Zhang[1], **Haoliang Zhang**[1,*], **Ding Lang**[2], **Yuguang Xu**[1], **Hong-an Li**[1] **and Xuewen Li**[3]

[1] College of Computer Science and Technology, Xi'an University of Science and Technology, Xi'an 710054, China

[2] College of Energy, Xi'an University of Science and Technology, Xi'an 710054, China

[3] College of Safety Science and Engineering, Xi'an University of Science and Technology, Xi'an 710054, China

* **Correspondence:** Email: z15594674191@163.com.

Abstract: The recognition of traffic signs is of great significance to intelligent driving and traffic systems. Most current traffic sign recognition algorithms do not consider the impact of rainy weather. The rain marks will obscure the recognition target in the image, which will lead to the performance degradation of the algorithm, a problem that has yet to be solved. In order to improve the accuracy of traffic sign recognition in rainy weather, we propose a rainy traffic sign recognition algorithm. The algorithm in this paper includes two modules. First, we propose an image deraining algorithm based on the Progressive multi-scale residual network (PMRNet), which uses a multi-scale residual structure to extract features of different scales, so as to improve the utilization rate of the algorithm for information, combined with the Convolutional long-short term memory (ConvLSTM) network to enhance the algorithm's ability to extract rain mark features. Second, we use the CoT-YOLOv5 algorithm to recognize traffic signs on the recovered images. In this paper, in order to improve the performance of YOLOv5 (You-Only-Look-Once, YOLO), the 3 × 3 convolution in the feature extraction module is replaced by the Contextual Transformer (CoT) module to make up for the lack of global modeling capability of Convolutional Neural Network (CNN), thus improving the recognition accuracy. The experimental results show that the deraining algorithm based on PMRNet can effectively remove rain marks, and the evaluation indicators Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are better than the other representative algorithms. The mean Average Precision (mAP) of the CoT-YOLOv5 algorithm on the TT100k datasets reaches 92.1%, which is 5% higher than the original YOLOv5.

Keywords: image deraining; multi-scale residual; CoT module; traffic sign recognition

1. Introduction

With the popularization of various means of transportation and the rapid development of the road traffic system, the problem of traffic safety has become a severe challenge. The emergence of the Intelligent Transportation System (ITS) [1] has alleviated this problem to some extent. As an important component of Intelligent Transportation Systems, Traffic Sign Recognition (TSR) has been receiving increasing attention. Especially in the context of intelligent driving systems, traffic sign recognition technology can act in assisted driving [2], as well as automated driving of unmanned vehicles [3]. In the process of vehicle driving, traffic sign recognition technology can be used to identify road traffic signs in real time through intelligent devices. Based on this technology, the assisted driving system can promptly give the driver the corresponding prompt or warning. The automatic driving system can control the vehicle according to the recognition result to prevent traffic accidents. In addition, traffic sign recognition technology can be used by road maintenance personnel to maintain and check damaged and lost traffic signs [4], so as to improve road maintenance efficiency.

At present, the research on traffic sign recognition has made great progress. The main research methods include traditional physical model-based traffic sign recognition methods and deep learning algorithms. Among them, the method of feature extraction based on physical characteristics is an early research method, and its basic idea is to recognize traffic signs by analyzing their features such as color and shape. However, this approach may not work well for complex traffic signs and does not meet the real time requirement in practical applications. Deep learning algorithms [5–7] are a mainstream method in current traffic sign recognition research, and its main idea is to use convolutional neural networks (CNNs) for feature extraction and recognition of traffic signs. By training with a large amount of data, deep learning algorithms can automatically learn the optimal feature representation and overcome the limitations of traditional algorithms to some extent. However, in practical applications, there are many types of traffic signs, and there are differences in characteristics such as shape, color, and size. This poses challenges to the design and optimization of traffic sign recognition algorithms. We need to further improve the feature extraction capability of the network to meet the recognition accuracy requirements of the algorithm.

In addition, most of the traffic sign recognition algorithms do not consider the effect of weather on recognition accuracy. As a kind of weather we commonly see, pictures taken under rainy conditions are often affected by rain marks. Especially in the case of heavy rain, the rain masks will occlude the background scene, which will seriously degrade the visual quality of the captured image and degrade the performance of subsequent image processing tasks. Typical image deraining application scenarios include person tracking [8], object detection [9], semantic segmentation [10], and some other image processing tasks [11, 12]. At present, the model-based image deraining methods mainly use the prior information of rain marks to constrain the deraining model, and are solved by designing an optimization algorithm to obtain a clean image. However, the generalization ability of this type of algorithm is low, and there is a problem of incomplete removal of rain marks. Image rain methods based on deep learning are becoming more and more popular. These methods exploit deep networks to automatically extract hierarchical features. This method is able to simulate more complex mappings from rainy images to clean background images. Although this type of algorithm can obtain an improved deraining effect, there are still problems of insufficient feature extraction and too complicated network design. Therefore, it is necessary to explore some new technologies and methods to study how to effectively recognize traffic signs under rainy weather conditions. For example, preprocessing technology based on image enhancement and denoising, recognition method based on multi-scale and multi-feature fusion, etc. These technologies can help the recognition system obtain clearer images of traffic signs in rainy conditions, and improve the accuracy and stability of recognition.

The contributions of this paper are summarized as follows:

- A two-stage solution was designed for traffic sign recognition in rainy weather conditions. First, we proposed an image deraining algorithm to obtain clean images. Then, the optimized YOLOv5 was used for traffic sign recognition on the clean images.
- Regarding the image deraining module, we proposed a progressive multi-scale residual network for image deraining. Our network utilizes multi-scale residual structures and employs skip connections. Moreover, the proposed method works in a multi-stage manner, which can significantly improve the deraining performance of images.
- We optimized the feature extraction module of YOLOv5 for traffic sign recognition. We replaced the 3×3 convolution in the C3 module with the CoT module. This module enhances the global feature representation while retaining the local feature extraction ability of the CNNs, thereby improving the accuracy and robustness of the algorithm.

2. Related works

Traditional approaches mainly use a model-driven approach [13], which is particularly concerned with adequately encoding the physical properties and a priori information of rain traces and background images into an optimized model, and designing reasonable algorithms to solve these cases. In terms of traditional model-driven deraining algorithms, Deng et al. [14] developed a global sparse model for rain masks removal, which takes into account the inherent characteristics of rain streaks such as directionality, structural knowledge, and background information. Wang et al. [15] proposed a rain convolutional dictionary model (RCDNet) and utilized the proximal gradient descent technique to design an iterative algorithm only containing simple operators for solving the model. These model-based techniques still lack the adapting ability to rain marks and backgrounds with complexities, and often require time-consuming iterative computations. Such algorithms often suffer from efficiency problems in practical applications. Recently, deep learning techniques have been rapidly developed in the field of image restoration and related techniques have been applied to image deraining tasks [16]. This class of methods is a data-driven model that learns a nonlinear mapping from images with rain to images without rain in an end-to-end manner. Li et al. [17] proposed a recurrent structure network (RESCAN) combined with the channel attention mechanism to obtain multi-level features of rain marks in different directions to remove rain. Ren et al. [18] designed a multi-stage deraining network. Progressive recurrent network (PReNet) consists of a simple network model based on residual networks, but each stage of this method uses only common residual blocks and cannot extract deeper features. Zamir et al. [19] used encoder-decoder structure on the basis of a multi-stage to extract more contextual information. Although the above methods can obtain an improved visual quality, the restoration results will still leave either rain marks or local blur.

To solve these problems, we propose a two-stage rainy traffic sign recognition algorithm. Firstly, this paper presents a progressive multi-scale residual deraining model in cases where the physical information of rain streaks is more complex, such as direction, shape, and density. Current single-stage and single-scale convolutional network cannot completely remove the rain marks. We employ a multi-stage network to address the incomplete removal of rain marks. Furthermore, a multi-scale residual structure is adopted to extract features of rain marks and solve the problem of insufficient image detail restoration. The Convolutional long-short term memory (ConvLSTM) [20] adopted in each stage can capture the global information of the image to enhance our feature extraction ability.

