EURASIP Journal on Advances
in Signal Processing

RESEARCH Open Access

Dynamic scene deblurring and image de-raining based on generative adversarial networks and transfer learning for Internet of vehicle

Bingcai Wei[1], Liye Zhang[1,2*], Kangtao Wang[1], Qun Kong[1] and Zhuang Wang[1]

*Correspondence:
zhangliye@sdut.edu.cn
[2] Shandong Big Data Development and Innovation Laboratory, Zibo, China
Full list of author information is available at the end of the article

Abstract

Extracting traffic information from images plays an increasingly significant role in Internet of vehicle. However, due to the high-speed movement and bumps of the vehicle, the image will be blurred during image acquisition. In addition, in rainy days, as a result of the rain attached to the lens, the target will be blocked by rain, and the image will be distorted. These problems have caused great obstacles for extracting key information from transportation images, which will affect the real-time judgment of vehicle control system on road conditions, and further cause decision-making errors of the system and even have a bearing on traffic accidents. In this paper, we propose a motion-blurred restoration and rain removal algorithm for IoV based on generative adversarial network and transfer learning. Dynamic scene deblurring and image de-raining are both among the challenging classical research directions in low-level vision tasks. For both tasks, firstly, instead of using ReLU in a conventional residual block, we designed a residual block containing three 256-channel convolutional layers, and we used the Leaky-ReLU activation function. Secondly, we used generative adversarial networks for the image deblurring task with our Resblocks, as well as the image de-raining task. Thirdly, experimental results on the synthetic blur dataset GOPRO and the real blur dataset RealBlur confirm the effectiveness of our model for image deblurring. Finally, as an image de-raining task based on transfer learning, we can fine-tune the pre-trained model with less training data and show good results on several datasets used for image rain removal.

Keywords: Internet of vehicle, Image deblurring, Image de-raining, Generative adversarial network, Transfer learning

1 Introduction

With the rapid development of automobile industry and the increasing number of vehicles, traffic safety and management problems have become more and more prominent. In order to improve the efficiency of road traffic, ensure the safety of drivers and vehicles and realize smart city and intelligent traffic, the interconnection between vehicles has become one of the key technologies. Therefore, the Internet of vehicle

(IoV) technology has been proposed by researchers. In the IoV technology, the information of vehicles, roads and personnel can be collected by sensors such as radar and camera, which can realize real-time monitoring of road traffic conditions, detect vehicle and pedestrian information, and use communication technology to share information with other vehicles. As a considerable part of the Internet of Things, IoV can use multifarious communication technologies for data interconnection [1, 2]. Finally, the connection of vehicle–road, vehicle–people and vehicle–vehicle is established to guarantee the safety of people and vehicles and the healthy operation of traffic. Among all the data acquisition technologies, computer vision technology has advantages over other technologies in terms of cost, interactivity and security, which has attracted the attention of researchers. Visual-based traffic information extraction has become one of the indispensable capabilities of vehicles. Vision-based information perception capabilities, such as pedestrian and vehicle detection, recognition and instance segmentation, require accurate feature learning of urban street scene images.

Transportation images contain a great many of major information, such as the number of vehicles and pedestrians, vehicle license plate number and traffic signs. This information is of great significance for traffic monitoring and automatic driving. Since the communication rate of the Internet of Things is limited by various conditions, the key information in the image must be extracted to reduce the amount of data transmitted [3–5]. As for the high-speed motion of the vehicle, the relative motion between the camera and the object is very remarkable in the short exposure time, and the turbulence of the vehicle causes the camera to vibrate, resulting in blurred motion in the image captured by the camera. Therefore, it is very difficult to extract information from the blurred image. In addition, the vehicle is usually located outside, in rainy weather, the camera will be blocked by rain, which seriously reduces the visual quality of the image and hinders the background objects. These visibility degradations have a negative affect on image feature learning, resulting in the failure of many computer vision systems. These make removing undesirable visual effects from images caused by movement and rain a very desirable technique.

In order to improve the quality of transportation images by image deblurring and de-raining technologies, researchers proposed a variety of algorithms. The previous image deblurring algorithms first estimated the blur kernel of the image from the given image before deblurring. Secondly, assuming that the image has a certain blur kernel, the image prior information is used in the process of image deblurring. Finally, the image deblurring is realized and the clear image is obtained. In recent years, deep learning algorithms represented by convolutional neural network (CNN) have been heavily applied to the field of image blind deblurring. Compared with earlier blind deblurring algorithms based on image prior, deep learning algorithms can achieve better results than them. For example, Xu et al. [6] introduced a novel, separable structure of convolutional structure for deconvolution and achieved good deblurring results, and Su et al. [7] used CNN for end-to-end training to achieve video deblurring using frame-to-frame information in video. Similarly, due to the rapid development of deep learning algorithm, especially the excellent performance of CNN, numerous methods based on deep learning have emerged in the field of image de-raining. The aim of these methods is to learn and then obtain a mapping function between rain image and clear image, and thus solve the problem of image de-raining based on this mapping function [8].

However, before the appearance of RealBlur dataset [9], all the training dataset used by image deblurring algorithm based on deep learning technology is synthetic, such as GOPRO dataset [10]. This is because a single imaging device (such as a camera or smartphone) cannot capture a blurred image while capturing a sharp image, but the model training of deep learning requires blurred-sharp image pairs, with blurred images as the inputs and clear images as the labels.

In 2014, Goodfellow et al. [11] proposed generative adversarial networks (GAN), they took the lead in generating handwritten numbers and face images from pre-sampled random noise through a multilayer perceptron network. GAN consists of two competing networks, one called generator and the other called discriminator. The generator is responsible for receiving random noise inputs and then synthesizing data samples. The synthetic data should be as real as possible in order to "fool" the discriminator. The discriminator is responsible for determining whether the input data is a "fake" sample synthesized by the generator or a real sample, in order to distinguish them. The goal of a good generative adversarial network is to make the probability of the discriminator close to 0.5, that is, impossible to judge whether this sample is a false sample generated by the generator or real data.

Due to its powerful performance, GAN was soon applied to the field of image deblurring. The generator in GAN is responsible for receiving and recovering the blurred image, and generating a deblurring image similar to a clear image to deceive the discriminator, and the discriminator receives the original clear image and the generator's deblurring image, respectively, and tries to distinguish them. GAN have been applied within the field of image deblurring, such as DeblurGAN proposed by Kupyn et al. [12], conditional GAN and content loss function (CLF) are used in this algorithm to eliminate the blurring of motion images. Due to the synthetic blurred images obtained by a single neural network on blurred and clear images cannot accurately simulate the blurring process of real scenes, two GANs were used to deblur the motion-blurred images, one GAN is used for image blurring and the other is used for image deblurring, and finally the real blurring and deblurring process is realized [13].

Transfer learning is to apply knowledge or patterns learned in a certain domain or task to different but related fields or problems [14, 15]. With the help of original domain data, transfer learning can reduce the dependence on target domain data. For the two tasks of image deblurring and image de-raining, they have a lot of similarity characteristics. Therefore, after using the blurred dataset to train the deep learning neural network, with the help of the transfer learning algorithm, we can save a large amount of calculation and data, and achieve better learning effect in image de-raining.

Inspired by the previous research results, the contributions of this paper are as follows. Firstly, we design a residual block which contains three convolution layers with 256 channels, and we use the LReLU activation function to replace the ReLU in the traditional residual block; Secondly, the proposed residual block is applied to GAN, and the GAN is used to train the image deblurring task. Thirdly, the experimental results on synthetic blur dataset GOPRO and real blur dataset RealBlur verify the effectiveness of our model. Finally, after fine-tuning, the pre-trained model is used to implement the image de-raining task based on transfer learning, which shows good results on multiple datasets.

