PAPER

Research on the Algorithm of License Plate Recognition Based on MPGAN Haze Weather*

Weiguo ZHANG†, Jiaqi LU†a), Jing ZHANG†, Xuewen LI†, *Nonmembers*, *and* Qi ZHAO†, *Student Member*

SUMMARY The haze situation will seriously affect the quality of license plate recognition and reduce the performance of the visual processing algorithm. In order to improve the quality of haze pictures, a license plate recognition algorithm based on haze weather is proposed in this paper. The algorithm in this paper mainly consists of two parts: The first part is MPGAN image dehazing, which uses a generative adversarial network to dehaze the image, and combines multi-scale convolution and perceptual loss. Multi-scale convolution is conducive to better feature extraction. The perceptual loss makes up for the shortcoming that the mean square error (MSE) is greatly affected by outliers; the second part is to recognize the license plate, first we use YOLOv3 to locate the license plate, the STN network corrects the license plate, and finally enters the improved LPRNet network to get license plate information. Experimental results show that the dehazing model proposed in this paper achieves good results, and the evaluation indicators PSNR and SSIM are better than other representative algorithms. After comparing the license plate recognition algorithm with the LPRNet algorithm, the average accuracy rate can reach 93.9%.

key words: *image dehazing, generative adversarial network, multi-scale convolution, perceptual loss, license plate recognition*

1. Introduction

The haze weather causes the image to become gray and the image quality is reduced, which cannot meet the needs of the society. Therefore, the image dehazing technology is very popular now, and this technology is used in various fields, such as: video surveillance [1], medical field [2] and unmanned driving [3]. The development of deep learning has enabled better application of image dehazing technology. Especially in the license plate recognition of traffic management [4], the license plate is the easiest sign to distinguish vehicles. The accuracy of the license plate recognition directly determines the quality and efficiency of traffic management. Deep learning image dehazing technology better solves the problem of unclear images in haze days.

The current image dehazing methods mainly include dehazing algorithms based on image enhancement [5], dehazing algorithms based on physical models [6], and dehazing algorithms based on deep learning [7]. The methods based on image enhancement mainly rely on enhancing the contrast to perform image dehazing. These methods have a large amount of calculation, a large running space requirement, and easily cause image distortion. The method based on the physical model is actually based on the atmospheric scattering model, using prior knowledge to estimate the atmospheric illumination value and the transmission matrix. He [8] proposed a dark channel prior algorithm (DCP), which can better estimate the transmittance map. For the deep learning method, Cai [9] proposed the DehazeNet network, which better estimated the transmission map, but did not consider the influence of the atmospheric light value on the model in detail. Li [10] proposed the AOD-Net dehazing model, which reduces the error through formula conversion, but still needs to be estimated, and the error is inevitable. P.L. Suárez [11] uses GAN to eliminate haze and proposes a multi-loss function scheme applied to conditional probability models, which increases the convergence speed and generalization ability of the model. Y. Dong [12] proposed the FD-GAN dehazing model, adding frequency information to the model to make the color distortion of the picture after dehazing less distorted. In terms of training models, these two methods use GAN for end-to-end training, but these two methods have not improved in feature extraction, and the model obtains less feature information, so the effect of dehazing is improved but not ideal.

In order to solve the above problems, a generative adversarial network model based on multi-scale convolution is proposed in this paper. The generative adversarial network [13] is used for image dehazing without the need to estimate the value in the atmospheric scattering model, which adds multi-scale convolution and perceptual loss [14]. Multi-scale convolution is used to extract features of the input haze image, which can extract deeper features, and the perceptual loss compensates for the shortcoming that the mean square error is greatly affected by outliers, so as to improve the effect of dehazing. The implementation results show that the method in this paper has a good improvement in the effect of dehazing.

License plate recognition is mainly the recognition of characters. It is the first time that convolutional neural networks are applied to digital recognition in the LeNet network model proposed by Y. Lecun [15]. The most common one is now the LeNet-5 model [16], which is a reference to LeNet. The simplification reduces the amount of calculation. The BP neural network proposed by Qin [17] realized the non-linear mapping function from input to output. But

Manuscript received August 23, 2021.
Manuscript revised December 19, 2021.
Manuscript publicized February 21, 2022.

†The authors are with Xi'an University of Science and Technology, China.

*This work was supported in part by the National Natural Science Foundation of China (Grant No. 61902311), the Postdoctoral Research Foundation of China (Grant No. 2019M663801), Key R & D Foundation of Shaanxi Province (Grant No. 2021SF-479) and Xi'an University of Science and Technology Summit Project.

a) E-mail: Lu_13038018813@163.com

DOI: 10.1587/transinf.2021EDP7178

the first two methods use character segmentation for training. This will increase training costs, and is time-consuming and prone to gradient problems. LPRNet [18] does not need to segment characters, saves time, and effectively solves the gradient problem, but the recognition accuracy of complex situations is not ideal, the training time is still long, and it is prone to crash during training. This paper improves the activation function and the maximum pooling step size on the basis of LPRNet, changing the activation function Relu to LeakyRelu to solve the phenomenon of training collapse; changing the maximum pooling step size to 2, which further saves training time. The improved LPRNet network improves the recognition effect in complex situations and shortens the training time.