Traditional traffic sign recognition algorithms mainly use image processing techniques to extract and classify features such as color, shape, and edges of images [21]. However, traditional methods still struggle to achieve the balance of real-time and accuracy that CNNs can achieve. In recent years the use of convolutional neural networks for target detection and recognition has become a mainstream approach. The method is mainly divided into two categories. The first one is a two-stage target detection algorithm represented by the R-CNN series. This series of algorithms will first form a region proposal and then process the region proposals to get the final result. The Faster R-CNN algorithm truly implements an end-to-end computation, which uses Region Proposal Networks (RPN) instead of Selective Search to generate region proposals. Li et al. [22] proposed a recognition algorithm combining Faster R-CNN with an attention-guided context feature pyramid network (AC-FPN) in order to improve the recognition accuracy of traffic signs. The Mask R-CNN algorithm adds a segmentation head to Faster R-CNN to generate a mask for each candidate region. Tabernik et al. [23] evaluated different traffic sign datasets using the Mask R-CNN algorithm. Although the two-stage algorithm can obtain a better recognition accuracy, due to the computational complexity of the algorithm itself, its recognition speed is slow and cannot meet the real-time requirements. The second category is the single-stage target detection algorithm represented by YOLO [24] and Single Shot MultiBox Detector (SSD) [25] algorithms. Wu et al. [26] combined SSD with Receptive Field Module (RFM) and Path Aggregation Network (PAN) and applied it to the recognition of traffic signs, but the SSD algorithm was not effective in detecting small targets. YOLOv3 [27] used darknet-53 as the backbone network and used multiple scales for prediction to solve the problem of detection and identification of multi-scale targets. YOLOv4 [28] uses Mosaic enhancements on top of the original YOLO, and the backbone network uses CSPDarknet-53. YOLOv5 [29] innovatively uses the structure of Focus, and combines the advantages of the previous version. It has advantages in recognition speed and accuracy. Recently, Dosovitskiy et al. [30] applied the Transformer method [31, 32] in the field of natural language processing to the field of computer vision for the first time. Hunag et al. [33] proposed a novel Transformer-based Cross Reference Network (TCRN), which fully exploits long-range context dependencies in both feature representation extraction and cross-modal integration. This method makes up for the lack of global understanding of images in CNNs-based methods. However, pure Transformer networks have high computational complexity and slow training process.

To address the problems of the current algorithm, we choose YOLOv5 as the backbone network for our traffic sign recognition. This algorithm not only makes up for the shortcomings of traditional algorithms in real-time, but also maintains a better recognition accuracy. The current deep learning algorithms based on CNNs suffer from inadequate feature extraction. This type of method does not sufficiently consider the global modeling capability of the network. This can lead to poor robustness of the algorithm. For this problem, we adopt a module combining traditional CNNs and Transformer. This method enhances the global expressive ability of the network, and preserves the inference speed and local feature extraction ability of CNNs. Compared with the current method, the improved YOLOv5 achieves better performance in the recognition of traffic signs.

3. Our approach

3.1. Progressive multi-scale residual deraining algorithm

Usually the input of a rainy day image can be expressed as $O \in R^{H \times W}$, H and W denote the height and width of the image, respectively. The rainy image model we often use is shown in Eq (3.1):

$$O = B + R \tag{3.1}$$

where B and R denote the background layer and rain layer of the rainy day image, respectively. The goal of progressive multi-scale residual-based image rain removal networks is to design a reasonable network architecture to learn a nonlinear mapping function from an input rainy image image O to its background layer B or residual rain layer R.

3.1.1. Progressive multi-scale residual network structure

The structure of the progressive multi-scale residual network is shown in Figure 1.The network adopts T recursive stages, and requires multiple stages to share the same network parameters to gradually remove rain marks in the image.

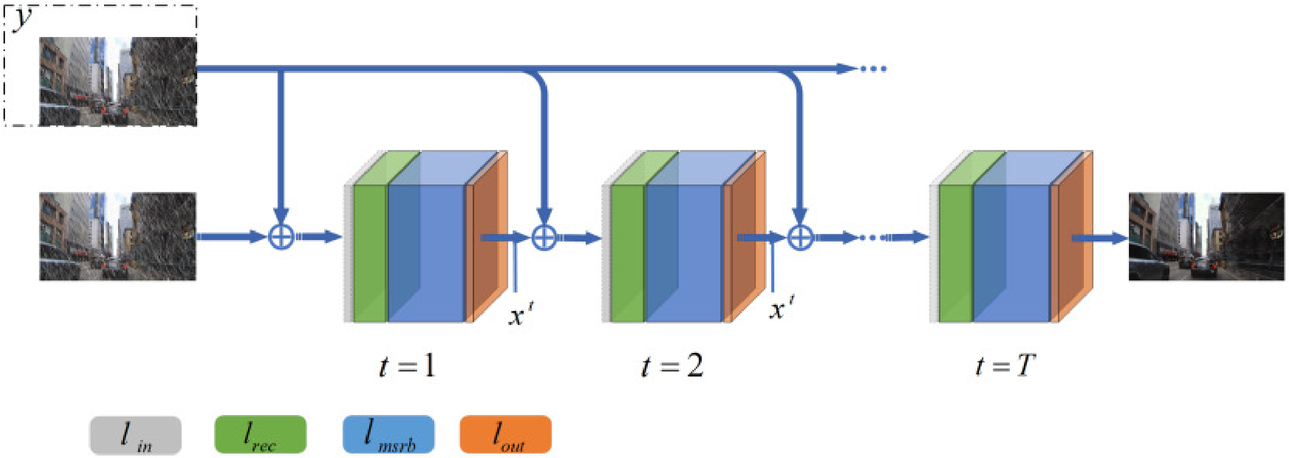


Figure 1. PMRNet model structure.

Each stage mainly consists of four parts:

1) The first layer l_{in} is a convolutional layer with Relu activation function, which is used to receive the input of the network.
2) The second layer is recursive layer l_{rec}. We use a Convolutional LSTM network to capture the global texture features recursively, so that we can obtain complementary and redundant information in the spatial dimension to represent the target rain marks. All convolutions of ConvLSTM have 32 input channels and 32 output channels.

3) The third layer l_{msrb} consists of five multi-scale residual modules to extract deep-level features.
4) The last layer l_{out} is a convolutional layer to output the result after image deraining. The size of all convolutions in the network is 3 × 3, and the padding is 1 × 1.

The formulation of a progressive multi-scale residual network with t stages can be expressed as Eq (3.2).

$$\begin{aligned} x^{t-0.5} &= l_{in}(x^{t-1}, y), \\ v^{t} &= l_{rec}(v^{t-1}, x^{t-0.5}), \\ x^{t} &= l_{out}(l_{msrb}(v^{t})) \end{aligned} \tag{3.2}$$

Among them, l_{in}, l_{out}, l_{rec}, and l_{msrb} of each stage remain unchanged, that is, the network parameters are reused in different stages. l_{in} takes the prediction x^{t-1} of the current stage and the rainy day image y together as the input of the network. The addition of y can further improve the performance of network deraining compared to just using it as an x^{t-1} input. l_{rec} takes the output l_{in} from the same stage, as well as the state v^{t-1} of recurrent unit from the previous stage, as inputs for the current stage.

3.1.2. Multi-scale residual module

Both the ordinary residual module, as well as the densely connected residual structure [34], use only a single size convolutional kernel, and as the network deepens the densely connected approach causes the computational complexity to grow at a high growth rate. To address these drawbacks, we adopt a multi-scale residual structure. Based on the residual structure, we introduce convolution kernels of different sizes for adaptively detecting image features at different scales. Additionally, skip connections are used between features at different scales so that the feature information can be shared and reused. This helps to take full advantage of the local features of the image. In addition, the 1 × 1 convolutional layer at the end can be used as a bottleneck layer as a way to facilitate feature fusion and reduce the computational complexity to ease the training. As shown in Figure 2, our multi-scale residual structure consists of two parts: multi-scale feature fusion and local residual learning.

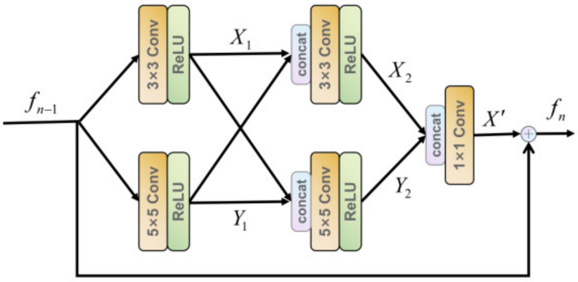


Figure 2. Structure diagram of multi-scale residual module.

Multi-scale feature fusion: we construct a two-bypass network and different bypass used different convolutional kernel. In this way, the information between these bypass can be shared with each other to be able to detect image features at different scales. The formula for this structure can be expressed as Eq (3.3).

$$
\begin{aligned}
X_1 &= \varepsilon(c^1_{3\times3} * f_{n-1} + b^1),\\
Y_1 &= \varepsilon(c^1_{5\times5} * f_{n-1} + b^1),\\
X_2 &= \varepsilon(c^1_{3\times3} * [X_1, Y_1] + b^2),\\
Y_2 &= \varepsilon(c^1_{5\times5} * [X_1, Y_1] + b^2),\\
X^{'} &= c^1_{1\times1} * [X_2, Y_2] + b^3
\end{aligned}
\tag{3.3}
$$

where c and b represent the weight and bias, respectively, the superscript represents the number of layers they are in, and the subscript represents the size of the convolution kernel used for that layer. $\varepsilon(x)$ represents the ReLU activation function and $[X, Y]$ represents the skip connection operation.