2 Related works

2.1 Image blurring: synthetic and real

To synthesize a blurred images, it can be generated by frame-by-frame blurring. For blurred spatially varying images, there is no camera response function (CRF) estimation technique [16], and the CRF can be approximated as a known CRF, as shown in Eq. (1).

$$g(I_{S[i]}) = I_{{S'[i]}^{\frac{1}{\gamma}}}. \tag{1}$$

In the above formula, γ is a parameter, it is generally considered to be equal to 2.2, and the real clear image $I_{S[i]}$ can be obtained from the observed clear image $I_{S'[i]}$.

The blurred image of the simulation can then be obtained by the following equation.

$$I_B \simeq g\left(\frac{1}{M}\sum_{t=1}^{M} I_{S[t]}\right) \tag{2}$$

where M is the number of sharp frames.

The real-world images are captured during continuous exposure, so a true blur is the integration of multiple clear frames. This can be expressed as:

$$I_B = g\left(\frac{1}{T}\int_{t=0}^{T} I_{S(t)}dt\right). \tag{3}$$

Before Jaesung Rim et al. proposed RealBlur, a real-world blur dataset that can be used for deep learning, deep learning methods in the field of image deblurring were mostly trained and tested on synthetic simulated blur datasets, where non-professional image collection devices could only save blurred images and not the corresponding clear images when taking blurred images. The RealBlur dataset is the first publicly available real blur dataset that can be used for deep learning, using multiple cameras and optical devices to capture both blurred and clear image pairs using professional image collection devices.

2.2 Image deblurring

Image deblurring is the operation of recovering, or deblurring, a given blurred image to obtain the corresponding clear image.

Non-blind deblurring refers to the deblurring of an image by a given known blur kernel, while the blind deblurring problem refers to the estimation of the original image X and the blur kernel Z from a given noisy image Y.

The blind deblurring process can be expressed as:

$$\{\hat{X}, \hat{Z}\} = argmin\|Z \oplus X - Y\|_2^2 + \phi(X) + \theta(Z) \tag{4}$$

where $\phi(X)$ and $\theta(Z)$ are the regularization terms and possible blur kernels of the expected clear image, respectively.

Non-blind deblurring refers to the process of recovering clear images according to the blur kernel when the blur kernel is known. Blind deblurring algorithm refers to the process of estimating the blur kernel and clear image only through the blur image, or directly abandoning the estimation process of the blur kernel to directly restore the clear image when the blur kernel is unknown. From the perspective of application, non-blind deblurring is more widely used. It also can be divided into traditional optimization-based methods and deep learning-based methods, especially the convolution neural network. All most of the deblurring methods based on deep learning do not need to estimate the blur kernel. A large majority of traditional optimization-based deblurring methods guide the maximum posteriori probability (MAP) process by assuming a priori [17]. For example, Pan et al. [18] proposed an image deblurring method using dark channel prior, Xu et al. [19] proposed a L0 gradient prior with clear edge information. Due to the rapid development of deep learning in recent years and the strong performance of convolution neural network, which is widely used in the field of computer vision, there are also a large number of methods using deep learning and convolution neural network in the field of image deblurring, which can be roughly divided into the methods of using convolution neural network to estimate the blur kernel. For example, Sun et al. [20] used CNN to estimate the probability distribution of the blur kernel. With the end-to-end deblurring method without estimating the blur kernel, most of the deblurring methods using CNN do not need to estimate the blur kernel, and directly realize the image deblurring process, such as Gong et al. [21] proposed a fully convolutional neural network, which can estimate the blur kernel at pixel level. Nah et al. [10] proposed a multi-scale convolutional neural network, which can use the feature association between images of different sizes, such as 64 × 64, 128 × 128, 256 × 256, to achieve more refined image deblurring. Tao et al. [22] also selected the multi-scale structure, and added LSTM (long short-term memory) into it to propose a multi-scale recurrent neural network. However, since the multi-scale convolutional neural network from coarse to fine takes more time in deconvolution operation, Zhang et al. [23] proposed a multi-slice structure, which cuts the input image into multiple slices, and at the same time, multiple encoders and decoders can be arranged in multiple ways to achieve better deblurring performance. The application of image deblurring in IoV, such as filter-DeblurGAN proposed by Zhou et al. [24], filter-DeblurGAN can judge whether the image is blurred, and can be directly applied to the vehicle logo detection (VLD) task. With the vehicle logo detection method, the motion-blurred vehicle logo image can be directly detected.

2.3 GAN

GAN (generative adversarial networks) is a deep learning model, which is one of the most promising unsupervised learning methods on complex distribution in recent years. In the early generative adversarial network, generator and discriminator are not required to be neural networks, and only the corresponding generative and adversarial functions are required. GAN belongs to the generative model (GM), which can be used for the modeling of supervised learning, semi-supervised learning and unsupervised learning. Its application in the field of image can be divided into generating images from random noise or text, and completing the conversion from image to image. At the same time, image restoration can be regarded as the conversion from low-quality image to clear image. The structure of GAN is shown in Fig. 1. GAN contains two competing networks generator and discriminator. The idea of confrontation in GAN can be traced back to Nash equilibrium of game theory. The two sides of confrontation are generator and discriminator. The generator is responsible for generating samples as close as possible to the target image and cheating the discriminator, while the discriminator is responsible for distinguishing the generated image from the samples generated by the generator. The objective function of the confrontation can be described as follows:

$$\min_G \max_D V(D,G) = E_{x\sim P_{data(x)}}[\log D(x)] + E_{z\sim P_{z(x)}}[\log(1-D(G(z)))]. \tag{5}$$

x indicates that the real sample comes from the real data distribution $P_{data(x)}$, $E_{x\sim P_{data(x)}}$ is the expectation of inputting clear images. $D(\cdot)$ represents the output of D, $G(\cdot)$ represents the output of G. The aim of G is to minimize it while the aim of D is to maximize it.

Since its invention, GAN has been the focus of research in the field of deep learning, and there are various variants, such as Wasserstein GAN (WGAN) [25], which improves GAN in terms of loss function and training strategy, proposes the $Earth - Mover(EM)$ distance $W(q,p)$, which is informally defined as the minimum cost of transporting mass in order to transform the distribution q into the distribution p. They also proposed a constraint strategy which can make the discriminator get rid of pattern collapse. The value function of WGAN is as follows:

$$\min_G \max_{D\in\mathcal{D}} \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] - \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] \tag{6}$$

where $\mathcal{D}$ is the set of 1-Lipschitz functions and $\mathbb{P}_g$ is once again the model distribution implicitly defined by $\tilde{x} = G(z), z \sim p(z)$. By clipping the weights of the discriminator to lie within a compact space $[-c, c]$, they let the discriminator follows the 1-Lipschitz constraint, with the help of EM distance, WGAN further improving the performance of GAN; WGAN-GP [26], which improves on WGAN, effectively preventing the gradient disappearance, gradient explosion and the difficulty of weight restriction that may occur in WGAN, etc. They found that the weights clipping strategy of WGAN pushes weights toward two values, so they propose gradient penalty strategy to enforce the 1-Lipschitz constraint. The value function of the discriminator of WGAN-GP is as follows:

$$L = \mathop{\mathbb{E}}_{\tilde{x}\sim\mathbb{P}_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim\mathbb{P}_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}[((\|\nabla_{\hat{x}} D(\hat{x})\|_2) - 1)^2] \tag{7}$$

where the first two part is the same as WGAN without the 1-Lipschitz constraint, while the last part of L, which is gradient penalty loss, can enforce the 1-Lipschitz constraint.