The contributions of this paper are listed as follows:

(a) For the problem of license plate recognition in haze days, this paper proposes a dehazing model based on MPGAN, which combines multi-scale convolution and perceptual loss on the basis of generating a confrontation network. This model is an end-to-end model and does not need to estimate the atmospheric illumination value and transmittance map. The model uses multi-scale convolution to learn deeper features, and adds perceptual loss to compensate for the large shortcomings of mean square error affected by outliers, so as to improve the effect of haze removal.

(b) This paper uses the dehazed image for license plate recognition and an improved LPRNet network model. On the one hand, the model changes the activation function to LeakyRelu, which solves the shortcomings of Relu crashing during training; on the other hand, the maximum pooling step size is changed to 2, which reduces the amount of calculation, saves time, and improves accuracy.

2. Our Approach

2.1 MPGAN Dehazing Algorithm

The method of this paper is to let the MPGAN model directly learn the non-linear mapping from haze image to dehaze image, without relying on the atmospheric scattering model [19]. The learning of this non-linear mapping is analogy between dehazing and denoising [20], Which is derived from the haze as a special kind of noise. The atmospheric scattering model is transformed into formula (1).

$$I(x) = J(x) + (A - J(x))(1 - t(x)) \tag{1}$$

In formula (1), I(x) is the image to be dehazed, J(x) is the clear image, A is the atmospheric illumination value, and t(x) is the transmission matrix. According to formula (2), formula (1) can be transformed into formula (3). Among them u(x) is an error term of nonlinearity related to blur effect.

$$u(x) = (A - J(x))(1 - t(x)) \tag{2}$$

$$I(x) = J(x) + u(x) \tag{3}$$

Through such a variant, dehazing can be regarded as a kind of non-linear noise, then the formula (3) can be written as:

$$I(x) = J(x) + w(x) \tag{4}$$

Among them, I(x) is a noisy image, and J(x) is a clear image. If the w(x) distribution belongs to N(0,σ_W^2), it is called Gaussian white noise. Although formula (3) and formula (4) are similar in structure, they represent different meanings. u(x) does not include processing additive Gaussian white noise.

The structure diagram of this network is shown as in Fig. 1. The haze image is input to the network, and go through the convolutional layer with the convolution kernel size of 3x3 and 7x7 respectively; enter the residual block containing the convolutional layer, the Batch Normalization (BN) layer, and the Elementwise Subtract layer [21]; the BN layer is solved in training In the process, the data distribution of the middle layer changes to prevent the gradient from disappearing or exploding, and to speed up the training speed. The Elementwise Subtract layer in the residual block subtracts the residual of the current sub-block from the output of the nearest sub-block. The Elementwise Subtract layer outside the residual block performs a subtraction of input and output. Finally, through the Elementwise add layer, two different features are merged, and finally a clear

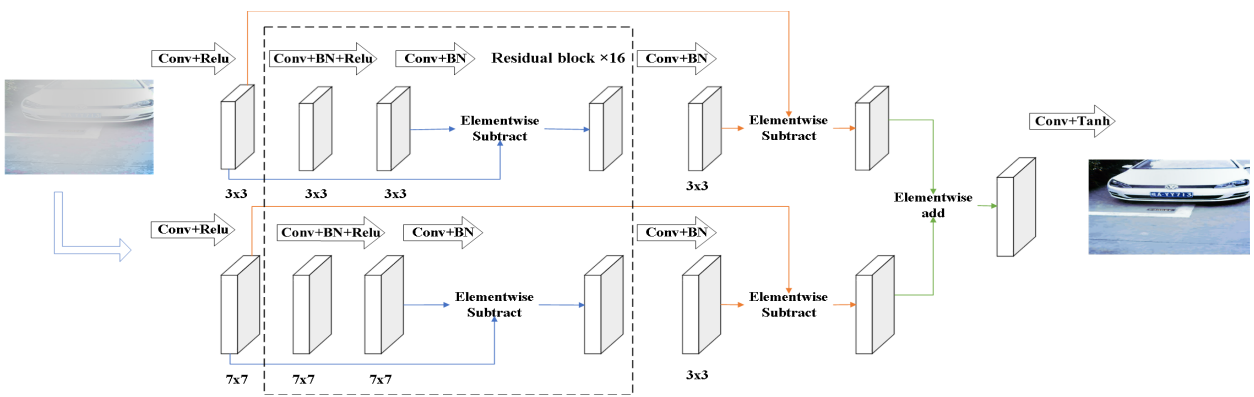


Fig. 1 MPGAN model structure

image is obtained through a convolutional layer with an activation function of Tanh.

2.1.1 Multi-Scale Convolution

A general convolutional neural network consists of a convolutional layer, a pooling layer and a fully connected layer. The convolutional neural network extracts features through the convolutional layer and the pooling layer to obtain a feature map. However, the general convolutional neural network cannot fully extract the features of the input image, so this paper uses multi-scale convolutional neural network instead of ordinary convolutional neural network to improve the feature extraction of the input image.