Local residual learning: To make the network more efficient, we used residual learning for each multi-scale module. We can formulate the multi-scale residual block (MSRB) as follows:

$$f_n = X^{'} + f_{n-1} \tag{3.4}$$

where f_n and f_{n-1} represent the input and output of the multi-scale residual module, respectively. It is worth mentioning that the local residual learning has a good improvement in network performance while maintaining a low computational complexity.

3.1.3. Loss function

Mean square error (MSE) [35] is a commonly used loss when training networks, however, the traditional MSE-based loss is not sufficient to express the human visual system's intuitive perception of a picture. In this paper, we adopt negative Structural Similarity Index Measure (SSIM) as the loss function of our rain-removing network. The SSIM loss function takes into account luminance, contrast, and structure metrics, which takes into account human visual perception. For a progressive multi-scale residual network with one T stage, if we monitor the output x^T of the final stage, the negative SSIM loss can be expressed as:

$$L = -SSIM(x^T, GT) \tag{3.5}$$

where GT is the corresponding ground-truth clean image. To achieve better training results, we supervise the intermediate results at each stage using SSIM loss, with the expression :

$$L = \sum_{t=1}^{T} \lambda_t SSIM(x^T, GT) \tag{3.6}$$

where λ_t is the tradeoff parameter for stage t. Of course, there are now many methods that use hybrid loss functions, such as mixing MSE with SSIM. Complex loss functions have better performance and can better supervise the learning process of the network; however, the more complex the loss function is, the more difficult it is to tune the hyperparameters. We find that a single negative SSIM loss function is sufficient to complete the training of our deraining network well.

3.2. Traffic sign recognition algorithm

The network structure diagram of YOLOv5 is shown in Figure 3, which is divided into three parts: Backbone (backbone network), Neck (multi-scale feature fusion network), and Head (predictive classifier).

Backbone: This structure is the backbone part of the network and is used to extract recognition features of images, including edge features, texture features, location information, etc. The backbone network in YOLOv5 is mainly composed of the Focus layer, wrapped convolutional layer CBS, C3 layer, SPP layer, and other structures. The Focus layer uses a slicing operation to split the high-resolution map into multiple low-resolution feature maps. The CBS module consists of convolution, normalization and Leaky Relu activation functions. C3 networks aim to reduce the model size to increase inference speed while maintaining accuracy. In addition, SPP modules refer to spatial pyramid modules, which perform maximum pooling by different sized kernels and complete fusion by concatenating features.

Neck: This structure is the fusion part of the network that mixes and combines the features and passes them to the prediction layer. The combination of an FPN structure with top-down delivery of strong semantic features in order to improve low-level feature ground propagation and a bottom-up feature pyramid containing two PAN structures operates to enhance the network feature fusion. The FPN structure that transfers strong semantic features from top to bottom is used to improve the propagation of low-level features, and the bottom-up feature pyramid containing two PAN structures is combined to enhance the ability of network feature fusion.

Prediction: This part is to predict and classify the corresponding category probabilities, target confidence, and prediction frame coordinates on three scales of 20×20, 40×40, and 80×80 feature maps using three 1×1 convolutional layers instead of fully connected layers.

In the output part, YOLOv5 uses GIoU [36] as the loss function, and also filters the target box by non-maximum suppression NMS [37].

3.2.1. CoT-YOLOv5 algorithm

CNNs are widely used in a variety of tasks due to their powerful visual representation learning capabilities. This structure of CNNs for local information modeling makes full use of spatial locality and translational equilateralism. But again, by only being able to model local information, CNNs lack the ability to model and perceive over long distances, which is important in many vision tasks. YOLOv5 is an efficient target detection algorithm, but it still has some false and missed detection problems for object detection in complex scenes. Transformer-style algorithms show strong global modeling capabilities for visual tasks and can handle complex scene information and relationships between objects very well. Combining it with YOLOv5 can help improve the accuracy of target detection.

Therefore, in this paper we use a CoT (Contextual Transformer) module, which combines the Transformer and CNNs. This module combines the dynamic context information aggregation of the self-attention mechanism in Transformer and the static context information aggregation of CNNs. As shown in Figure 3, we fuse the CoT modules into the backbone structure of the network. $CoT3$ represents the C3 module combined with CoT. The introduction of the CoT module can enhance the robustness of the model and have better adaptability to some complex traffic scenes and noise data. Additionally, the attention mechanism of the CoT module has some interpretability, which can better understand the prediction results of the model.

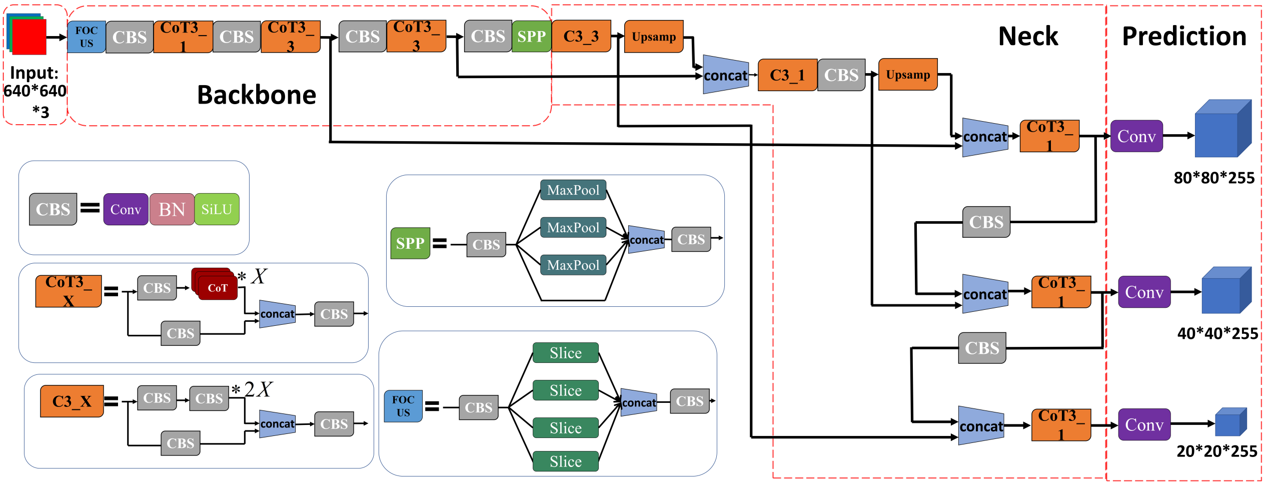


Figure 3. CoT-YOLOv5 network structure.

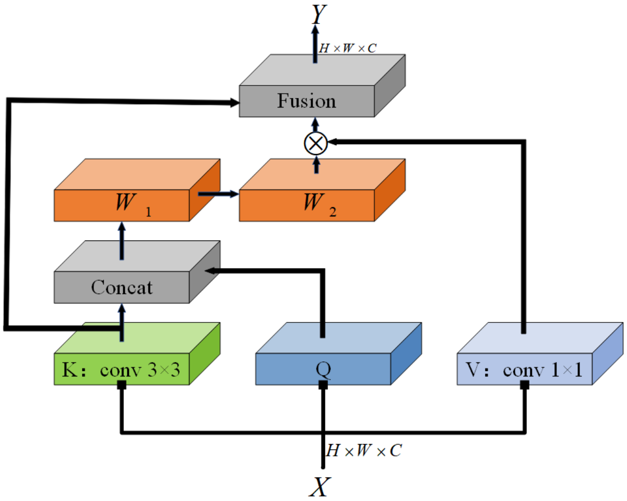


Figure 4. Contextual Transformer block.

The structure of the CoT module is shown in Figure 4, assuming that the input of an incoming two-dimensional feature map $X \in R^{H \times W \times C}$ (H: height, W: width, C: number of channels). The Key, Query, and Value can be defined as, $K = X$, $Q = X$, and $V = XW_V$, respectively. Where W_V is the embedding matrix, we use a 1×1 convolution to make it relational mapping. First, a group convolution is performed on K with a 3×3 convolution to obtain K^1 with local contextual information representation, and this can be seen as a static modeling on the local information. Then, connect K^1 and Q to obtain the attention matrix through two consecutive 1×1 convolutions (with ReLU activation function W_1 and W_2 without activation function), the expression is as follows:

$$A = [K^1, Q]W_1W_2 \tag{3.7}$$

In order to obtain the dynamic context information of the input, the attention matrices A and V are multiplied, and the expression is as follows:

$$K^2 = A \otimes V \tag{3.8}$$

where ⊗ represents the local matrix multiplication operation, and finally we fuse the local static context information with the global dynamic context information to obtain the final output Y.