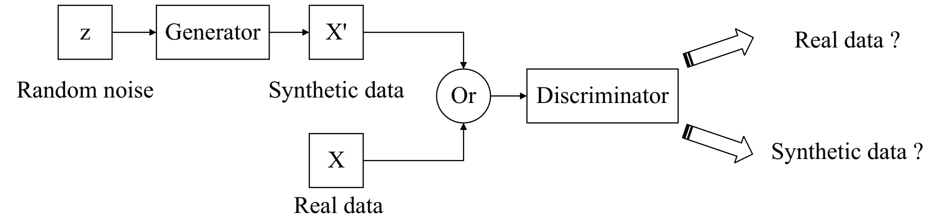

Fig. 1 Structure of GAN

$\mathbb{P}_{\hat{x}}$ is sampling uniformly along straight lines between pairs of points sampled from the data distribution $\mathbb{P}_r$ and the generator distribution $\mathbb{P}_g$. λ is set to 10 in original WGAN-GP.

2.4 Transfer learning

In this work, our transfer learning component focuses on the transformation from image deblurring to image de-raining. We found that using transfer learning, the model pre-trained by a large number of image deblurring can achieve good results in image de-raining tasks which is similar to image deblurring after fine-tuning with a small amount of image de-raining datasets.

2.5 Image de-raining

In simple and broad terms, the band rain image model can be defined as:

$$Y = X + W \tag{8}$$

where Y represents the rain-bearing image, X represents the original clear image, and W represents rain-streak component. Thus, the goal of a single image de-raining task is to recover the original clear image X from a given rain-bearing image Y. Similar to image deblurring, which is also a low-level vision task, earlier image de-raining methods also used image priors to solve it. For example, sparse coding-based methods [27], GMM-based (Gaussian mixture model) [28]-based methods and patchrank prior methods [29]. Applications of single image de-raining in IoV, such as Sun et al. [24], proposed a convolution neural network with rainy images as input, which can directly recover clear images in the case of atmospheric veiling effects caused by distant rain-streak accumulation.

3 Methods

3.1 The proposed resblocks

The residual block [30] consists of two weight layers, including a ReLU activation function in the middle, then a shortcut, and then a ReLU activation function. The shortcut can realize the cross-layer propagation of gradient, which is helpful to overcome the gradient attenuation phenomenon. By adding residual blocks, the problem of gradient vanishing and gradient exploding can be solved with deepening network structure. Due to the strong performance of residual block and its effective solution to the problems existing in deep neural networks, residual block is soon used as the backbone block of most deep neural networks.

Shortcut, also known as residual connection, can establish a direct connection between weight layers separated by multiple weight layers. With this residual connection, even if the gradient in the middle weight layer disappears, this direct connection can ensure the existence of the gradient in the process of gradient propagation. At the same time, the combination of multiple residual blocks can ensure that the output results of the final output layer take into account the shallow layer of the network, in other words, deepen the connection of the whole network. Meanwhile, the combination of multiple residual blocks can also be regarded as an integrated learning. Calling different shortcuts each time can change the network into a combination of networks with different weight layers.

Our residual block in this paper consists of three convolutional layers, each with 256 channels, two Leaky-ReLU activation functions are used, which allows for faster convergence, and a Dropout layer with probability 0.5 is added between the first and second convolutional layers, which helps prevent overfitting of the model while speeding up model training. Finally, there is a jump connection module which helps to solve the gradient disappearance problem as well as the gradient explosion problem. Also, since BN layers have been shown to increase computational complexity and degrade performance [10, 13], the discriminator in this paper removes the Batch Normalization (BN) layer, while most of the studies in this research area that use deep learning for deblurring use small batches for training, such as Nah et al. [10] with a training batch of 2 and Kupyn et al. [12] proposed DeblurGAN, with a training batch of 1, and Zhang et al. [13] proposed realistic blur-based deblurring with a batch of 4. The use of small batches for training is not suitable for using batch normalization layers. The structure of original residual block [30], residual block in [10], and our residual block is shown in Fig. 2. All the channel of conv layers in the Resblock is 256. The structure of some Resblocks is shown in Fig. 2.

Our residual block in this paper consists of three convolutional layers, each with 256 channels, two Leaky-ReLU activation functions are used, which allows for faster convergence, and a Dropout layer with probability 0.5 is added between the first and second convolutional layers, which helps prevent overfitting of the model while speeding up model training. Finally, there is a jump connection module which helps to solve the gradient disappearance problem as well as the gradient explosion problem. Also, since BN layers have been shown to increase computational complexity and degrade performance [10, 13], the discriminator in this paper removes the Batch Normalization (BN)

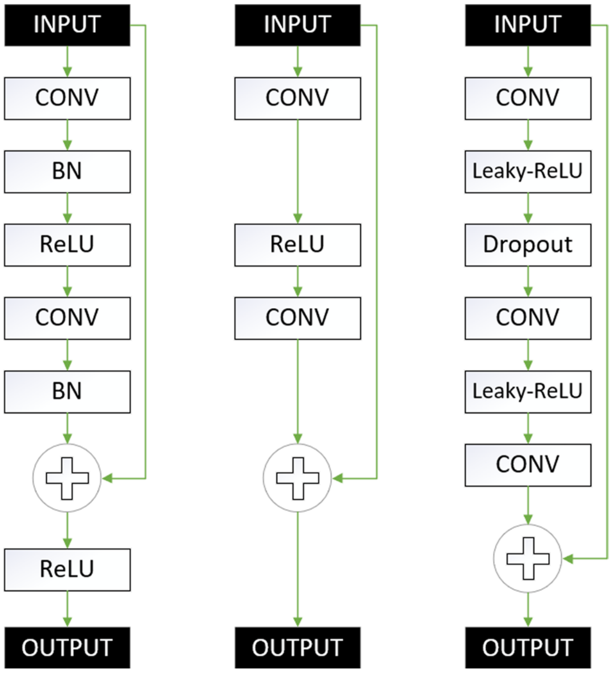

Fig. 2 Structure of some Resblocks

layer, while most of the studies in this research area that use deep learning for deblurring use small batches for training, such as Nah et al. [10] with a training batch of 2 and Kupyn et al. [12] proposed DeblurGAN, with a training batch of 1, and Zhang et al. [13] proposed realistic blur-based deblurring with a batch of 4. The use of small batches for training is not suitable for using batch normalization layers. The structure of original residual block [30], residual block in [10], and our residual block is shown in Fig. 2. All the channel of conv layers in the Resblock is 256.

3.2 Loss function

In this paper, we use the Wasserstein distance from WGAN [25] as the loss function of the discriminator, which is defined as:

$$L_d = \underset{\tilde{x}\sim\mathbb{P}_g}{\mathbb{E}}[D(\tilde{x})] - \underset{x\sim\mathbb{P}_r}{\mathbb{E}}[D(x)] \tag{9}$$

in the above formula, we do not need to make the discriminator follow the 1-Lipschitz constraint, so the set of 1-Lipschitz functions is not employed in this paper. We also keep the sigmoid layer in the discriminator, so the aim of D in this paper is to output 1 when inputting real image and 0 when inputting fake image output by the generator.

Meanwhile, this paper uses Perceptual loss [31] as the loss function of the generator. Perceptual loss is an L2 loss based on the difference between the CNN feature maps of the generated and target images. Unlike the normal L2 loss, the content loss is defined by the output features of one layer of the pre-trained network.

$$L_x = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\Phi_{i,j}(I^S)_{x,y} - \Phi_{i,j}(G_{\theta_G}(I^B))_{x,y})^2 \tag{10}$$

where $\Phi_{i,j}$ represents the feature map extracted by pre-training the convolutional neural network, the pre-trained model used in this paper is VGG19. $H_{i,j}$ and $W_{i,j}$ represents the size of the feature map.