The multi-scale convolutional neural network is to down-sample the input image several times, perform feature extraction on the results of these down-sampling, extract image features of different scales, and then fuse these features to finally obtain a multi-scale feature map.

Feature maps of different scales obtained by down-sampling need to be feature fused, and then passed into the next layer of convolution. Feature fusion [22] is to generate new features from the extracted different features in a certain way, so that the new features are paired with The subsequent operations have a better effect.

There are many different ways to achieve multi-scale convolution [23], such as using different sizes of convolution kernels or using pooling layers. This paper does not use a pooling layer for multi-scale implementation. Although the pooling layer can increase the receptive field and allow the convolution to see more information, the pooling will definitely cause the loss of some features, such as average pooling (average pooling). pooling) will blur important features, so this experiment does not use a pooling layer, and uses different sizes of convolution kernels to achieve multi-scale convolution.

As shown in Fig. 1 of the dehazing model structure, convolution is performed through two different lines, 3x3 and 7x7 convolution kernels, to achieve multi-scale convolution operations, and the size of the feature map obtained by each convolution is as follows As shown in formula (5).

$$Output\_s = \frac{(Input\_s - Filter\_s)}{Stride} + 1 \tag{5}$$

Among them, Output_s is the size of the output feature map, Input_s is the size of the input image, Filter_s is the size of the convolution kernel, and Stride is the step size.

2.1.2 Loss Function

The loss functions used in training the network in this paper are mean square error (MSE) [24], VGG19 [25] and perceptual loss. The formula of the loss function can be expressed as formula (6).

$$L = \omega_1 L_{MSE} + \omega_2 L_{VGG} + \omega_3 L_{PER} \tag{6}$$

Among them, ω_i (i=1,2,3) is the weight to control each loss, L_{MSE} stands for mean square error, L_{VGG} stands for VGG loss, and L_{per} stands for perceptual loss.

MSE refers to the expected value of the square of the difference between the estimated value of the parameter and the true value of the parameter. The smaller the value of MSE, the better the accuracy of the experimental data described by the prediction model. The formula (7) is as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{m} \omega_i (y_i - \hat{y}_i)^2 \tag{7}$$

Among them, y_i is the real data, $\hat{y}_i$ is the fitted data, $\omega_i > 0$, and n is the number of samples.

The MSE function curve is continuous and derivable everywhere, which is convenient for using the gradient descent method, and as the error decreases, the gradient is also reduced, which is conducive to convergence, even if a fixed learning rate is used, it can quickly converge to the minimum. However, MSE suffers from uncertainties in the data set, there may be various equally possible results for the same input. It is more sensitive to outliers and is greatly affected by them. Therefore, this paper introduces perceptual loss to improve the effect of MSE. Perceptual loss is to constrain the original image and the generated image at the depth feature level. Perceived loss includes content loss and style loss, namely formula (8), formula (9) and formula (10):

$$L_{feat}^{\phi,j}(x, x') = \sum_{i=1}^{N} \frac{1}{C_j H_j W_j} [\|\phi_j(x) - \phi_j(x')\|_2^2] \tag{8}$$

$$G_j^{\phi}(x)_{c,c'} = \frac{1}{C_j H_j W_j} \sum_{h=1}^{H_j} \sum_{w=1}^{W_j} \phi_j(x)_{h,w,c} \phi_j(x)_{h,w,c'} \tag{9}$$

$$L_{style}^{\phi,j}(x, x') = \|G_j^{\phi}(x) - G_j^{\phi}(x')\|_2^F \tag{10}$$

In formula (8) (9) (10), ϕ is the pre-trained VGG loss network, and j is the j-th layer of the network. x and x'represent the real picture and the generated picture, respectively. C represents the number of channels, H and W represent the height and width of the picture respectively. c and c' indicate different channels. The formula (8) is the content loss, that is, the Euclidean distance between different features. The formula (9) is the inner product of different channel characteristics. The formula (10) is the style loss, which is the Euclidean distance between different layers. x, x' respectively represent the real image and the generated image, ϕ represents the trained neural network, and j represents the jth layer of the network.

2.2 License Plate Recognition

In terms of traffic management, due to the influence of haze, the images taken by the camera are not clear and the license plate information cannot be accurately recognized. Therefore, this paper recognizes the license plate based on the dehazed image. The specific steps of license plate recognition

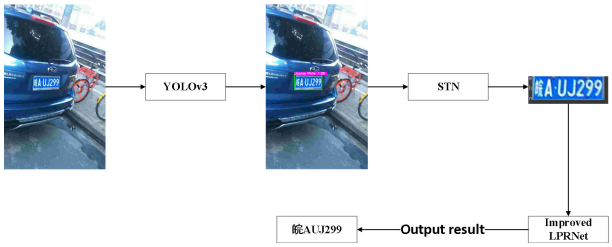


Fig. 2 License plate recognition experiment process

are as follows:

Step 1: Use YOLOv3 network to locate the license plate in the input image.

Step 2: Because the license plate needs to be corrected when the license plate is tilted, the STN network is used to correct the license plate in the image.

