Application research on improved CGAN in image raindrop removal

Min Zhu1, Chao Fang1, Haibo Du1, Meibin Qi2, Zhiwei Wu1
1School of Electrical Engineering and Automation of Hefei University of Technology, Hefei, People’s Republic of China
2School of Computer Science and Information Engineering of Hefei University of Technology, Hefei, People’s Republic of China

E-mail: haibo.du@hfut.edu.cn

Abstract: Rainy weather can greatly reduce the image quality and hinder the subsequent processing of the image. In order to achieve raindrop removal on rainy images, the single image raindrop removal method based on conditional generative adversarial networks (CGAN) is proposed. In this method, CGAN is used as the basic framework. The network receives the rainy image as an additional condition information and adds Lipschitz constraint on the network. The network model is trained by the combination of condition adversarial loss, content loss, and perception loss to repair the raindrop area and reconstruct the image. The experimental results show that the proposed method has better raindrop removal effect than the existing algorithm and can avoid image blurring on the basis of ensuring the raindrop removal effect.

1 Introduction

In outdoor scenes, the restoration of raindrop images is of great significance to the work of computer vision system. Images affected by raindrops may not only lose important information but also seriously affect the visual effect of images due to the reduced visibility of images. There is an urgent need for effective image raindrops removal technology in image enhancement, target tracking, target recognition, and other fields. It also has great application prospects in industrial production, traffic detection and so on. Therefore, image restoration technology of raindrop images has been widely studied, and the effect of raindrop removal is also improving.

At present, a large number of raindrop removal algorithms have been proposed, the common method of which is to use sparse coding and dictionary learning methods to remove raindrops [1–4]. This method of sparse coding and dictionary learning makes it difficult to remove raindrop from the image when it is very close to the background. Convolutional neural network (CNN) has been developed rapidly in the fields of target detection [5], image classification [6, 7], and image restoration [8, 9], so it is also widely used in the field of image raindrop removal [10, 11].

In this paper, conditional generative adversarial network (CGAN) [12] is used to train the network model using both rainless image and rainy image. The combination of conditional adversarial loss and content loss is used to train the network to achieve raindrop removal. By adding perceptual loss [13] to avoid image blurring, a clearer reconstructed image can be generated. Then, the spectral normalisation [14] method is used to realise the Lipschitz constraint [15], which makes the training of GAN [12] more stable. The raindrop removal effect of the model is validated on synthetic and real raindrop datasets, and compared with the existing raindrop removal algorithms. The experimental results show that the raindrop removal algorithm proposed in this paper is more effective than other raindrop removal algorithms, and the image details are more complete.

2 Related work

2.1 Conditional generative adversarial networks

CGAN is developed on the basis of original GAN, in which generating model \( G \) is used to capture the data distribution and discriminant model \( D \) is used to estimate the probability that a sample comes from real data rather than generated data. Both the generators and discriminators can add additional information \( x \) as a condition, and CGAN is achieved by feeding additional information \( x \) to the discriminator and generator as part of the input layer. In the generation model, the prior input noise \( p(z) \) and conditional information \( x \) are combined to form a joint hidden layer representation. The framework of adversarial training is quite flexible in the way of representation of hidden layer.

Models \( G \) and \( D \) are trained simultaneously: by fixing \( D \), adjusting the parameters of \( G \) to minimise the expectation of \( \log(1 - D(G(z|x))) \), by fixing \( G \), adjusting the parameters of \( D \) to maximise the expectation of \( \log D(y|x) + \log(1 - D(G(z|x))) \). Therefore, the optimisation process of CGAN can be summarised as two-player minimax game with conditional probability:

\[
\begin{align*}
\min_{G} \min_{D} & \quad V(D,G) = E_{z 
\rightarrow P(z)}[\log D(z|x)] + E_{x 
\rightarrow P(x)}[\log(1 - D(G(z|x)))]
\end{align*}
\] (1)

2.2 Lipschitz constraint

Arjovsky et al.’s [15] proposed using Wasserstein Gan (WGAN) to solve the problem of training instability in GAN. Earth-mover (EM) distance was used to measure the distance between real distributed \( P_{r} \) and generated distributed \( P_{G} \), which replaced the traditional JS divergence and KL divergence.