The special mechanism of the CoT module allows global modeling of the entire image and retains the ability to model local information. This helps the model to better understand information such as the relationships and positions of objects in the image. The multi-head attention mechanism in the CoT module can extract multiple hierarchical representations of features and integrate these representations to better extract target features in images. This is very important for object detection tasks. Finally, using the residual connection technique with 1 × 1 convolution can avoid the gradient disappearance and gradient explosion problems in deep networks, thus improving the training efficiency and robustness of the model.

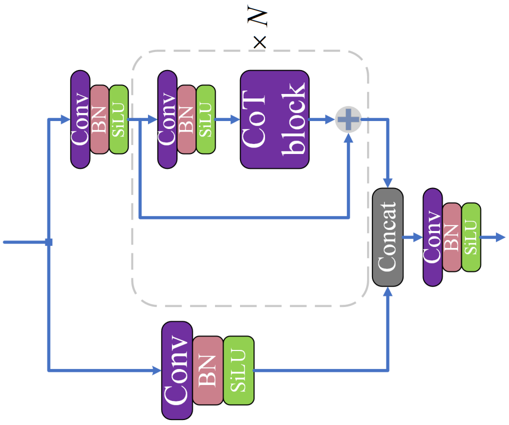


Figure 5. C3 module with a CoT block.

In order to improve the global expression ability of the backbone network, we replace all the 3 × 3 element convolutions in the C3 module with the CoT module, and the optimized C3 module structure is shown in Figure 5. By fusing the CoT and C3 modules, the backbone network is able to extract more target features from the images. Other basic modules in the C3 structure are composed of convolutional layers, BN layers, and SiLU activation functions. This combination of multiple technologies can better capture the relationship between targets and improve the accuracy of traffic sign recognition. Since the number of input and output channels of the CoT module remains unchanged, this method can improve the feature extraction capability of the backbone network without increasing the parameters. It can effectively improve the accuracy and efficiency of traffic sign recognition, and also has good interpretability, robustness and flexibility.

4. Experiment

4.1. Datasets

Deraining Datasets: For the image deraining part, we mainly used two benchmark datasets: Rain100H and Rain800, both of which are artificial synthetic datasets. Rain100H [38] is a heavy rain datasets that contains five types of rain marks. The rain-free background images of the Rain800 [39] datasets are from the UCID and BSD-500 datasets, and rain marks of different densities are added on the background. These two datasets are often used as comparison datasets.

Traffic sign recognition datasets: For the traffic sign recognition part, we use the TT100K [40] traffic sign datasets, which contains five different cities. The data set has too little data for some categories, so the data is unbalanced. Full training will lead to overfitting, so we select 45 types of traffic signs that meet the requirements for training, and we use 9150 images for training and 1120 for testing.

Figure 6 shows the classification of TT100K traffic signs. The signs in yellow, red, and blue boxes are warning, prohibition, and mandatory signs, respectively. Each traffic sign has a unique label. Besides, in order to validate the actual traffic sign recognition scenario, we selected a part of TT100K data assembled into 8000 images with rain marks and tested them based on our deraining model.The CPU used in this experiment is Intel(R) Core(TM) i9-9820X CPU @ 3.30 GHz and the GPU is NVDIA GeForce RTX 2080Ti.

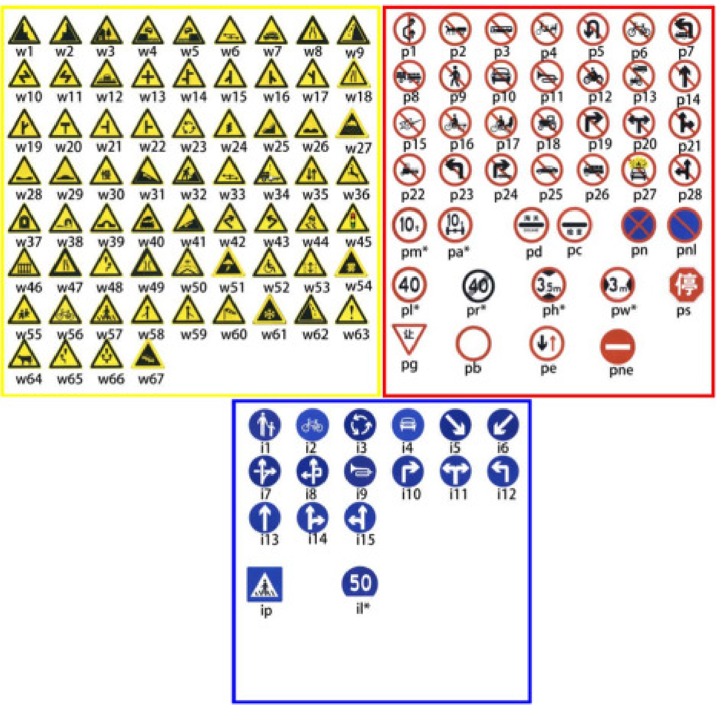

w1 w2 w3 w4 w5 w6 w7 w8 w9
w10 w11 w12 w13 w14 w15 w16 w17 w18
w19 w20 w21 w22 w23 w24 w25 w26 w27
w28 w29 w30 w31 w32 w33 w34 w35 w36
w37 w38 w39 w40 w41 w42 w43 w44 w45
w46 w47 w48 w49 w50 w51 w52 w53 w54
w55 w56 w57 w58 w59 w60 w61 w62 w63
w64 w65 w66 w67
p1 p2 p3 p4 p5 p6 p7
p8 p9 p10 p11 p12 p13 p14
p15 p16 p17 p18 p19 p20 p21
p22 p23 p24 p25 p26 p27 p28
pm* pa* pd pc pn pnl
pl* pr* ph* pw* ps
pg pb pe pne
i1 i2 i3 i4 i5 i6
i7 i8 i9 i10 i11 i12
i13 i14 i15
ip il*

Figure 6. TT100K traffic sign category.

4.2. Experiments of the deraining algorithm

Evaluation metrics. In this paper, we adopt the most common Peak Signal-to-Noise Ratio (PSNR) [41] and Structural Similarity (SSIM) [42] as quantitative indicators of model performance. SSIM evaluates the similarity of two images by brightness, contrast and structure, and the formula is as follows:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + 1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2\sigma_y^2 + C_2)} \tag{4.1}$$

where $\mu_*, \sigma_*^2, \sigma_{xy}$ are the mean,variance, and covariance of x, y respectively. C_1, C_2 are constants. PSNR is mainly used to measure the image distortion or noise level, and the expression is as follows:

$$PSNR(x,y) = 10 \times \log_{10}\left(\frac{MAX_1^2}{MSE}\right) \tag{4.2}$$

MAX_1 indicates the maximum value that represents the color of the image point and MSE is the mean square error. PSNR is measured in dB, where the larger the value, the smaller the distortion.

Implementation details. To highlight the advantages of our model, we set the number of stages T of PMRnet to 5. We use Adam as the model optimizer. We use SSIM loss to supervise the intermediate results of each stage. The initial learning rate is set to 10^{-3} and the total number of training epochs is set to 100.

We discuss the effect of loss functions on the deraining performance, including MSE loss, negative SSIM loss, and MSE + SSIM hybrid loss functions. We have trained each of the above three loss functions based on PMRNet. Table 1 lists their PSNR and SSIM values on Rain100H. It can be seen that PMRNet-SSIM outperforms PMRNet-MSE in terms of SSIM as well as PSNR. We also noticed that the two indicators of PMRNet-MSE+SSIM only increased slightly, but this method greatly increased the burden of hyperparameter tuning. Therefore, a single negative SSIM loss function is sufficient to train our model. In the following experiments, the negative SSIM loss is adopted as the default value.

Table 1. Comparison of PMRNet with different loss functions.

Loss	PMRNet-MSE	PMRNet-SSIM	PMRNet-MSE + SSIM
PSNR	27.53	28.76	**28.73**
SSIM	0.880	0.901	**0.905**

We give comparison images of the deraining effect of some other methods and our method on the Rain100H and Rain800 datasets, where (a) is the input rain image, (b) is the deraining effect of RESCAN, (c) is the deraining effect of MPRNet, (d) is the deraining effect of PreNet, and (e) is the deraining effect of the method shown in this paper.

Table 2. Experimental results of Rain100H datasets.