3.3 Network structure

3.3.1 Structure of generator

The generator in this paper contains a series of convolution layers and our own designed residual blocks. Specific as follows: $C7S1-64, C3S2-128, C3S2-256, RB256\times 10, UC3-128, UC3-64, C7S1-3,$ where $C7S1-k$ represents a 7×7 ConvReLU (Convolution+ReLU) block with stride 1 and k filters,$C3S2-k$ represents a 3×3 ConvReLU block with stride 2 and k filters. $RBk\times n$ denotes k filters and n own designed residual blocks which contain three 3×3 convolution layers, one dropout layer, and two Leaky-ReLU layers rather than ReLU activation function. $UC3-k$ represents an Upsample layer followed by a 3×3 ConvReLU layer. Finally, there is a global skip connection block, and the padding type of all the convolution layers is same. The network structure is shown in Fig. 3.

The network structure is shown in Fig. 3. The first part in Fig. 3 is a convolution layer with convolution core size of 7×7, and the output results are input into two convolution blocks with step length of 2, and six improved residual blocks are located at the

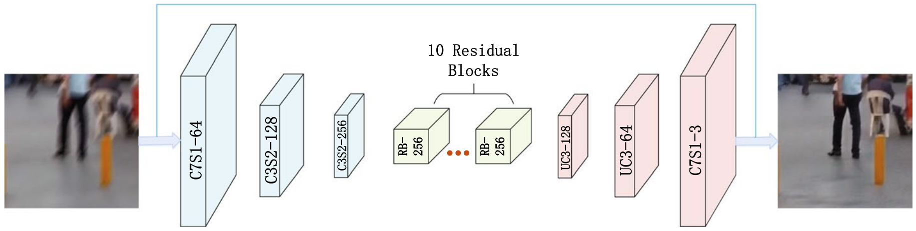

Fig. 3 Structure of generator

third layer, then the system connects two transposed convolution blocks with convolution core size of 3 × 3 with step length of 2. The next layer is a 7 × 7 convolution layer with a tanh activation function as the activation function. In addition to this layer, the generator's activation functions are all ReLU activation functions. Finally, there is a jump connection block, and the filling method of all the convolution layers is the same.

3.3.2 Structure of discriminator

PatchGAN [32] is a Markovian discriminator proposed by Phillip Isola et al. that can efficiently model images as Markov random fields. The PatchGAN discriminator attempts to classify each $N \times N$ block in an image and uses this to determine whether it is a generator-generated fake sample or a real sample, averaging all the eigenvalues of the final layer output to PatchGAN extends the perceptual field of the bottom convolutional layer to 70 × 70 by superimposing five convolutional layers. Inspired by the PatchGAN discriminator, the discriminator structure used in this paper is similar with it, but we add two dense layers at the bottom of it. The structure of discriminator is shown in Table 1.

4 Results and discussion

4.1 Train details

Although GAN has good generative models, the training of the model is not stable enough, resulting in difficulty in convergence and mode collapse. To address these drawbacks, WGAN uses weight pruning to achieve constraints on the discriminator, thus solving the above problems of generative adversarial networks, but it is not completely solved, for example, it is still difficult to converge sometimes. Later, WGAN-GP, which uses a gradient penalty to achieve better performance and further solve the problems of GAN. In contrast to the above mentioned methods of improving GAN's shortcomings, we found that we can also solve the shortcomings of GAN to some extent by a small trick, which we call the gradient training strategy. Specifically, after a certain stage of training, the trained weights of the generator are saved and training is stopped, but the weights of the discriminator are not saved, and then training is restarted and the generator is loaded with the previously trained weights, while the discriminator is trained again. We use an incremental training strategy throughout the training process, and our experiments show that this training technique can address the shortcomings of the GAN to a certain extent without making changes to the network internals during the training process.

The total training period includes 1500 epochs, which can be divided into six parts, the first part includes 500 epochs, the remaining stages each contain 200 epochs. At the end of each stage, stop training and then use the progressive training strategy mentioned above. We use the progressive training strategy throughout the training process, and our experiments show that this training technique can address the shortcomings of the GAN to a certain extent without making changes to the network internals during the training process.

The simulation experiments in this paper were conducted on a server configured with an Nvidia Tesla-P100 using the TensorFlow2 framework with the Adam optimizer, with the initial learning rate set to 10^{-4}. After several iterations of training, the final learning rate decreased linearly to 10^{-7}. During training, images are randomly flipped horizontally and vertically to improve the generalization capability and robustness of the model, while at the same time random Gaussian noise is added to the images during the training process.

Due to the input requirements of the network structure in this paper, the images in the training dataset need to be cropped to a size of 256×256. The generator is a Fully Convolutional Network (FCN), which can be applied to images of arbitrary size.

4.2 Deblurring on GOPRO datasets

GOPRO dataset is one of the most common datasets for image deblurring research. GOPRO4 camera is used to shoot the video with 240 frames per second, and then generate blur images to restore real motion blur. We used 2103 pictures as the training set to train the model, and the remaining 1110 pictures as the test set to test the training effect of the model. The comparison of deblurred images obtained by the proposed method and other methods is displayed in Figs. 4 and 5, and the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) of different algorithms are shown in Table 2. Through the simulation results in Figs. 4 and 5, it can be seen that the results obtained by using the proposed algorithm for blurred image restoration are more clear, and the data of PSNR and SSIM also confirm our simulation results. The PSNR of the proposed algorithm reaches 29.19, and the SSIM reaches 0.883.

4.3 Deblurring on RealBlur datasets

The RealBlur dataset, the first large blurred clear image pair dataset that can be used as a deep learning training dataset, was created by two cameras in a single photoreceptor. The real blur dataset consists of two parts. One part is RealBlur-J, which is a pair of JPEG images formed after camera shooting; the other part is RealBlur-R, which is the untransformed image pair captured by the camera. Some examples of the deblurring results trained and tested on the RealBlur dataset are shown in Figs. 6 and 7, which show that our model can also achieve good deblurring results on the RealBlur dataset. The PSNR and SSIM results are displayed in Table 3, compared with other algorithms, the proposed algorithm has the best PSNR and SSIM results.

Table 1 Structure of discriminator

#	Layer	Size of feature map	Stride	Parameters	Receptive field
1	Input	256 × 256 × 3	–	–	–
2	Conv	128 × 128 × 64	2	3136	142
3	Conv	64 × 64 × 64	2	65,600	70
4	Conv	32 × 32 × 128	2	131,200	34
5	Conv	16 × 16 × 256	2	524,544	16
6	Conv	16 × 16 × 512	1	2,097,644	7
7	Conv	16 × 16 × 1	1	8193	4
8	Dense	–	–	263,168	–
9	Dense	–	–	1025	–

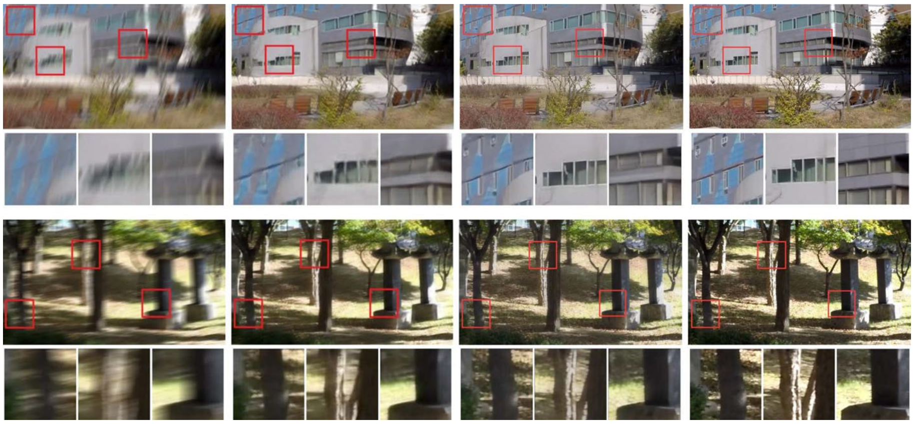

Fig. 4 Results on the GOPRO test dataset. From left to right: blurred, result in [12], our deblurred, sharp photograph