Step 3: The improved LPRNet network is used to extract the features of the images corrected by the STN network, and finally the result is obtained.

The experiment flow chart is shown in Fig. 2.

2.2.1 YOLOv3

YOLO [26] is currently one of the most popular target detection algorithms. This paper uses YOLOv3 [27] for license plate location. The backbone network of YOLOv3 is Darknet53. The network is fully convolutional and has no pooling layer. Downsampling is performed by adjusting the step size of the convolution. The jump connection of the residual network is used extensively, and in order to reduce the gradient caused by pooling As a negative effect, YOLOv3 directly abandoned POOLing and used the stride of convolution conv to achieve downsampling. The basic component of YOLOv3 is the DBL layer, that is, convolution + BN (batch normalization) + activation function Leaky relu, and the BN layer is Batch Normalization [28] to speed up the convergence of the neural network. Yolov3 draws on the output of FPN [29] and has 3 feature maps of different scales. By using multiple scales to detect targets of different sizes, very fine objects can be obtained.

2.2.2 Spatial Transformer Networks

Spatial Transformer Networks (STN) [30] is the operation of correcting slanted license plates. When collecting license plate data, there will be some unavoidable situations, such as tilt, flip, etc. If these images are directly input into the neural network without processing, it will have a negative impact on the training, so it needs to be input before the neural network, we use the spatial transformation network to preprocess the data spatially. The structure of the STN is shown in Fig. 3.

It mainly includes three parts: Localisation Network, Grid Genator and Sampler. The local network is a network of regression transformation parameters. It takes the feature map as input, passes through a series of hidden layers, and

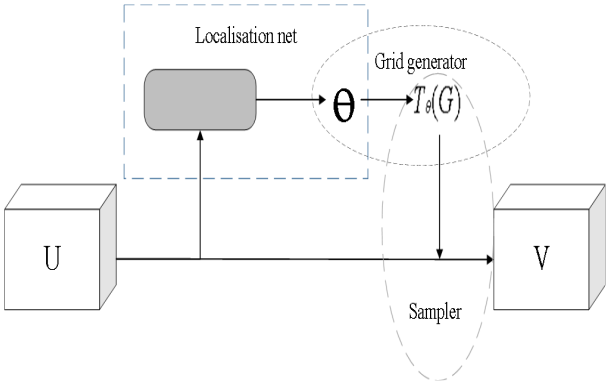


Fig. 3 STN structure

outputs transformation parameter θ through formula (11), namely:

$$\theta = f_{loc}(U) \tag{11}$$

The grid generator constructs a sampling network based on the transformation parameters output from the local network. The grid generator obtains a mapping relationship $T_\theta(G)$. If the coordinate of each pixel of the feature map U in the figure is (x_i^u, y_i^u),the feature map The coordinate of each pixel of V is (x_i^v, y_i^v),and the corresponding relationship between them is shown in formula (12) and formula (13), where u and v represent the feature map U and the feature map V, respectively.

$$\begin{pmatrix} x_i^u \\ y_i^u \end{pmatrix} = T_\theta(G_i) \tag{12}$$

$$T_\theta(G_i) = A_\theta \begin{pmatrix} x_i^v \\ y_i^v \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^v \\ y_i^v \\ 1 \end{pmatrix} \tag{13}$$

The sampler uses the sampling network and the input feature map as input, and obtains the transformed result of the feature map through formula (14).

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c max(0, |x_i^v - m|) max(0, |y_i^v - n|) \tag{14}$$

Generally, the neural network model for license plate recognition needs to be trained with character segmentation during training. Such a method will increase time-consuming and increase training costs. The method used in this paper does not require character segmentation to train the model during training, it is an end-to-end training method. The loss function used in this paper is CTCLoss [31]. The training process is similar to the traditional neural network. The difference is that the traditional neural network has the smallest training error for each frame of data, while the CTC training is in sequence, such as maximizing p(y|x), an input sequence can correspond to multiple paths. CTCLoss calculates the loss between the continuous time series and the target series. CTCLoss sums the probability that the input may be aligned with the target, and

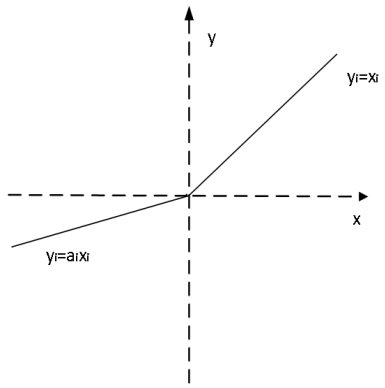


Fig. 4 LeakyRelu function

Table 1 LPRNet structure

Larger Type	Parameters
Input	RGB License Plate Image
Convolution	64 3x3 stride 1
MaxPooling	64 3x3 stride 2
Small Basic Block	128 3x3 stride 1
MaxPooling	64 3x3 stride 2
Small Basic Block	256 3x3 stride 1
Small Basic BlockX	256 3x3 stride 1
MaxPooling	64 3x3 stride 2
Dropout	0.5 ratio
Convolution	256 4x1 stride 1
Dropout	0.5 ratio
Convolution	class_num 1x13 stride 1

produces a loss value that is differentiable with respect to each input node. The alignment of the input and the target is assumed to be "many-to-one", which limits the length of the target sequence, making it necessary to input the length.