The EM distance can be expressed as

\[
W(P_{r}, P_{G}) = \frac{1}{K} \sup_{1 \leq k \leq K} E_{x \sim P_{r}}[f(x)] - E_{x \sim P_{G}}[f(x)]
\] (2)

The main idea of WGAN is to add a Lipschitz constraint to the discriminator's mapping function, limiting the maximum local variation of a continuous function to be less than the constant \( K \). However, WGAN uses weight clipping to implement the Lipschitz constraint, which constrains all parameters of the discriminator to a threshold range, which tends to cause the parameters of the discriminator to be concentrated in the maximum value in this range. In addition, the setting of the threshold is required to be high. A small setting is easy to cause the gradient to disappear, while a large setting is easy to cause the gradient to explode.

Takeru et al. ‘s proposed using spectral normalisation method to satisfy Lipschitz condition. The discriminator is regarded as a multi-layer network (assume that it is the \( N \) layer), and its input–output relationship can be expressed as
\[ f(x) = D_N W_N \cdots D_1 W_1 x \] (3)

where \( x \) means input, \( f(x) \) means output, the diagonal matrix \( D_N \) is used to represent the function of the activation function. When the corresponding input is negative, the diagonal element is 0; when the corresponding input is positive, the diagonal element is 1. \( W \) is the network parameter matrix.

Requirements of Lipschitz constraint on the gradient of \( f(x) \):

\[ \| \nabla f(x) \| \leq \| D_N W_N^{-1} \cdots D_1 W_1 \| \leq \| D_N \| \| D_N^{-1} \| \cdots \| D_1 \| \| W_1 \| \] (4)

Here, \( W \) represents the gradient of discriminator function and \( \sigma(W) \) represents the largest singular value of the parameter matrix \( W \) of layer \( i \) neural network.

As the SN of the diagonal matrix corresponding to the activation function ReLU is at most 1. In order to satisfy Lipschitz constraints for \( f(x) \), we can normalise (4):

\[ \| \nabla f(x) \| \leq \| D_N W_N^{-1} \cdots D_1 W_1 \| \sigma(W) \leq \| D_N \| \| D_N^{-1} \| \cdots \| D_1 \| \| W_1 \| \sigma(W) \leq 1 \] (5)

Among them, \( \sigma(W) \) is the largest singular value of matrix \( W \). For diagonal matrix \( D \), there is \( \sigma(D) = \max(d_1, \ldots, d_n) \), the largest element on the diagonal element. Thus, (4) can be expressed as:

\[ \| \nabla f(x) \| \leq \sum_{i=1}^{N} \| \sigma(W_i) \| \] (6)

Among them, \( \| \nabla f(x) \| \) represents the gradient of discriminator function and \( \sigma(W) \) represents the largest singular value of the parameter matrix \( W \) of layer \( i \) neural network.

As the SN of the diagonal matrix corresponding to the activation function ReLU is at most 1. In order to satisfy Lipschitz constraints for \( f(x) \), we can normalise (4):

\[ \| \nabla f(x) \| \leq \| D_N W_N^{-1} \cdots D_1 W_1 \| \sigma(W) \leq \| D_N \| \| D_N^{-1} \| \cdots \| D_1 \| \| W_1 \| \sigma(W) \leq 1 \] (7)

Among them, \( D_i(W_N/\sigma(W_N)) \) represents the network parameters of layer \( N \) neural network divided by the maximum singular value of the layer parameter matrix. In this way, each layer of parameters of the discriminator network can satisfy the constraints of Lipschitz = 1. Therefore, SN is used to normalise the parameters of each layer of the discriminator network, so that \( D \) has Lipschitz constraints.

3 Network structure

3.1 Generator structure

We use U-Net [16] network as the generator, and U-Net network adopts skip connection to splice the low-level features of the encoding layer in the network with the high-level features obtained from the corresponding decoding layer as the input of the next layer. This method can ensure the consistency of the underlying structure of the image to the maximum extent after the rain removal of the input image. Moreover, we introduce dilated convolution [17] into U-Net network to improve the receptive field. Fig. 1 shows the network structure of the generator. Only the encoding layer in the network with the high-level features obtained of layer parameter matrix. In this way, each layer of parameters of the discriminator network can satisfy the constraints of Lipschitz = 1. Therefore, SN is used to normalise the parameters of each layer of the discriminator network, so that \( D \) has Lipschitz constraints.

3.2 Discriminator structure

The discriminator and the generator continuously improve each other's ability by adopting a method of adversarial learning. The goal of the discriminator is to improve the accuracy of identifying true and false samples. Fig. 1 shows the network structure of the discriminator network. SN convolution in the figure indicates SN of the convolutional layer parameters. The activation function of the convolutional layer uses ReLU, and the last layer removes the sigmoid activation function. BN is used in all convolution layers except for the first layer and the last layer. The discriminator includes two types of input of the image output by the generator and the real sample. In order to ensure that the generated samples are sufficiently close to the real sample, a sufficiently similar sample is generated by combining the conditional adversarial loss and content loss. Considering that the image generated by Gan is ambiguous, based on the above loss function, the image is made clearer by adding perceptual loss. The related loss function is introduced one by one below.