4.4 Image de-raining based on transfer learning

To validate the transfer learning-based image de-raining capability of our proposed model, we fine-tuned and tested it on the following datasets: Rain800 [35], Rain1800 [36]. Note, the test set of Rain1800 has two cases: Rain100L with only one type of rain streaks and Rain100H with five rain streaks. All the above datasets are synthetic image de-raining datasets used for deep learning. On the test set of the Rain800 dataset, we utilize transfer learning to fine-tune on the Rain800 training set using a model that had been previously trained for image deblurring. On the Rain100H dataset, the transfer learning-based image deblurring model just trained on the Rain800 training set was used, and the same on the Rain100L dataset.

The rain removal effect of our model is shown in Table 4 compared with some other models. It is worth noting that in order to quickly apply the models in the image deblurring domain to the image rain removal task, which is in line with the definition of transfer learning and can use less training data, unlike the other models in Table 4 that were trained on Rain14000 dataset [37], Rain800 dataset [35] and Rain1800 dataset [36]. The training set of Rain14000 consists of 11,200 image pairs, it obviously contains much more images compared with Rain800 and Rain1800, which contain 700 and 1,800 image pairs for training, respectively (Figs. 8 and 9). When tested on Rain800 dataset,

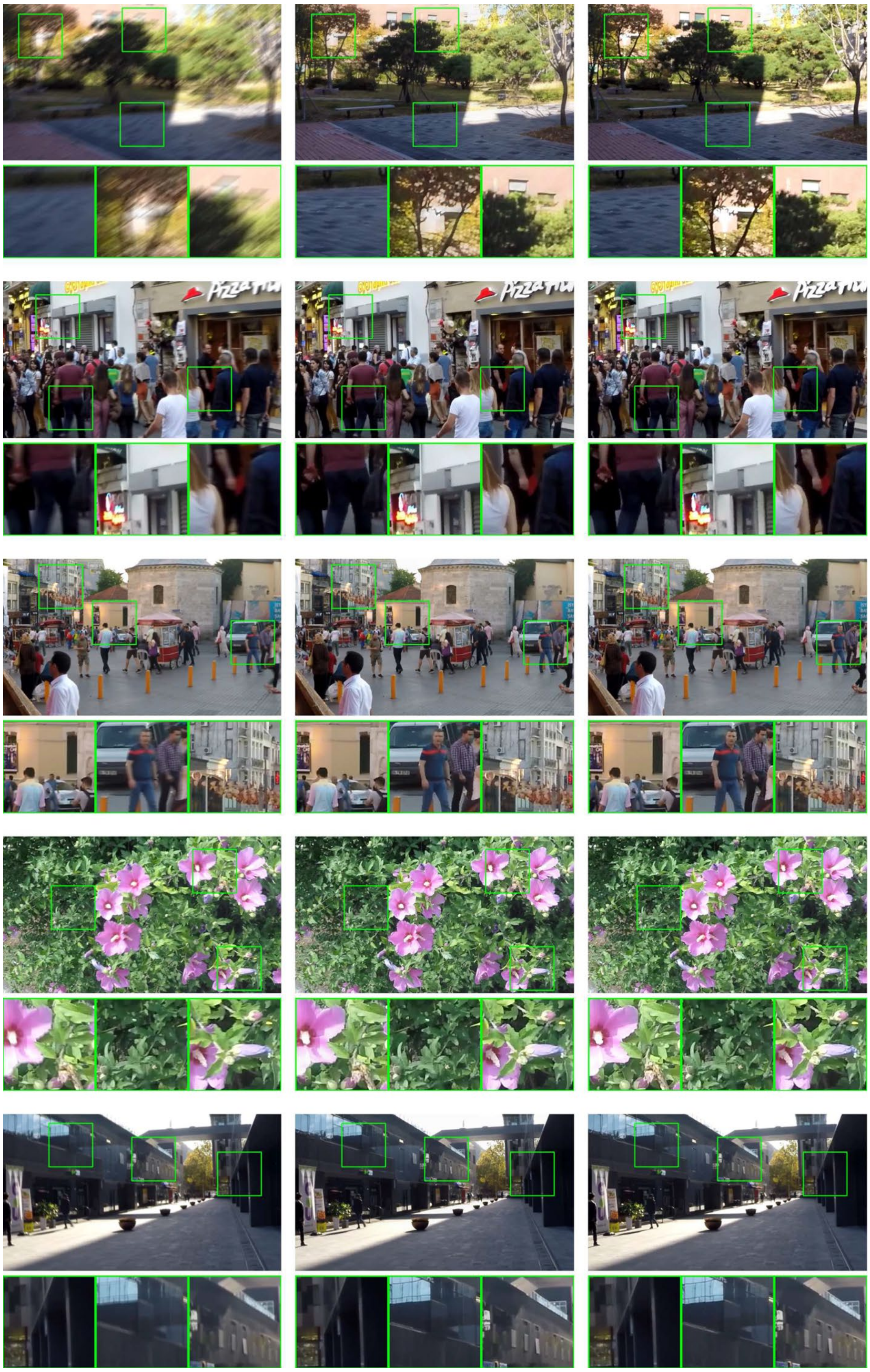

Fig. 5 Results on the GOPRO test dataset. From left to right: blurred, our deblurred, sharp photograph, all green small areas in the pictures in the above different scenes are randomly selected

Table 2 Performance comparison on the GOPRO dataset

Method	Xu [6]	Kim [33]	Sun [20]	Gong [21]	DeblurGAN [12]	Ours
PSNR	21.00	23.64	24.64	26.40	28.70	29.19
SSIM	0.741	0.824	0.843	0.863	0.858	0.883