2.2.3 Improved LPRNet Network

The LPRNet network uses a lightweight neural network as the skeleton network, and adds the Batch Normalization layer and the Relu activation layer after each step of the convolution operation. In this experiment, the activation function is changed to LeakyRelu [32], which makes up for when the input value of Relu is negative, the output is always 0, and its first derivative is always 0, which leads to the defect that the neuron cannot update the parameters, and solves the shortcoming of Relu during training. The problem of crash. It is obvious from the comparison of formula (15) and formula (16) that leak assigns a non-zero slope to all negative values, and the LeakyReu function is shown in Fig. 4.

$$Relu : y = max(0, x) \tag{15}$$

$$LeakyRelu : y = max(0, x) + leak * min(0, x) \tag{16}$$

The improved LPRNet changes the convolution step size of the maximum pooling to 2, which reduces the amount of calculation and saves time. The structure of the improved LPRNet is shown in Table 1, and the structure of the Small Basic Block module is shown in Table 2.

Table 2 Small basic block structure

Larger Type	Parameters/Dimensions
Input	C_{in}×H×W feature map
Convolution	C_{out}/4 1x1 stride 1
Convolution	C_{out}/4 3x1 strideh=1, padh=1
Convolution	C_{out}/4 1x3 stridew=1, padw=1
Convolution	C_{out} 1x1 stride 1
Output	C_{out}×H×W feature map

3. Experiments

The experiment in this paper is divided into two parts: the dehazing part uses the O-HAZY dataset (https://data.vision.ee.ethz.ch/cvl/ntire18/o-haze/) and the synthetic hazy license plate dataset. The O-HAZY dataset trains the model and synthesizes the hazy license plate dataset for testing. The license plate recognition part uses the CCPD2019 data set (https://github.com/detectRecog/CCPD). The O-HAZY data set is an outdoor scene data set, composed of pairs of hazy and corresponding fog-free images, including 45 types Different outdoor scenes. A total of 100 hazy license plates were synthesized. This experiment uses the CCPD2019 data set. The CCPD2019 data set is mainly collected in the parking lot of Hefei. The license plate photos taken involve a variety of complex environments, including blur, tilt, rain, snow, etc. This experiment uses the CCPD-Blur data set in the CCPD2019 data set, and takes a total of 10,000 images, of which 8,000 are used for training the model and 2,000 are used for testing. Part of the data set is shown in Fig. 5. The CPU used in this experiment is Intel(R) Core(TM) i9-9820X CPU @ 3.30GHz, and GPU is NVDIA GeForce RTX 2080 Ti.

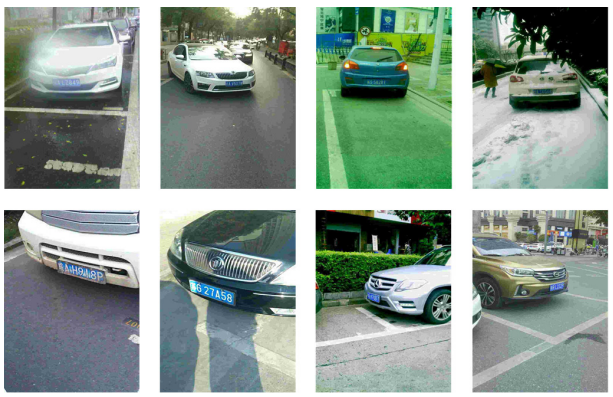
3.1 Eperiment of MPGAN Dehazing Algorithm

In this paper, PSNR (Peak Signal-to-Noise Ratio) [33] and SSIM (Structural Similarity) [34] are used as the evaluation indicators of the dehazing effect. The expression of the peak signal-to-noise ratio is as in formula (17), which is the logarithm of the mean square error between the original image and the processed image relative to $(2n-1)_2$, n is the number of bits per sample. The unit of peak signal-to-noise ratio is dB. SSIM is an index to measure the similarity of two images, usually expressed as formula (18), u_X and u_Y represent the mean values of images X and Y, σ_X and σ_Y represent the standard deviations of images X and Y, and σ_{XY} represents the covariance of image X and Y. SSIM adds C_1 and C_2 to the denominator to avoid the situation where the denominator is zero. This method uses the O-HAZY data set for training. The effect comparison on the O-Haze data set is shown in Fig. 6 and Table 3. The dehazing effect on the synthetic hazy license plate data set is shown in Fig. 7 and Table 4.