4 Loss function

4.1 Generator loss function

The goal of the generator is to ensure that the generated sample is sufficiently real so that the discriminator will consider it to be a real sample. In order to ensure that the generated samples are sufficiently close to the real sample, a sufficiently similar sample is generated by combining the conditional adversarial loss and content loss. Considering that the image generated by Gan is ambiguous, based on the above loss function, the image is made clearer by adding perceptual loss. The related loss function is introduced one by one below.

4.1.1 Conditional adversarial loss: In order to ensure the similarity between the image after raindrop removed and the image with raindrops in the underlying structure, CGAN model is adopted. GAN learns image \( y \) through random phasor \( z \), while CGAN learns image \( y \) through input image \( x \) and random phasor \( z \). Its objective function is as follows:

\[ L_{CGAN} = -E_{x,y,p(x,y)} E_{z,p(z)} D(x, G(x, z)) \] (8)

4.1.2 Content loss: It is hoped that the images generated by GAN are real enough. Therefore, it is considered to increase the content loss to ensure that the area after the raindrop removal is as real as possible, and the part without raindrops is not affected. This loss uses the L1 distance to measure the difference between the true rainless image label and the resulting rainless image. Its objective function is as follows:

\[ L_{L1} = E_{x,y,p(x,y)} z \in p(z) \| y - G(x, z) \|_1 \] (9)

4.1.3 Perception loss: Using the global perceptual loss of the image to measure the global difference between the image after raindrop removal and the real rainless label image, the image features of these images can be obtained from a trained convolutional neural network. Here, the VGG16 pre-training model is adopted. The image features of the generator network output image and the real rainless tag image are extracted by using the VGG-16 network, and then the difference between their image features is measured using the L1 distance. Its objective function is as follows:

\[ L_P = E_{x,y,p(x,y)} z \in p(z) \| V G G(y) - V G G(G(x, z)) \|_1 \] (10)

The whole loss function of the generator can be expressed as:
Among them, $\lambda_1$, $\lambda_2$, $\lambda_3$ are the weight coefficients that balance different loss functions.

4.2 Discriminator loss function

The discriminator needs to score the input real sample and the generated sample. The goal of the discriminator is to give a lower evaluation of the generated sample and a higher evaluation of the real sample. The discriminator's loss includes the real sample and the generated sample loss. As the conditions are introduced in the discriminator, the real sample loss is $E_{(x,y)} - p(x,y)D(x,y)$. The generated sample loss is $E_{z} - p(z)G(x,z)D(G(x,z), y)$.

The whole loss function of the discriminator can be expressed as:

$$L_D = E_{z} - p(z)G(x,z)D(G(x,z), y) - E_{(x,y)} - p(x,y)D(x,y)$$

5 Experimental results and analysis

5.1 Synthetic raindrop dataset

It is difficult to obtain a large number of datasets containing clean/rainy image pairs in actual data. Therefore, the synthetic raindrop dataset is considered to train the network. Zhang et al.'s developed a new raindrop dataset using outdoor images [10]. The dataset contains light, medium, and heavy images of three raindrop intensity levels. The dataset is better in quantity and quality, and it is more difficult to remove raindrops. In order to compare the superiority of the algorithm with other algorithms, the dataset is used to train and test the network model.

In the experiment, 4000 training pictures were used for training the neural network, and 1200 test pictures were used to evaluate the trained network model. Each image is composed of a rainless image and a raindrop image formed by artificially simulating the addition of raindrops through Photo Shop. Raindrops are randomly added in different directions and sizes.

5.2 Neural network parameter setting

The discriminator and the generator are alternately trained once each time by using generative adversarial network in the raindrop removal training dataset. We choose $\lambda_1 = 1$, $\lambda_2 = 0.8$, $\lambda_3 = 0.5$ for the loss function of the generator can get better convergence, setting the size of each batch to 1, iteratively training 100 period, iteratively training 4000 times for each period. The learning rate of the neural network is set to 0.0002, considering that setting the learning rate too large may cause the network gradient to disappear/explode. Therefore, the stochastic gradient descent (SGD) method is used here to reduce the learning rate by multiplying the learning rate by 0.1 for every 10 epochs. The Adam optimisation algorithm is used to optimise the network, with the parameter beat1 set to 0.5 and beat2 set to 0.999. The total number of parameters of $G$ is 61,414 M, and $D$ is 4.797 M. The whole training time takes 6.72 h.