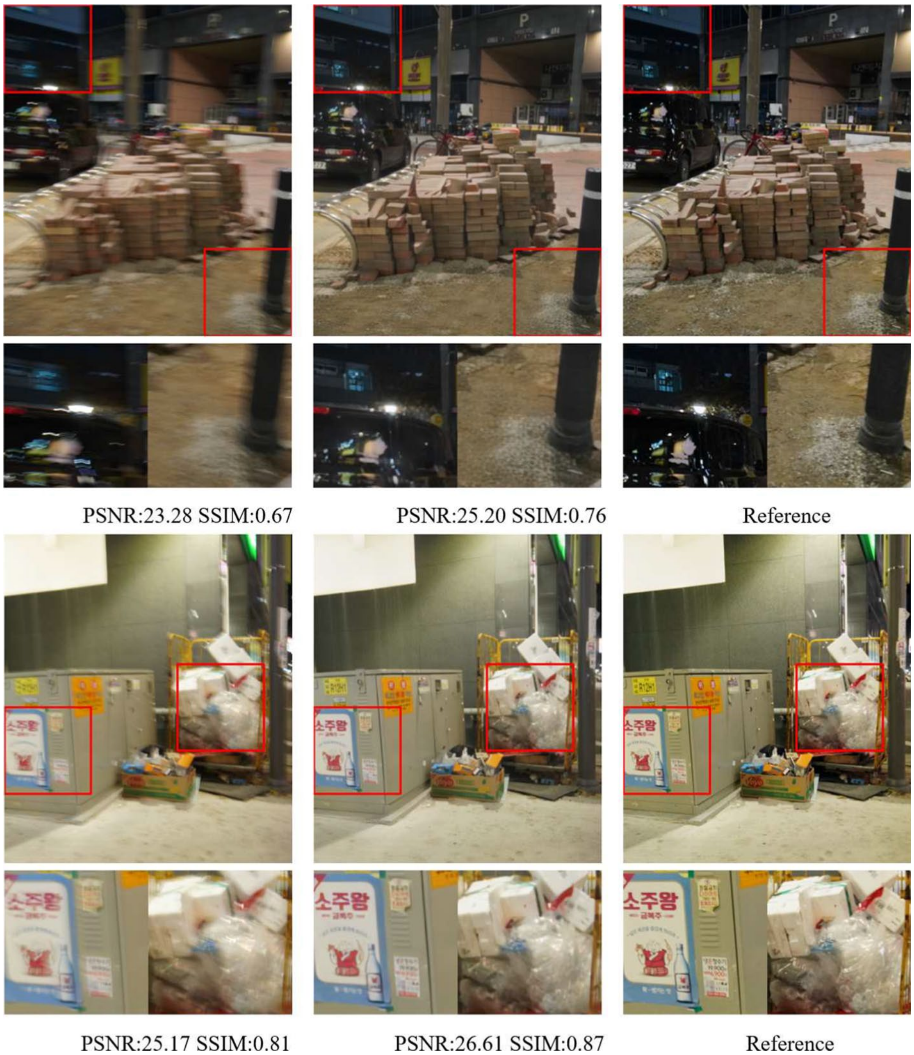

Fig. 6 Results on the RealBlur-J test dataset. From left to right: blurred, our deblurred, sharp photograph

our model was only trained on it. The results shown in Table 4, except for our method, were trained on Rain14000, Rain800 and Rain1800. Our method was never trained on Rain14000 dataset, which has the largest amount of data. As for the test set of Rain1800, it is divided into two parts, Rain100H and Rain100L, however, our model trained only on the Rain800 dataset can achieve PSNR 18.30 and SSIM 0.476 on Rain100H and SSIM 0.865 and PSNR 26.21 on Rain100L.

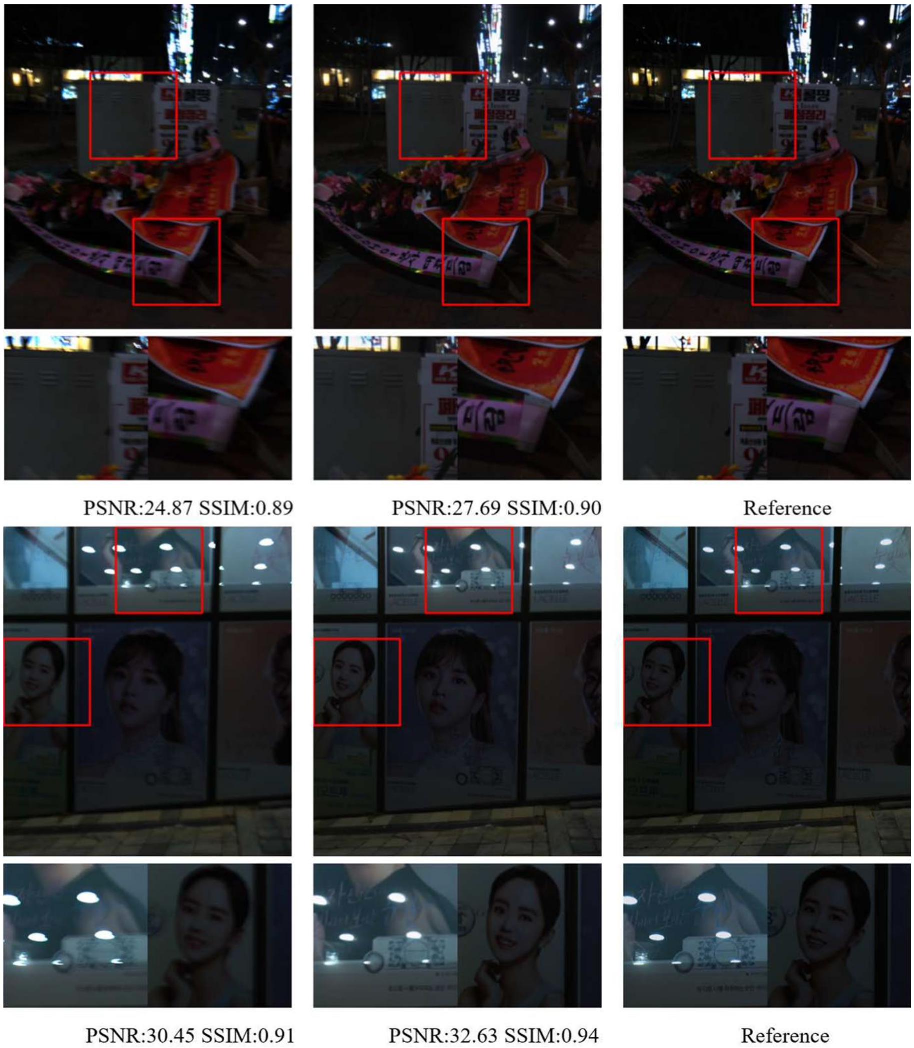

Fig. 7 Results on the RealBlur-R test dataset. From left to right: blurred, our deblurred, sharp photograph

Table 3 Performance comparison on the RealBlur dataset

Method	RealBlur-J		RealBlur-R	
	PSNR	SSIM	PSNR	SSIM
Hu [34]	26.41	0.803	33.67	0.916
Nah [10]	27.87	0.827	32.51	0.841
DeblurGAN [12]	27.97	0.834	33.79	0.903
Pan [18]	27.22	0.790	34.01	0.916
Ours	27.73	0.833	34.32	0.911

5 Conclusions

In IoV, transportation image is one of the important data sources. In order to extract traffic information from motion-blurred images and rain images, the image deblurring method and the image de-raining method based on deep learning and transfer

Table 4 De-raining performance comparison on the several datasets

Method	Rain800		Rain100L		Rain100H	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
DerainNet [38]	22.77	0.810	27.03	0.884	14.92	0.592
SEMI [39]	23.64	0.788	25.03	0.842	16.56	0.486
DIDMDN [8]	22.56	0.818	25.23	0.471	17.35	0.524
Ours	25.50	0.825	29.91	0.934	21.99	0.787

learning are studied in this paper, and a motion deblurring algorithm based on deep residual generation adversarial network with our own designed Resblock is proposed to achieve higher accuracy in the image deblurring task. After the model training, in order to reduce the training workload of the image rain removal model, the trained model is applied to the image de-raining task by transfer learning, and satisfactory results are also achieved. We tested the proposed method on multiple public datasets,