$$PSNR = 10log_{10}(\frac{(2^n - 1)^2}{MSE}) \tag{17}$$

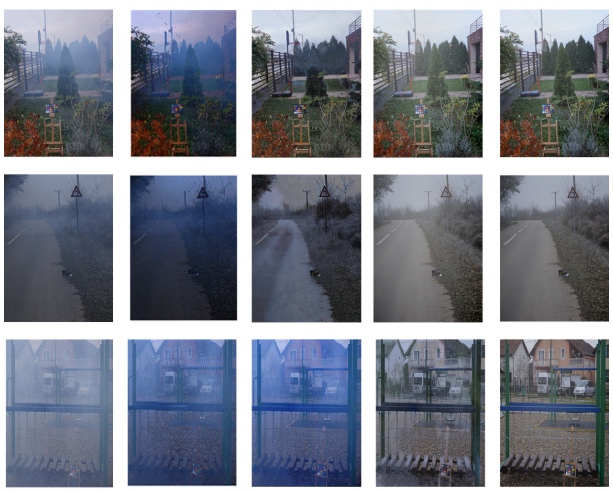
$$SSIM(X, Y) = \frac{(2u_X u_Y + C_1)(2\sigma_{XY} + C_2)}{(u_X^2 + u_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{18}$$

(a) O-HAZY data set



(b) CCPD data set

Fig. 5 Experimental data set



(a) input (b) He's algorithm[8] (c) Cai's algorithm[9] (d) Li's algorithm[10] (e) ours

Fig. 6 Comparison results of the dehazing of the O-HAZY dataset

Table 3 Experimental results of the O-Haze dataset

Indicator	PSNR	SSIM
He's algorithm [8]	18.98	0.843
Cai's algorithm [9]	19.25	0.865
Li's algorithm [10]	20.64	0.873
Our approach	**21.82**	**0.895**

(a) input (b) He's algorithm[8] (c) Cai's algorithm[9] (d) Li's algorithm[10] (e) ours

Fig. 7 Comparison results of the dehazing of the license plate images

Table 4 Experimental results of synthesizing the hazy license plate dataset

Indicator	PSNR	SSIM
He's algorithm [8]	18.67	0.821
Cai's algorithm [9]	19.18	0.849
Li's algorithm [10]	20.48	0.866
Our approach	**21.78**	**0.892**

In the above dehazing effect comparison diagram, (a) is the image to be dehazed, (b) is the dehazing effect of He's algorithm, (c) is the dehazing effect of Cai's algorithm, (d) is the dehazing effect of Li's algorithm, (e) is the experimental result of this paper. From the comparison of Fig. 6 and Fig. 7, it can be clearly seen that the dehazing effect of MPGAN is significantly better than that of the traditional neural network.

It can be seen from Table 3 and Table 4 that the method proposed in this paper is superior to the other three algorithms in terms of peak signal-to-noise ratio (PSNR) and structural similarity (SSIM).

With the development of image processing technology in deep learning, this section proposes the MPGAN dehazing model, which is a generative adversarial network dehazing model, combined with multi-scale convolution and perceptual loss, and is a complete end-to-end dehazing. The model does not need to rely on the atmospheric scattering model and achieves a better effect in dehazing.

3.2 Eperiment of License Plate Recognition Algorithm

This section is to identify the license plate images after dehazing in Sect. 3.1. First, this paper uses the YOLOv3 method to locate the dehazing license plate, as shown in Fig. 8. Secondly, in the process of license plate positioning, the license plate will be tilted, which is not conducive to the license plate recognition work as shown in (a) (c) in Fig. 8. This experiment uses the STN space transformation network to correct this tilt and intercept the license plate, the

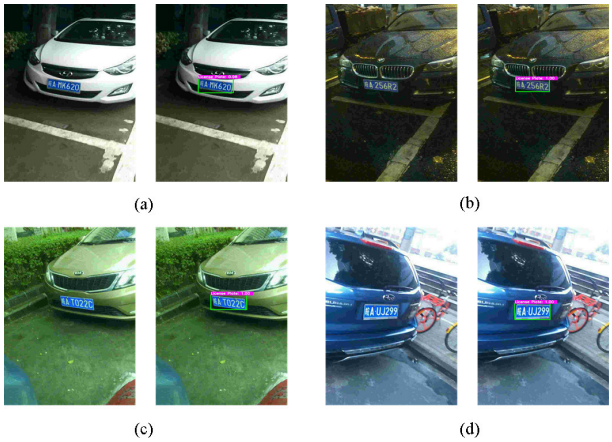


Fig. 8 YOLOv3 license plate location map

Fig. 9 STN license plate correction image

Table 5 Recognition accuracy rate with or without STN network

Model	Accuracy
LPRNet without STN network	91.8%
LPRNet with STN network	93.3%

Table 6 Comparison table of accuracy rate of license plate recognition

Method	LPRNet	Improved LPRNet
Average accuracy	93.3%	93.9%
Average time	0.403s	0.317s

corrected result is shown in Fig. 9. The impact of using STN network on accuracy is shown in Table 5. It can be seen that the accuracy of LPRNet with STN network is 1.5% higher than that of LPRNet without STN network.

Finally, the license plate corrected by the STN network is input into the improved LPRNet network to get the result. The experiment flow chart is shown in 9.

The recognition accuracy of the improved LPRNet is higher than before. Compared with the LPRNet before the improvement, the average accuracy is improved by 0.6%, and the average time for each recognition is shortened by 0.086 seconds. The experimental results are shown in Table 6.