Method	SSIM/dB	PSNR/dB
RESCAN [17]	0.795	26.45
PreNet [18]	0.881	27.61
MPRNet [19]	0.890	28.42
PMRNet (our)	**0.901**	**28.76**

It can be seen from Figure 7 that there will be more rain marks remaining in the results of RESCAN on the Rain100H datasets, although MPRNet and PreNet can remove most of the rain marks, there will still be some unremoved rain and the loss of details. Additionally, it can be seen from Tables 2 and 3 that our method outperforms the other three methods in both SSIM and PSNR evaluation results on Rain100H and Rain800 data.

Table 3. Experimental results of Rain800 datasets.

Method	SSIM/dB	PSNR/dB
RESCAN [17]	0.821	25.16
PreNet [18]	0.774	26.78
MPRNet [19]	0.851	27.51
PMRNet(our)	**0.874**	**28.76**

Figure 8 is the comparison effect on the Rain800 datasets. It can be seen that PreNet left more rain marks on the Rain800, and did not remove most of the rain marks like the results on the Rain100H. The optimal performance of our method on different types of datasets indicates that our method has good robustness.

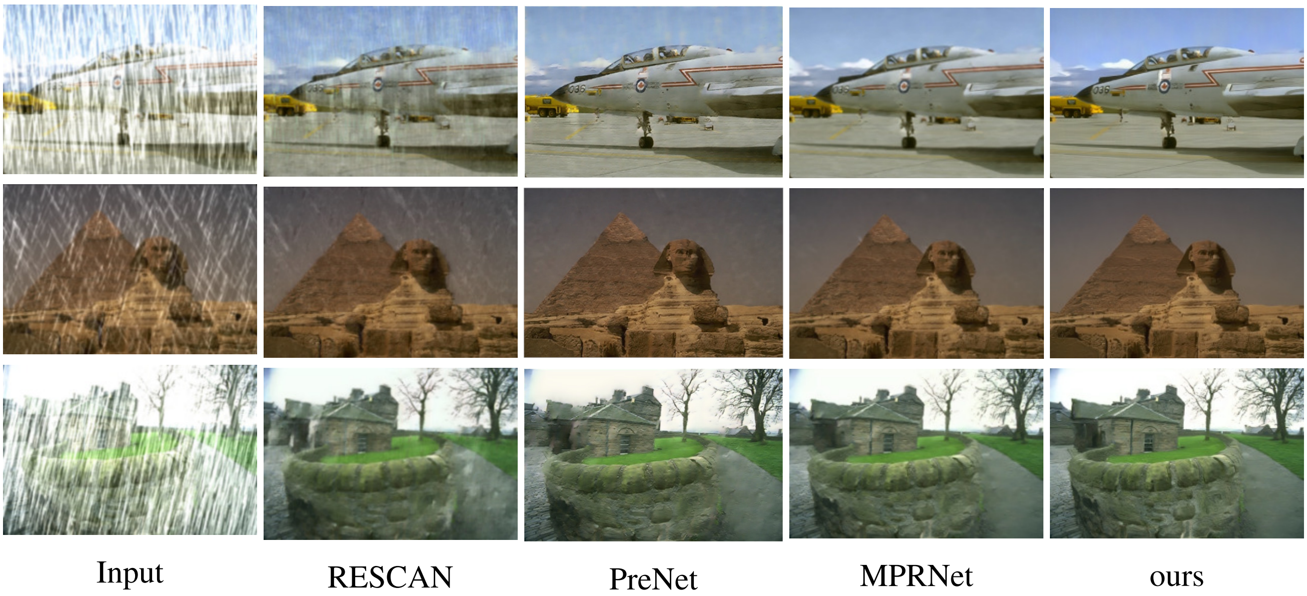


Figure 7. Comparison results on the Rain100H datasets.

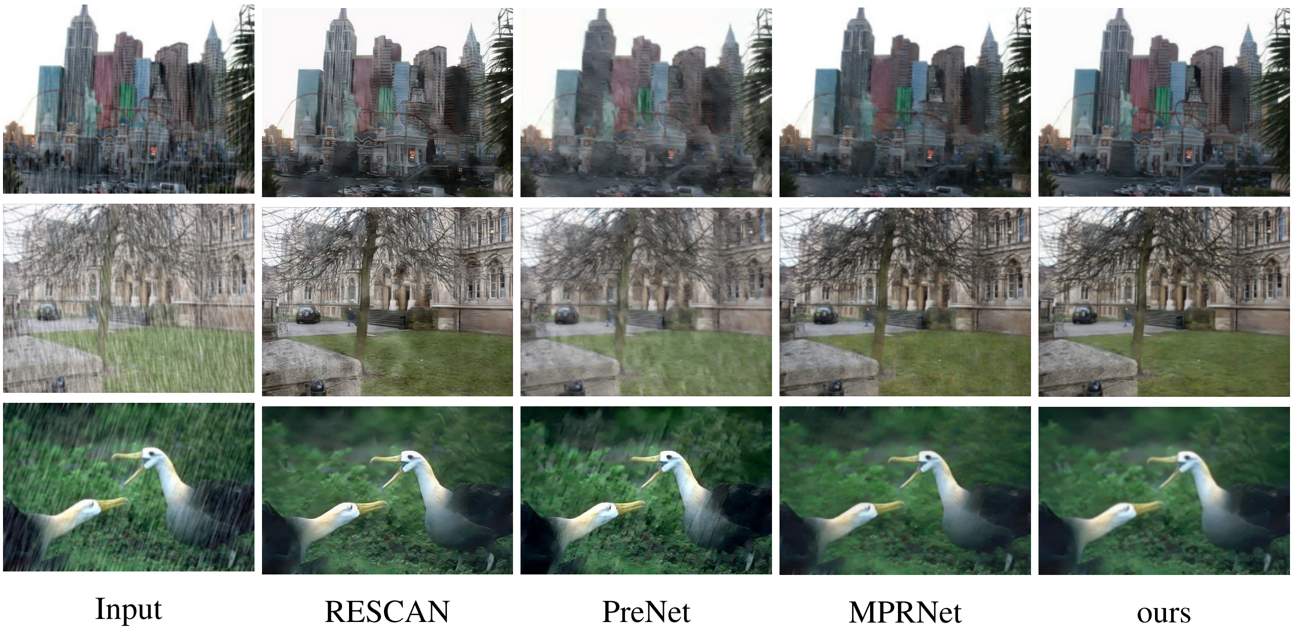


Figure 8. Comparison results on the Rain800 datasets.

To test the rain removal effect of PMRNet in a traffic sign recognition scenario, we tested the synthetic rain datasets based on TT100K, and Figure 9 shows our test results. Additionally, two common metrics were verified, SSIM reached 0.943 and PSNR reached 29.50, which shows that PMRNet has a very good performance on our synthetic datasets.

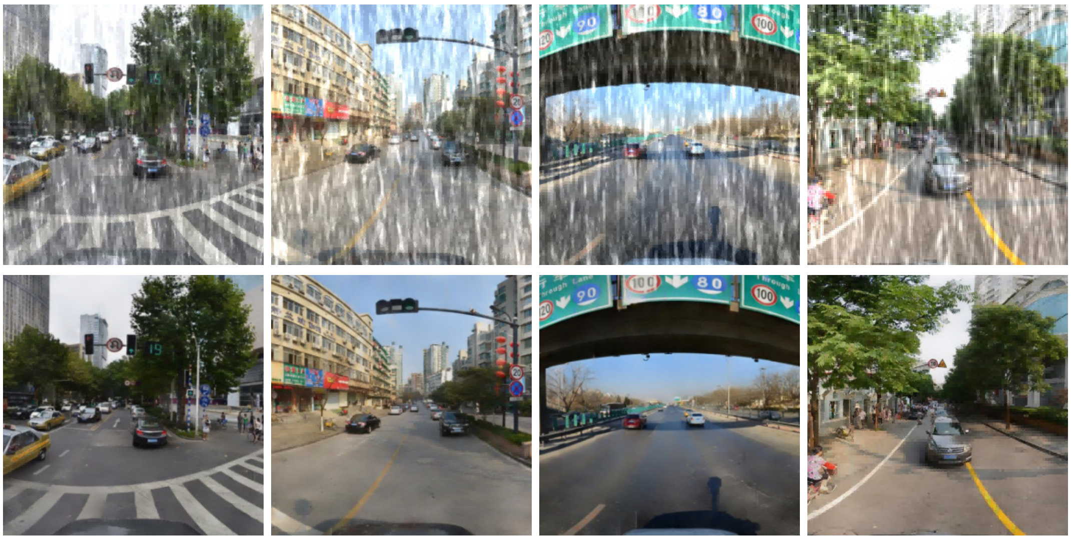


Figure 9. PMRNet (our) test results on TT100K-rain datasets.