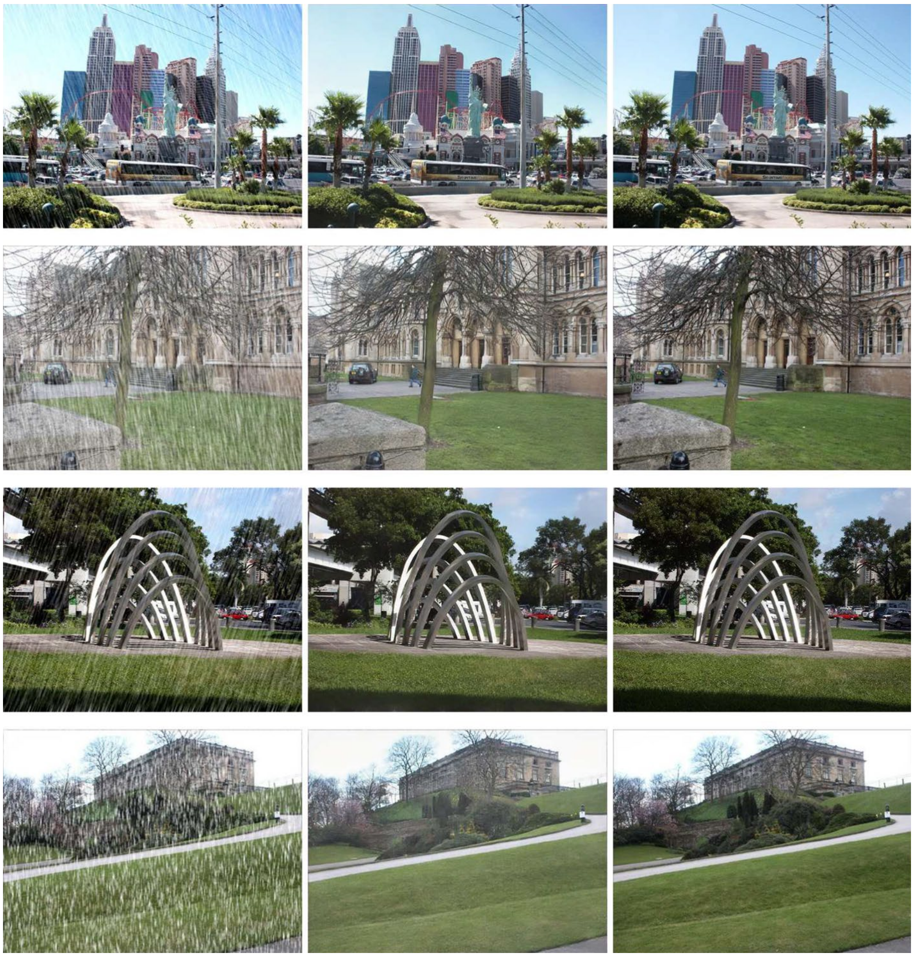

Fig. 8 De-raining results on the Rain800 dataset. From left to right: rained, Our de-raining, no-rain photograph

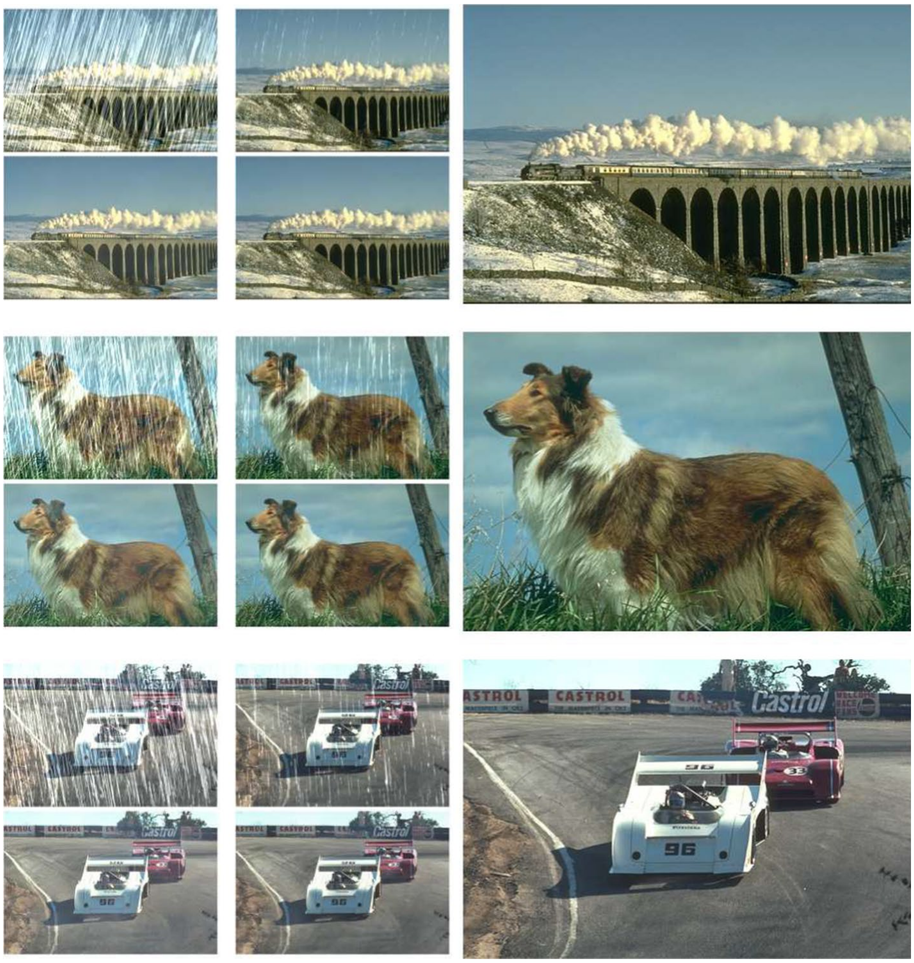

Fig. 9 De-raining results on the Rain100H and Rain100L test dataset. Part a of the image above is the image with rain and the de-raining effect, and part b is the corresponding image without rain. the upper part of part a is the image with rain and the lower part is the corresponding de-raining effect

and the experimental results demonstrate that the proposed method leads to significantly improvements on image deblurring and image de-raining.

Abbreviations

IoV: Internet of vehicle; GAN: Generative adversarial network; LReLU: Leaky rectified linear unit; CNN: Convolutional neural network; CLF: Content loss function; CRF: Camera response function; MAP: Maximum a posteriori probability; LSTM: Long short-term memory; WGAN: Wasserstein GAN; WGAN-GP: WGAN gradient penalty; GMM: Gaussian mixture model; FCN: Fully convolutional network; PSNR: Peak signal-to-noise ratio; SSIM: Structural similarity.

Acknowledgements

This work is supported by School of Computer Science and Technology, Shandong University of Technology.

Authors' contributions

The algorithms proposed in this paper have been conceived by LZ and BW. BW and LZ made the analysis and experiment and wrote the paper. KW, QK and ZW investigated, validated and revised this paper. The authors approved the final manuscript. All authors read and approved the final manuscript.

Authors' Information

Bingcai Weireceived the bachelor's degree in software engineering from Qufu Normal University. He is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, Shandong University of Technology. His current research interests include machine learning and image processing.

Liye Zhang received the M.Sc. and Ph.D. degrees in communication engineering from the Harbin Institute of Technology, in 2011 and 2018, respectively. From 2014 to 2015, he was a Visiting Scholar with Department of Electrical and Computer Engineering, University of Toronto, Canada. He is currently a Lecturer with the Shandong University of Technology. His current research interests include Indoor Localization, Computer Vision and Machine Learning.
Kangtao Wang received his Bachelor of Engineering degree from the School of Software of Pingdingshan College in 2020. He is currently pursuing a master's degree in the School of Computer Science and Technology at Shandong University of Technology. His current research interests include machine learning and image stitching.
Qun Kong received the bachelor's degree in computer science and technology from DeZhou University in 2020. She is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, Shandong University of Technology. Her current research interests include machine learning and binocular vision.
Zhuang Wang received the bachelor's degree in computer science and technology from Shandong Youth University for Political Sciences in 2020. He is currently pursuing the M.Sc. degree with the School of Computer Science and Technology, Shandong University of Technology. His current research interests include machine learning and indoor localization.

Funding

This paper is supported by Shandong Provincial Natural Science Foundation, China (Grant Number ZR2019BF022) and National Natural Science Foundation of China (Grant Number 62001272).

Availability of data materials

All the datasets used for training the model of this paper are from Internet.