4. Conclusion

The haze weather has seriously affected the quality of the images, making it impossible to perform license plate recognition. This paper is mainly divided into two parts, namely dehazing and license plate recognition. The first part proposes the MPGAN dehazing model, which is image dehazing algorithm based on generative adversarial network, combing with multi-scale convolution and perceptual loss, which can better remove the haze. Then we combine this method with actual traffic work for license plate recognition in foggy or haze conditions. The second part is to recognize the license plate based on the dehazing. First, we use YOLOv3 to locate the license plate, then use the STN network to spatially correct the license plate. Finally, we input the license plate into the improved LPRNet network to get the result. The experiment proves that the method proposed in this paper has improved the recognition accuracy. The MPGAN proposed in this paper has a good dehazing effect in most cases, but the dehazing effect of hazy images that are too blurry or distorted needs to be improved. The license plate recognition network in this paper can recognize the license plate after dehazing, and achieves a good effect, but the situation that the license plate is blocked or faded can be further improved.

Deep learning image dehazing technology has become a trend. Relying on independent learning, it solves the difficulty of manual operation and the problem of low image quality in hazy conditions, and provides help for license plate recognition, even in hazy conditions. It can also accurately identify the license plate information. Therefore, how to design and train neural networks more efficiently, so that the model can analyze data more accurately, and then better apply to license plate recognition is a problem that needs to be studied in the future.

References

[1] H. Zhu, X. Yin, and X. Wang, "Improved image defogging method and its application in video monitoring system based on ARM," Sensor Review, vol.34, no.3, pp.266–272, 2014. DOI:10.1108/SR-05-2012-684

[2] B. Savelli, A. Bria, A. Galdran, C. Marrocco, M. Molinara, A. Campilho, and F. Tortorella, "Illumination Correction by Dehazing for Retinal Vessel Segmentation," 2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS), pp.219–224, 2017. DOI:10.1109/CBMS.2017.28

[3] T. Gao, K. Li, T. Chen, M. Liu, S. Mei, K. Xing, and Y.H. Li, "A Novel UAV Sensing Image Defogging Method," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol.13, pp.2610–2625, 2020. DOI:10.1109/JSTARS.2020.2998517

[4] Y. He and Z. Liu, "A Feature Fusion Method to Improve the Driving Obstacle Detection under Foggy Weather," IEEE Transactions on Transportation Electrification, vol.7, no.4, pp.2505–2515, 2021. DOI:10.1109/TTE.2021.3080690

[5] H. Zhang, X. Liu, Z. Huang, and Y. Ji, "Single image dehazing based on fast wavelet transform with weighted image fusion," 2014 IEEE International Conference on Image Processing (ICIP), pp.4542–4546, 2014. DOI:10.1109/ICIP.2014.7025921