Table 4. Comparison of PMRNet models with different stages.

Method	$PMRNet_1$	$PMRNet_2$	$PMRNet_3$	$PMRNet_4$	$PMRNet_5$	$PMRNet_6$
PSNR/dB	21.38	26.92	28.16	29.26	**29.50**	29.49
SSIM/dB	0.842	0.920	0.937	0.940	**0.943**	0.943

Table 4 shows the SSIM and PSNR of the PMRNet model with stages T = 1, 2, 3, 4, 5, 6. PMRNet leads to higher SSIM and PSNR as the number of stages grows. Both values will stop growing when T = 5. Larger T also makes PRMNet more difficult to train, so T = 6 is the current optimal number of stages.

4.3. Experiment of traffic sign recognition algorithm

Evaluation metrics. This part of the experiment uses average recall(AR), mean average precision (mAP), number of model parameters, and recognition speed as evaluation metrics for model evaluation results. The formulas of precision and recall are shown in Eqs (4.3) and (4.4).

$$precision = \frac{TP}{(TP + FP)'} \tag{4.3}$$

$$recall = \frac{TP}{(TP + FN)'} \tag{4.4}$$

where TP denotes the number of samples that are actually positive cases and predicted by the classifier as positive cases, FP denotes the number of samples that are actually negative cases but predicted by the classifier as positive cases, and FN denotes the number of samples that are actually positive cases but predicted by the classifier as negative cases.

AP is used to evaluate the strengths and weaknesses of the model in each category. The region enclosed by the accuracy and recall is called the PR curve. The result is calculated using the integral, as shown in Eq (4.5).

$$AP(n) = \int_0^1 p(r_n)dr_n \tag{4.5}$$

where n denotes the category, r_n denotes the recall belonging to category n, and $p(r_n)$ denotes the precision in the PR curve corresponding to category n.

The mAP is the average of the AP values of all categories, which can reflect the detection performance of the model on the whole dataset. The formula is shown in Eq (4.6).

$$mAP = \frac{1}{N}\sum_{n=1}^{N} AP(n) \tag{4.6}$$

where N represents all categories.

Implementation details. Since the image size in TT100K dataset is 2048 × 2048 pixels, which is not conducive to training. Therefore, we set the input size to 640 × 640. Adam as a model optimizer.

We supervise the training using the GIOU loss function, which is used to measure the degree of overlap and proportional differences between the predicted and true frames. The batch size is set to 16 and the momentum is 0.9. The initial learning rate is set to 10^{-3} and the total number of training epochs is set to 150.

We selected 1000 images from the TT100K-rain for the ablation study. We first tested the different methods on the dataset with rain marks. Then, we used two-stage method with PMRNet to perform image deraining before traffic sign recognition.

Table 5. Results of ablation experiments on TT100K-rain datasets

Method	Parameters/M	AR	mAP	Speed/s
Yolov5	12.41	0.736	0.721	0.20
CoT-YOLOv5	12.46	0.793	0.787	0.025
PMR-Yolov5	14.01	0.831	0.871	0.022
PMR-CoT-Yolov5(our)	**14.06**	**0.893**	**0.921**	**0.027**

It can be seen from Table 5 that if the image does not go through PMRNet's deraining operation, the accuracy of the algorithm will drop significantly. This shows the importance of image deraining operation. Additionally, because the structure of PMRNet is not complicated, the speed reduction of the algorithm is still within an acceptable range. After adding the CoT module, the accuracy of the algorithm has obvious advantages in both rainy images and clean images.

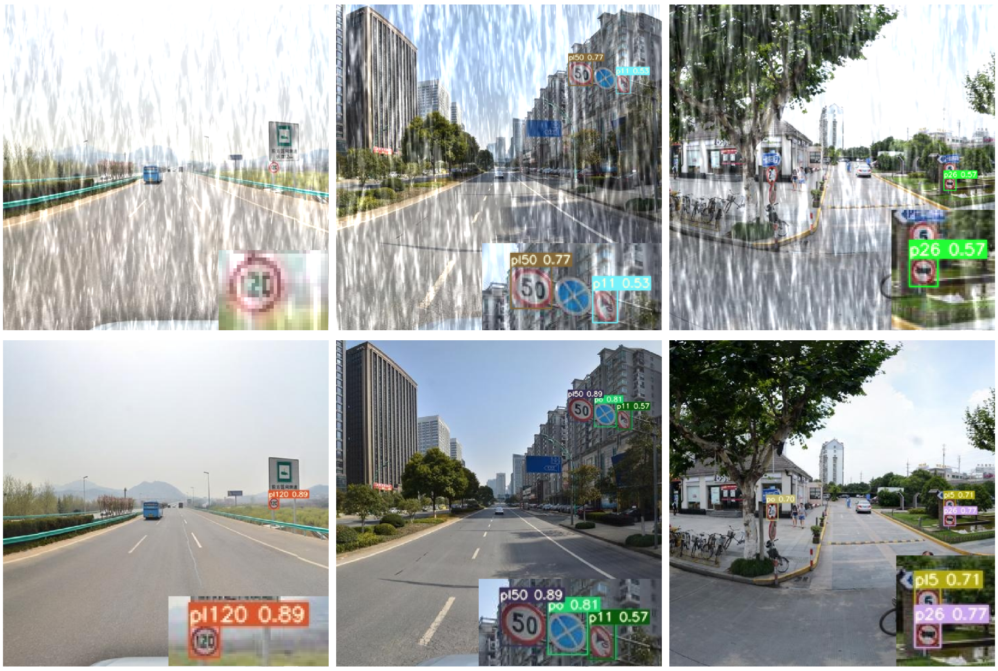


Figure 10. PMRNet-based recognition effect of CoT-YOLOv5.

Figure 10 shows the experimental results of our PMR-CoT-YOLOv5 method. The first column is the effect of not using PMRNet. It can be seen that due to the interference of rain marks, some traffic sign recognition has not been recognized. The second column is the recognition effect of CoT-YOLOv5 after the deraining operation. It can be seen that all traffic sign recognition is accurately recognized.

We evaluated our CoT-YOLOv5 on the TT100K-rain after deraining and compared it to the original YOLOv5. We use metrics including number of parameters, average recall AR, average precision mean mAP, and single image processing speed to evaluate the performance, and the results are shown in Table 6. From the results, the CoT-YOLOv5 improved mAP by 5% and AR by 6.2%, while the number of parameters and the recognition speed of a single image increased only slightly. Although the speed decreases by 0.005 s, it still meets our requirements for detection speed.

Table 6. Experimental results of TT100K-rain datasets.

Method	Parameters/M	AR	mAP	Speed/s
Faster R-CNN + ACFPN [22]	**137.59**	0.876	0.901	1.42
Mask R-CNN [23]	94.20	0.891	0.913	1.10
SSD [25]	29.49	0.813	0.837	**0.012**
SSD-RP [26]	30.69	0.841	0.858	0.019
YOLOv4 [28]	64.12	0.843	0.875	0.031
Yolov5 [29]	12.41	0.831	0.871	0.020
CoT-Yolov5(our)	12.46	**0.893**	**0.921**	0.025