5.3 Analysis of experimental results

First, the effects of different loss functions on the image rain are analysed, and then the rain removal effects of the algorithm and other algorithms are compared on the synthetic raindrop dataset and the real rain image.

5.3.1 Performance analysis of different loss functions: In order to verify the effects of different loss functions on the removal of raindrops, the loss functions involved in the algorithm in this paper are compared and tested. Table 1 shows the mean values of PSNR and SSIM of 100 groups tested by different loss functions after the same number of iterative training. It can be seen from the data in the table that $L_{CGAN}$ loss function can produce higher PSNR value. $L_P$ can improve the SSIM value of the image and preserve the overall structure of the image.

Fig. 2 shows the comparison of processing results of test image from the synthetic raindrop dataset with different loss functions. Among them, the results of raindrop removal by $L_{CGAN}$ show that the addition of only $L_{CGAN}$ will cause severe distortion of image texture structure and colour. In addition, the addition of $L_L$ will overcome the disadvantage of $L_{CGAN}$ to a certain extent and generate fewer artefacts in the effect drawing of rain removal. However, some reconstructed images still cause some fuzzy residuals, and the image will be partially distorted and supersaturated. In addition, adding the perceptual loss function $L_P$ will avoid image blurring to generate a clear image to some extent. Therefore, the combination of $L_{CGAN} + L_L + L_P$ can not only eliminate artefacts but also retain more image details, thus generating more realistic and clearer images.

5.3.2 Synthetic raindrop dataset test: In order to verify the superiority of the algorithm used in this paper compared with other algorithms, it is compared with the results of the literature [4, 10] and the literature [11] algorithm. Fig. 3 shows the results of the rain removal processing for synthesising raindrop image under different
algorithms. In order to clearly observe the raindrop removal effect in detail, the image details reconstructed by raindrop removal process are displayed by enlarging any part of the image. The enlarged portion is marked with a rectangular frame.

Fig. 3a shows the original no-rain image; Fig. 3b shows the synthetic raindrop image to be repaired; Fig. 3c is the result of the literature [4] algorithm; Fig. 3d is the result of the literature [10] algorithm; Fig. 3e is the result of the literature [11] algorithm; and Fig. 3f is the result of the algorithm used in this paper. It can be seen that the algorithm of Fig. 3c does not remove raindrops very well and there is still a lot of residue. Although the algorithm of Fig. 3d basically removes the raindrops, the details of the image are poorly restored and the texture is blurred. Although the method of Fig. 3e has a better rain removal effect, the edge information of the image is lost, and the colour saturation of the image is not good. Fig. 3f is the method used in this paper. The raindrop removal effect is the best and the basic chromatic information of the original rainless image is retained. The effect of removing the raindrop is better than other methods.

PSNR and SSIM were used to evaluate the image quality of these four algorithms. The PSNR and SSIM values of one image in different rain removal algorithms and original rainless tag are given in Table 2, in which the algorithms in literature [c]-[e] in the table, respectively, represent those in literature [4], literature [10], and literature [11]. It can be seen that the two evaluation indexes of the algorithm in this paper are obviously better than the results of other algorithms. It can restore the image well visually. Meanwhile, in Table 2, 100 and 200 artificially added synthetic raindrop images in the test dataset were tested and the average value was calculated. The experimental results show that the PSNR and SSIM values of this algorithm are also very high.

5.3.3 Real rain image test: In order to verify that the algorithm proposed in this paper also has a good removal effect on the raindrops in the real scene, several algorithms are adopted to remove real raindrops actually taken. In Fig. 4, Fig. 4a is a raindrop picture taken in the real scene. Fig. 4b is the result of the literature [4] algorithm. It can be seen that the algorithm only removes some raindrops, and there are still residual raindrops. Fig. 4c is the result of the literature [10] algorithm, which removes a large number of raindrops, but has a low degree of detail reduction and overall image quality deviation. Fig. 4d is the result of the algorithm in literature [10], which removes a large number of raindrops, but has a low degree of detail reduction and overall image quality deviation. Fig. 4e is the result of the algorithm in literature [11]. Although the algorithm removes most of the raindrops, the image is blurred and the visual effect is not good. Fig. 4e is the rain removal result of the algorithm in this paper. It can be seen that this algorithm can complete the rain
removal well and ensure that the edge details and chromaticity information of the image are not lost.