[6] K.B. Gibson and T.Q. Nguyen, "On the effectiveness of the Dark Channel Prior for single image dehazing by approximating with minimum volume ellipsoids," 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp.1253–1256, 2011. DOI:10.1109/ICASSP.2011.5946638
[7] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single Image Dehazing via Multi-scale Convolutional Neural Networks," European Conference on Computer Vision, vol.9906, pp.154–169, 2016. DOI:10.1007/978-3-319-46475-6_10
[8] K. He, J. Sun, and X. Tang, "Single Image Haze Removal Using Dark Channel Prior," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.33, no.12, pp.2341–2353, 2011. DOI: 10.1109/TPAMI.2010.168
[9] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao, "DehazeNet: An End-to-End System for Single Image Haze Removal," IEEE Transactions on Image Processing, vol.25, no.11, pp.5187–5198, 2016. DOI: 10.1109/TIP.2016.2598681
[10] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, "AOD-Net: All-in-One Dehazing Network," 2017 IEEE International Conference on Computer Vision (ICCV), pp.4780–4788, 2017. DOI:10.1109/ICCV.2017.511
[11] P.L. Suárez, A.D. Sappa, and B.X. Vintimilla, "Deep Learning Based Single Image Dehazing," IEEE/CVF Conference on Computer Vision & Pattern Recognition Workshops, 2018. DOI: 10.1109/CVPRW.2018.00162
[12] Y. Dong, Y. Liu, H. Zhang, S. Chen, and Y. Qiao, "FD-GAN: Generative Adversarial Networks with Fusion-Discriminator for Single Image Dehazing," Proceedings of the AAAI Conference on Artificial Intelligence, vol.34, no.7, pp.10729–10736, 2020. DOI: 10.1609/aaai.v34i07.6701
[13] L. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Networks," Association for Computing Machinery, vol.63, no.11, pp.139–144, 2020. DOI:10.1145/3422622
[14] Q. Wu, C. Fan, Y. Li, and J. Hu, "A novel perceptual loss function for single image super-resolution," Multimedia Tools and Applications, vol.79, no.1, pp.21265–21278, 2020. DOI:10.1007/s11042-020-08878-7
[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol.86, no.11, pp.2278–2324, 1998. DOI:10.1109/5.726791
[16] Z. Zhao, S. Yang, and Z. Ma, "License plate character recognition based on convolutional neural network LeNet-5," Journal of System Simulation, 2010. DOI:10.3724/SP.J.1187.2010.00953
[17] X. Qin, Z. Tao, X. Wang, and X. Dong, "License plate recognition based on improved BP neural network," International Conference on Computer, vol.5, pp.171–174, 2010. DOI:10.1109/CMCE.2010.5610030
[18] S. Zherzdev and A. Gruzdev, "LPRNet: License plate recognition via deep neural networks," Computer Vision and Pattern Recognition, 2018
[19] G. Bi, J. Ren, T. Fu, T. Nie, C. Chen, and N. Zhang, "Image Dehazing Based on Accurate Estimation of Transmission in the Atmospheric Scattering Model," IEEE Photonics Journal, vol.9, no.4, pp.1–18, 2017. DOI:10.1109/JPHOT.2017.2726107
[20] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a Fast and Flexible Solution for CNN based Image Denoising," IEEE Transactions on Image Processing, vol.27, no.9, pp.4608–4622, 2018. DOI: 10.1109/TIP.2018.2839891
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.770–778, 2016. DOI:10.1109/CVPR.2016.90
[22] Y. Zhou, Z. Chen, B. Sheng, P. Li, J. Kim, and E. Wu, "AFF-Dehazing: Attention-based feature fusion network for low-light image Dehazing," Computer Animation and Virtual Worlds, vol.32, no.3-4, 2021. DOI:10.1002/cav.2011
[23] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single Image Dehazing via Multi-scale Convolutional Neural Networks," European Conference on Computer Vision, vol.9906, pp.154–169, 2016 DOI:10.1007/978-3-319-46475-6_10
[24] J. Wang, J. Zhuang, L. Duan, and W. Cheng, "A multi-scale convolution neural network for featureless fault diagnosis," 2016 International Symposium on Flexible Automation (ISFA), pp.65–70, 2016. DOI:10.1109/ISFA.2016.7790137
[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Computer Science, 2014.
[26] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.6517–6525, 2017. DOI:10.1109/CVPR.2017.690
[27] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Computer Vision and Pattern Recognition, 2018.
[28] S. Loffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," Machine Learning, 2015.
[29] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," pp.936–944, 2017. DOI:10.1109/CVPR.2017.106
[30] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "ST-GAN: Spatial Transformer Generative Adversarial Networks for Image Compositing," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.9455–9464, 2018. DOI:10.1109/CVPR.2018.00985
[31] A. Graves, "Connectionist Temporal Classification," Springer Berlin Heidelberg, vol.385, pp.61–93, 2012. DOI:10.1007/978-3-642-24797-2_7
[32] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," Computer Science, 2015.
[33] T. Zhao, J. Wang, Z. Wang, and C.W. Chen, "SSIM-based coarse-grain scalable video coding," IEEE Transactions on Broadcasting, vol.61, no.2, pp.210–221, 2015. DOI:10.1109/TBC.2015.2424012
[34] M. Malarvel, G. Sethumadhavan, P.C.R. Bhagi, S. Kar, T. Saravanan, and A. Krishnan, "Anisotropic diffusion based denoising on X-radiography images to detect weld defects," Digital Signal Processing, vol.68, pp.112–126, 2017. DOI:10.1016/j.dsp.2017.05.014

Weiguo Zhang is male, professor, his native place is Weinan, Shaanxi. He is mainly engaged in the teaching and research of optimization theory, algorithm analysis and design.

Jiaqi Lu was born in Yuncheng, Shanxi, China in 1997. He received a bachelor's degree in computer science and technology in 2020. He is currently studying for a master's degree in electronic information at Xi'an University of Science and Technology. His research interests include image processing, deep learning and computer vision.

Jing Zhang was born in Xi'an, Shaanxi, China in 1988. She received the B.S. degree in Mathematics department from Northwest University, Shaanxi, China, in 2010, the M.S. degree in 2013 and the Ph.D. degree in 2018 in computer science and technology from Northwest University, Shaanxi, China. From 2018 to 2021, she was a lecturer with the Xi'an University of Science and Technology, Shaanxi, China. Her research interest includes the image processing and 3D reconstruction techniques, fundamental study of signal processing and signal imaging. Dr. Author has been conducting 1 NSFC project and 1 project from Shaanxi Science and Technology Department. She has been published 10 papers in SCI and EI journal.

Xuewen Li was born in 1971. He received the M.E. degree in electric power transmission and automation from Xi'an Institute of Mining and Technology, Shaanxi, China in 1997. His research interest includes computer algorithm analysis, big data analysis, and image processing, published 21 papers in SCI, EI journal.

Qi Zhao was born in Weinan County, Shanxi Province China, in 1998. She received her B.S. degree in network engineering from North University of China in 2019. She is currently pursuing her M.S. degree in computer science and technology at Xi'an University of Science and Technology. Her research interests include deep learning, convolutional neural networks, defects detection and the manufacture of intelligent robots. Miss. Zhao has received awards and honors from the Computer Application Competition in Five North China Provinces.