Declarations

Ethics approval and consent to participate

Not applicalbe.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1]School of Computer Science and Technology, Shandong University of Technology, Zibo, China. [2]Shandong Big Data Development and Innovation Laboratory, Zibo, China.

Received: 13 September 2021 Accepted: 29 November 2021
Published online: 14 December 2021

References

1. F. Li, K. Lam, X. Liu, J. Wang, K. Zhao, L. Wang, Joint pricing and power allocation for multibeam satellite systems with dynamic game model. IEEE Trans. Veh. Technol. **67**(3), 2398–2408 (2018)
2. X. Liu, X. Zhang, NOMA-based resource allocation for cluster-based cognitive industrial internet of things. IEEE Trans. Ind. Inf. **16**(8), 5379–5388 (2020)
3. X. Liu, X. Zhang, Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer. IEEE Internet Things J. **6**(4), 5971–5980 (2019)
4. X. Liu, X. Zhang, M. Jia et al., 5G-based green broadband communication system design with simultaneous wireless information and power transfer. Phys. Commun. **28**, 130–137 (2018)
5. X. Liu, X. Zhai, L. Weidang, W. Celimuge, QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA. IEEE Trans. Ind. Inf. **17**(3), 2052–2061 (2021)
6. L. Xu, J.S.J. Ren, C. Liu et al., Deep convolutional neural network for image deconvolution. Adv. Neural. Inf. Process. Syst. **2**, 1790–1798 (2014)
7. S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, O. Wang, Deep video deblurring for hand-held cameras. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 237–246 (2017). https://doi.org/10.1109/CVPR.2017.33
8. H. Zhang, V.M. Patel, Density-aware single image de-raining using a multi-stream dense network. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. **2018**, 695–704 (2018). https://doi.org/10.1109/CVPR.2018.00079
9. J. Rim, H. Lee, J. Won, S, Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms, in ECCV (2020)
10. S. Nah, T.H. Kim, K.M. Lee, Deep multi-scale convolutional neural network for dynamic scene deblurring. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 257–265 (2017). https://doi.org/10.1109/CVPR.2017.35
11. I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., Generative adversarial networks. Adv. Neural. Inf. Process. Syst. **27**, 2672–2680 (2014)
12. O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, J. Matas, DeblurGAN: blind motion deblurring using conditional adversarial networks. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. **2018**, 8183–8192 (2018). https://doi.org/10.1109/CVPR.2018.00854
13. K. Zhang et al., Deblurring by realistic blurring. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) **2020**, 2734–2743 (2020). https://doi.org/10.1109/CVPR42600.2020.00281
14. F. Zhuang et al., A comprehensive survey on transfer learning. Proc. IEEE **109**(1), 43–76 (2021). https://doi.org/10.1109/JPROC.2020.3004555

15. R. Ribani, M. Marengoni, A survey of transfer learning for convolutional neural networks, in 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T) (2019). p. 47–57. https://doi.org/10.1109/SIBGRAPI-T.2019.00010
16. Y. Tai et al., Nonlinear camera response functions and image deblurring: theoretical analysis and practice. IEEE Trans. Pattern Anal. Mach. Intell. **35**(10), 2498–2512 (2013). https://doi.org/10.1109/TPAMI.2013.40
17. J. Wu, X. Di, Integrating neural networks into the blind deblurring framework to compete with the end-to-end learning-based methods. IEEE Trans. Image Process. **29**, 6841–6851 (2020). https://doi.org/10.1109/TIP.2020.2994413
18. J. Pan, D. Sun, H. Pfister, M. Yang, Blind image deblurring using dark channel prior. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2016**, 1628–1636 (2016). https://doi.org/10.1109/CVPR.2016.180
19. L. Xu, S. Zheng, J. Jia, Unnatural L0 sparse representation for natural image deblurring. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2013**, 1107–1114 (2013). https://doi.org/10.1109/CVPR.2013.147
20. J. Sun, Wenfei Cao, Zongben Xu, J. Ponce, Learning a convolutional neural network for non-uniform motion blur removal, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2015). p. 769–777. https://doi.org/10.1109/CVPR.2015.7298677
21. D. Gong et al., From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 3806–3815 (2017). https://doi.org/10.1109/CVPR.2017.405
22. X. Tao, H. Gao, X. Shen, J. Wang, J. Jia, Scale-recurrent network for deep image deblurring. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. **2018**, 8174–8182 (2018). https://doi.org/10.1109/CVPR.2018.00853
23. H. Zhang, Y. Dai, H. Li, P. Koniusz, Deep stacked hierarchical multi-patch network for image deblurring. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) **2019**, 5971–5979 (2019). https://doi.org/10.1109/CVPR.2019.00613
24. H. Sun, M.H. Ang, D. Rus, A convolutional network for joint deraining and dehazing from a single image for autonomous driving in rain. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS) **2019**, 962–969 (2019). https://doi.org/10.1109/IROS40897.2019.8967644
25. M. Arjovsky, S. Chintala, L. Bottou, Wasserstein generative adversarial networks, in 34th International Conference on Machine Learning, ICML 2017,v 1, (2017). p. 298–321
26. I. Gulrajani, F. Ahmed, M. Arjovsky, V . Dumoulin, A. Courville, Improved training of Wasserstein GANs advances in neural information processing systems, v 2017-December, p 5768-5778 (2017)
27. D. Huang, L. Kang, Y.F. Wang, C. Lin, Self-learning based image decomposition with applications to single image denoising. IEEE Trans. Multimedia **16**(1), 83–93 (2014). https://doi.org/10.1109/TMM.2013.2284759
28. Y . Li, R. T. Tan, X. Guo, J. Lu, M.S. Brown. Rain streak removal using layer priors, in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016). p. 2736–2744
29. Y.-L. Chen, C.-T. Hsu, A generalized low-rank appearance model for spatio-temporally correlated rain streaks, in IEEE ICCV (2013). p. 1968–1975
30. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2016**, 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
31. J. Johnson, A. Alahi, L. Fei-Fei, Perceptual losses for real-time style transfer and super-resolution, in European Conference on Computer Vision (2016)
32. P. Isola, J. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 5967–5976 (2017). https://doi.org/10.1109/CVPR.2017.632
33. T. Hyun Kim, B. Ahn, K. Mu Lee. Dynamic scene deblurring, in ICCV (2013)
34. Z. Hu, S. Cho, J. Wang, M. Yang, Deblurring low-light images with light streaks. IEEE Conf. Comput. Vis. Pattern Recognit. **2014**, 3382–3389 (2014). https://doi.org/10.1109/CVPR.2014.432
35. H. Zhang, V. Sindagi, V.M. Patel, Image de-raining using a conditional generative adversarial network. IEEE Trans. Circuits Syst. Video Technol. **30**(11), 3943–3956 (2020). https://doi.org/10.1109/TCSVT.2019.2920407
36. W. Yang, R.T. Tan, J. Feng, J. Liu, Z. Guo, S. Yan, Deep joint rain detection and removal from a single image. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 1685–1694 (2017). https://doi.org/10.1109/CVPR.2017.183
37. X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, J. Paisley, Removing rain from single images via a deep detail network. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR) **2017**, 1715–1723 (2017). https://doi.org/10.1109/CVPR.2017.186
38. X. Fu, J. Huang, X. Ding, Y. Liao, J. Paisley, Clearing the skies: a deep network architecture for single-image rain removal. IEEE Trans. Image Process. **26**(6), 2944–2956 (2017). https://doi.org/10.1109/TIP.2017.2691802
39. W. Wei, D. Meng, Q. Zhao, Z. Xu, Y. Wu, Semi-supervised transfer learning for image rain removal. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) **2019**, 3872–3881 (2019). https://doi.org/10.1109/CVPR.2019.00400

Publisher's Note