6 Conclusion
In this paper, a new raindrop removal method for single image based on conditional generation adversarial network is proposed. By introducing dilated convolution into the generating model, this method increases the receptive field, improves the ability of neural network, and improves the effectiveness of the raindrop removal algorithm. By increasing the perceptual loss, the high-frequency detail information of the generated image is close to the real image, so that a clearer result with richer details can be obtained. Experimental results show that the proposed method can effectively remove raindrops and preserve image details.

7 Acknowledgments
The work was supported by the National Natural Science Foundation of China (61771180).

8 References
[1] Kang, L.W., Lin, C.W., Fu, Y.H.: ‘Automatic single-image-based rain streaks removal via image decomposition’, IEEE Trans. Image Process., 2011, 21, (4), pp. 1742–1755
[2] Chen, D.Y., Chen, C.C., Kang, L.W.: ‘Visual depth guided color image rain streaks removal using sparse coding’, IEEE Trans. Circuits Syst. Video Technol., 2014, 24, (8), pp. 1430–1455
[3] Luo, Y., Xu, Y., Ji, H.: ‘Removing rain from a single image via discriminative sparse coding’. Proc. IEEE Int. Conf. Computer Vision, Santiago, Chile, December 2015, pp. 3397–3405
[4] Gu, S., Meng, D., Zuo, W., et al.: ‘Joint convolutional analysis and synthesis sparse representation for single image layer separation’. Proc. IEEE Int. Conf. Computer Vision, Venice, Italy, October 2017, pp. 1708–1716
[5] Ren, S., He, K., Girshick, R., et al.: ‘Faster r-CNN: towards real-time object detection with region proposal networks’, Adv. Neural Inf. Process. Syst., 2015, 28, pp. 91–99
[6] He, K., Zhang, X., Ren, S., et al.: ‘Deep residual learning for image recognition’. Proc. IEEE Conf. Computer Vision Pattern Recognition, Las Vegas, NV, USA, June 2016, pp. 770–778
[7] Szegedy, C., Liu, W., Jia, Y., et al.: ‘Going deeper with convolutions’. Proc. IEEE Conf. Computer Vision Pattern Recognition, Boston, MA, USA, June 2015, pp. 1–9
[8] Pathak, D., Krahenbuhl, P., Donahue, J., et al.: ‘Context encoders: feature learning by inpainting’. Proc. IEEE Conf. Computer Vision Pattern Recognition, Las Vegas, NV, USA, June 2016, pp. 2536–2544
[9] Yang, C., Lu, X., Lin, Z., et al.: ‘High-resolution image inpainting using multi-scale neural patch synthesis’. Proc. IEEE Conf. Computer Vision Pattern Recognition, Honolulu, HI, USA, July 2017, pp. 6721–6729
[10] Fu, X., Huang, J., Ding, X., et al.: ‘Clearing the skies: a deep network architecture for single-image rain removal’, IEEE Trans. Image Process., 2017, 26, (6), pp. 2944–2956
[11] Zhang, H., Patel, V.M.: ‘Density-aware single image de-raining using a multi-stream dense network’. Proc. IEEE Conf. Computer Vision Pattern Recognition, Salt Lake City, UT, USA, June 2018, pp. 695–704
[12] Goodfellow, I., Pouget-Abadie, J., Mirza, M., et al.: ‘Generative adversarial nets’. Adv. Neural Inf. Process. Syst., 2014, 27, pp. 2672–2680
[13] Johnson, J., Alahi, A., Fei-Fei, L.: ‘Perceptual losses for real-time style transfer and super-resolution’. European Conf. Computer Vision, Amsterdam, The Netherlands, October 2016, pp. 694–711
[14] Miyato, T., Kataoka, T., Koyama, M., et al.: ‘Spectral normalization for generative adversarial networks’. arXiv, 2018, pp. 1–26
[15] Gulrajani, I., Ahmed, F., Arjovsky, M., et al.: ‘Improved training of wasserstein gan’. Adv. Neural Inf. Process. Syst., 2017, 30, pp. 5767–5777
[16] Ronneberger, O., Fischer, P., Brox, T.: ‘U-net: convolutional networks for biomedical image segmentation’. Int. Conf. Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, October 2015, pp. 234–241
[17] Yu, F., Koltun, V.: ‘Multi-scale context aggregation by dilated convolutions’. arXiv, 2015, pp. 1–9

Fig. 4 Raindrop removal effect of real image
(a) A raindrop picture taken in the real scene, (b) The result of the algorithm in literature [4], (c) The result of the algorithm in literature [10], (d) The result of the algorithm in literature [11], (e) The removal result of the algorithm in this paper