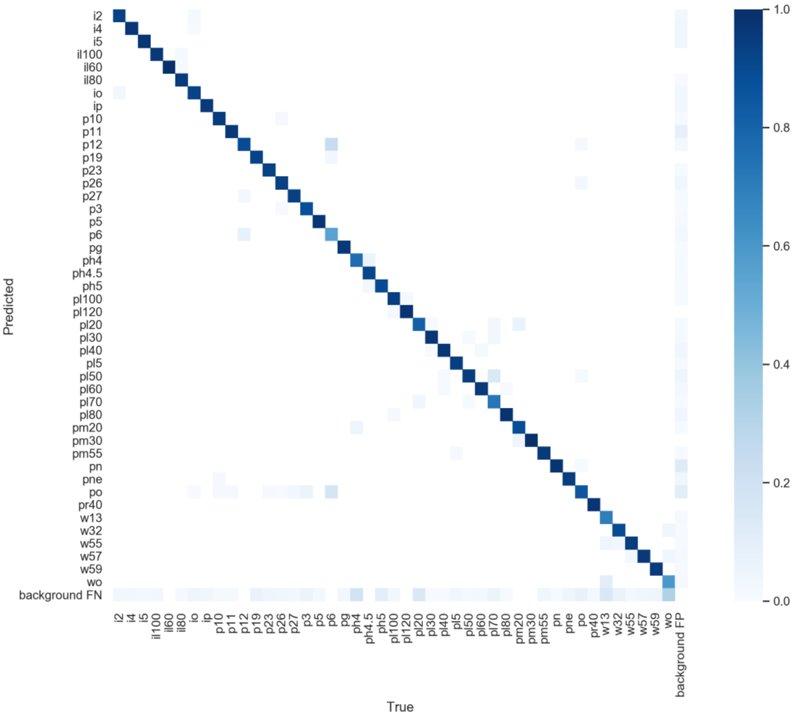
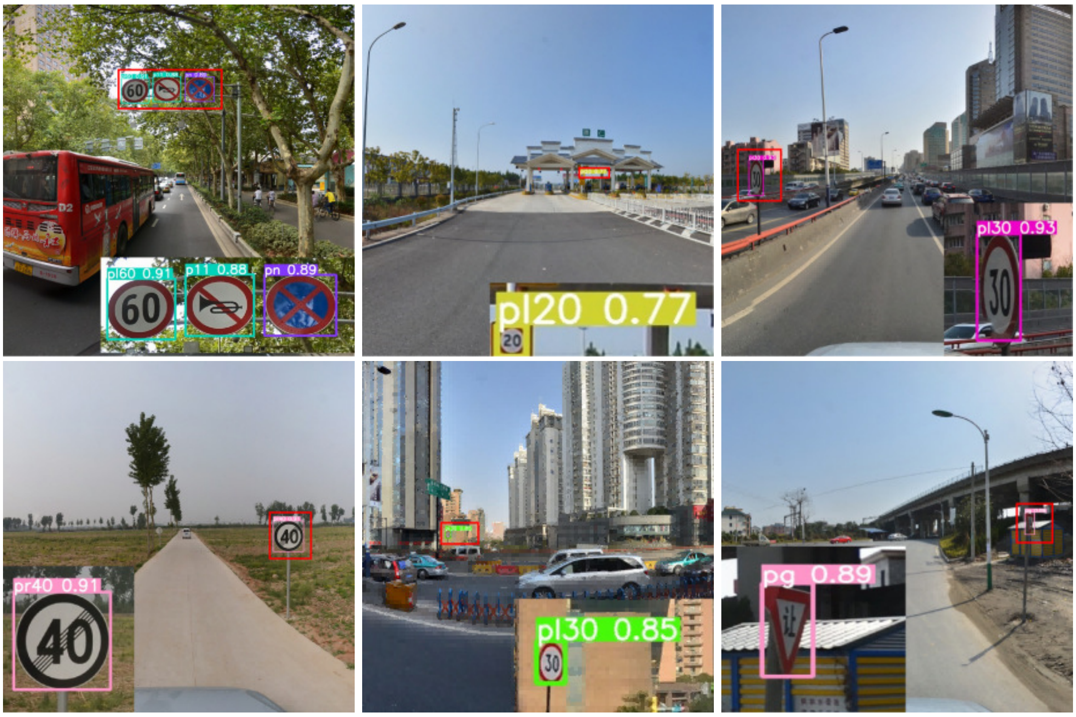


Figure 11. Confusion matrix of 45 types of traffic signs.

Figure 11 shows the confusion matrix obtained after testing the 45 categories of traffic signs we have trained, where the horizontal axis of the confusion matrix represents the manually true labeled categories and the vertical axis represents the predicted categories. The shade of the diagonal color in the figure represents the probability value of true positives for this category. The depth of the diagonal color in the figure represents the probability value of true positives of this category, that is, the probability of being correctly classified as a positive sample. It can be seen that except for individual categories that are lighter in color due to the less content in the data set, most categories have better performance values between 0.8 and 1.0.

The actual traffic sign recognition effect of our method is shown in Figure 12, where (a) is a large-sized traffic sign (b) is a small-sized traffic sign (c) is a sloping traffic sign. Our method can accurately identify the target in all three cases. This shows that CoT-YOLOv5 has good precision as well as robustness.



(a) (b) (c)

Figure 12. Traffic sign recognition effect of CoT-Yolov5.

5. Discussion

In the experiments of image deraining, our PMRNet algorithm outperforms other comparative algorithms in both metrics SSIM and PSNR on the Rain100H and Rain800 datasets. This shows that the multiscale residual module solves the problem of inadequate feature extraction from other networks to some extent. As can be seen from Table 3, the multi-stage network can progressively improve the deraining effect. This approach can effectively solve the residual problem of rain marks.

Therefore, our multi-scale residual module and multi-stage structure can effectively enhance the removal of rain marks. Then, we tested it based on the TT100K-rain dataset, and it can be seen from the results that the method can adapt to various backgrounds and rain marks, and has good generalization ability.

Then, we combined with the deraining algorithm to carry out the experiment of traffic sign recognition algorithm. As shown in Table 4, adding the CoT module YOLOv5 increases the accuracy by 6.6 and 5% in the rain image and the image after PMRNet preprocessing, respectively. From the results, we can see that the CoT module effectively enhances the feature extraction capability of YOLOv5. Compared with the direct recognition of rain images, the recognition accuracy of the CoT-YOLOv5 algorithm with PMRNet's pre-processing rises by 13.4%. This shows that our two-stage algorithm can effectively solve the problem of traffic sign recognition in rainy days. Then, we conducted a comparative experiment with the current representative algorithm on the image after deraining. Two-stage algorithms, such as Faster R-CNN + ACFPN, have higher accuracy, but are slower. Single-stage algorithms, such as the SSD series, have obvious advantages in speed, but recognition accuracy still needs to be improved. Our CoT-YOLOv5 algorithm has a clear advantage in accuracy. The embedding of PMRNet and CoT increases our training cost to some extent, but the structure of these two parts is not complicated. Therefore, the speed is only slightly decreased and still meets the real-time requirement.

6. Conclusions

Rainy weather will seriously affect the quality of the captured images, resulting in the degradation of the performance of the traffic sign recognition algorithm. In order to solve this problem, this paper divides the algorithm into two parts:image deraining and traffic sign recognition. In the first part, a deep learning-based progressive multiscale residual deraining network is proposed, which divides the network into multiple stages by recursion as a way to reduce the residual degree of rain marks, and uses multiscale residuals and ConvLSTM to enhance the representation of image features as a way to obtain better rain removal results. In the second part, this paper performs the recognition of traffic signs based on the recovered rain image, and in this part we use the improved YOLOv5 for the traffic sign recognition task. In order to improve the recognition accuracy of the network, we replace the 3 × 3 convolution in the C3 module with the CoT module, which solves the problem that the traditional CNN lacks the ability to model global information. The experimental results show that the method can effectively improve the recognition accuracy.

The method proposed in this paper is applicable to rainy weather, and since rainfall may occur simultaneously with other weather conditions (e.g., haze and snow), accumulating different situations for multi-task learning to improve performance is also worth exploring in future research. Of course, excluding the natural climate, traffic signs may also be obscured by buildings, trees and other objects that cannot be recovered. How to reasonably design deep learning network models to recognize traffic signs based on the characteristics of obscured traffic signs is also a problem that needs attention in the future. It is also worth exploring how to embed recognition algorithms into in-vehicle systems. Shi et al. [43] worked on automatic traffic sign recognition using video recorded by an in-vehicle vehicle recorder. With the development of smart cars, our algorithms can be deployed on embedded systems in some assisted driving systems and driverless systems that require low power consumption and high performance. Or algorithms could be deployed to run on a central processing unit to perform recognition on images captured from a front-facing camera. In the future, we will focus more on how to apply algorithms to various intelligent systems in a rational way.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant No.61902311, 51904226, 52274137, 62172330), in part by the Post doctoral Research Foundation of China (Grant No. 2019M663801), the Key R & D Foundation of Shaanxi Province (Grant No.2021SF- 479), and the Scientific Research Project of Shaanxi Provincial Education Department (No.22JK0459).

References

1. D. Chattaraj, B.Bera, A. Das, S. Saha, P. Lorenz, Y. Park, Block-CLAP: Blockchain-Assisted certificateless key agreement protocol for internet of vehicles in smart transportation, *IEEE Trans. Veh. Technol.*, **70** (2021), 8092–8107. https://doi.org/10.1109/TVT.2021.3091163
2. C. Chang, H. Lina, S. Huang, Traffic sign detection and recognition for driving assistance system, *Adv. Image Video Process.*, **6** (2018). https://doi.org/10.14738/aivp.63.4603
3. A. Madhu, V. S. Nair, Traffic sign detection and recognition for automated driverless cars based on SSD, *Int. J. Trend Sci. Res. Dev.*, **4** (2020).
4. C. Gerhardt, W. Broll, Neural network-based traffic sign recognition in 360° images for semi-automatic road maintenance inventory, in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, (2020). https://doi.org/10.1109/ITSC45102.2020.9294610
5. H. Li, D. Wang, J. Zhang, Z, Li, T. Ma, Image super-resolution reconstruction based on multi-scale dual-attention, *Connect. Sci.*, (2022). https://doi.org/10.1080/09540091.2023.2182487
6. H. Li, L. Hu, J. Zhang, Irregular mask image inpainting based on progressive generative adversarial networks, *Imaging Sci. J.*, (2023), 1–14. https://doi.org/10.1080/13682199.2023.2180834
7. J. Zhang, Q. Yan, X. Zhu, K. Yu, Using synthetic data for person tracking under adverse weather conditions, *Digital Commun. Networks*, **8** (2022), 1–86. https://doi.org/10.1016/j.dcan.2022.08.002
8. A. Kerim, U. Celikcan, E. Erdem, A. Erdem, Using synthetic data for person tracking under adverse weather conditions, *Image Vision Comput.*, **111** (2021), 104187. https://doi.org/10.1016/j.imavis.2021.104187
9. S. Huang, Q. Hoang, T. Le, SFA-Net: A selective features absorption network for object detection in rainy weather conditions, *IEEE Trans. Neural Networks Learn. Syst.*, (2022), 2162–2388. https://doi.org/10.1109/TNNLS.2021.3125679
10. S. Di, Q. Feng, C. Li, M. Zhang, H. Zhang, S. Elezovikj, et al., Rainy night scene understanding with near scene semantic adaptation, *IEEE Trans. Intell. Trans. Syst.*, **22** (2021), 1594–1602. https://doi.org/10.1109/TITS.2020.2972912

11. S. Kim, J. Lee, T. Yoon, Road surface conditions forecasting in rainy weather using artificial neural networks, *Safety Sci.*, **140** (2021), 0925–7535. https://doi.org/10.1016/j.ssci.2021.105302

12. R. R. Boukhriss, E. Fendri, M. Hammami, Moving object detection under different weather conditions using full-spectrum light sources, *Pattern Recognit. Lett.*, **129** (2020), 0925–7535. https://doi.org/10.1016/j.ssci.2021.105302

13. W. Yang, R. T. Tan, S. Wang, Y. Fang, J. Liu, Single image deraining: From model-based to data-driven and beyond, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 4059–4077. https://doi.org/10.1109/TPAMI.2020.2995190

14. L. J. Deng, T. Z. Huang, X. L. Zhao, T. X. Jiang, A directional global sparse model for single image rain removal, *Appl. Math. Model.*, **59** (2018), 662–679. https://doi.org/10.1016/j.apm.2018.03.001

15. H. Wang, Q. Xie, Q. Zhao, D. Meng, A model-driven deep neural network for single image rain removal, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, **59** (2020), 3103–3112. https://doi.org/10.1109/CVPR42600.2020.00317

16. X. Wang, Z. Li, H. Shan, Z. Tian, W. Zhou, FastDerainNet: A deep learning algorithm for single image deraining, *IEEE Access*, **8** (2020), 127622–127630. https://doi.org/10.1109/ACCESS.2020.3008324

17. X. Li, J. Wu, Z. Lin, L. Hong, H. Zha, Recurrent squeeze-and-excitation context aggregation net for single image deraining, in *Proceedings of the European conference on computer vision (ECCV)*, **11211** (2020), 262–277. https://doi.org/10.48550/arXiv.1807.05698

18. D. Ren, W. Zuo, Q. Hu, P. Zhu, D. Meng, Progressive image deraining networks: A better and simpler baseline, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), 3937–3946. https://doi.org/10.1109/CVPR.2019.00406

19. S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. H. Yang, et al., Multi-Stage progressive image restoration, in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, **129** (2021), 14821–14831. https://doi.org/10.1109/CVPR46437.2021.01458

20. L. Wang, X. Xu, R. Gui, R. Yang, F. Pu, Learning rotation domain deep mutual information using convolutional LSTM for unsupervised PolSAR image classification, *Remote Sens.*, **12** (2020). https://doi.org/10.3390/rs12244075

21. S. Luo, L. Yu, Z. Bi, Y. Li, Traffic sign detection and recognition for intelligent transportation systems: a survey, *J. Int. Technol.*, **21** (2021), 1773–1784. https://doi.org/10.3966/160792642020112106018

22. X. Li, Z. Xie, X. Deng, Y. Wu, Y. Pi, Traffic sign detection based on improved faster R-CNN for autonomous driving, *J. Supercomput.*, **78** (2022), 7982–8002. https://doi.org/10.1007/s11227-021-04230-4

23. D. Tabernik, D. Skočaj, Deep learning for large-scale traffic-sign detection and recognition, *IEEE Trans. Intell. Trans. Syst.*, **4** (2020), 1427–1440. https://doi.org/10.1016/j.patrec.2022.06.006

24. J. Du, Understanding of object detection based on CNN family and YOLO, *J. Phys. Conf. Ser.*, **1004** (2018), 012029. https://doi.org/10.1088/1742-6596/1004/1/012029

25. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, et al., Ssd: Single shot multibox detector, in *Computer Vision–ECCV 2016: 14th European Conference*, (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

26. J. Wu, S. Liao, Traffic sign detection based on SSD combined with receptive field module and path aggregation network, *Comput. Intell. Neurosci.*, **129** (2022), 1–13. https://doi.org/10.1155/2022/4285436

27. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, 2018, preprint, arXiv: 0707.0078.

28. A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, 2020, preprint, arXiv: 2004.10934.

29. D. Snegireva, A. Perkova, Traffic sign recognition application using Yolov5 architecture, in *2021 International Russian Automation Conference (RusAutoCon)*, (2021), 112–126. https://doi.org/10.1109/RusAutoCon52004.2021.9537355

30. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., An image is worth 16x16 words: Transformers for image recognition at scale, 2020, preprint, arXiv: 2010.11929.

31. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, *Adv. Neural Inform. Process. Syst.*, **30** (2017). https://doi.org/10.48550/arXiv.1706.03762

32. Y. Li, T. Yao, Y. Pan, T. Mei, Contextual transformer networks for visual recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, **45** (2022), 1489–1500. https://doi.org/10.1109/TPAMI.2022.3164083

33. K. Huang, C. Tian, J. Su, J. C. Lin, Transformer-based cross reference network for video salient object detection, *Pattern Recognit. Lett.*, **160** (2022), 122–127. https://doi.org/10.1016/j.patrec.2022.06.006

34. J. Zhou, J. Liu, J. Li, M. Huang, S. A. Nawaz, Mixed attention densely residual network for single image super-resolution, *Comput. Syst. Sci. Eng.*, **39** (2021), 133–146. https://doi.org/10.32604/csse.2021.016633

35. S. Bande, V. Bhatia, S. Prakash, MSE-based analysis of circular grating self-images for testing beam collimation, *Appl. Opt.*, **59** (2020), 7160–7168. https://doi.org/10.1364/AO.395348

36. H. Rezatofighi, N. Tsoi, J. Y. Gwak, A. Sadeghian, S. Savarese, Generalized intersection over union: A metric and a loss for bounding box regression, in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2019), 658–666. https://doi.org/10.1109/CVPR.2019.00075

37. W. Ma, T. Zhou, J. Qin, Q. Zhou, Z. Cai, Joint-attention feature fusion network and dual-adaptive NMS for object detection, *Knowl. Based Syst.*, **241** (2019). https://doi.org/10.1016/j.knosys.2022.108213

38. W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, S. Yan, Deep joint rain detection and removal from a single image, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, (2017), 1357–1366. https://doi.org/10.1109/CVPR.2017.183

39. H. Zhang, V. Sindagi, V. M. Patel, Image de-raining using a conditional generative adversarial network, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2020), 3943–3956. https://doi.org/10.1109/TCSVT.2019.2920407

40. C. Sun, M. Wen, K. Zhang, P. Meng, R. Cui, Traffic sign detection algorithm based on feature expression enhancement, *Multimedia Tools Appl.*, **80** (2021), 33593–33614. https://doi.org/10.1007/s11042-021-11413-x

41. J. Yan, S. Chen, Y. Zhang, X. Li, Neural architecture search for compressed sensing magnetic resonance image reconstruction, *Comput. Med. Imaging Graphics*, **85** (2020), 101784. https://doi.org/10.1016/j.compmedimag.2020.101784

42. M. Malarvel, G. Sethumadhavan, P. C. R. Bhagi, S. Kar, T. Saravanan, A. Krishnan, Anisotropic diffusion based denoising on X-radiography images to detect weld defects, *Digital Signal Process.*, **68** (2017), 112–126. https://doi.org/10.1016/j.dsp.2017.05.014

43. J. H. Shi, H. Y. Lin, A vision system for traffic sign detection and recognition, in *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, (2017), 1596–1601. https://doi.org/10.1109/ISIE.2017.8001485