ISBN: 978-981-18-5852-9
WCSE 2022 Spring Event: 2022 9th International Conference on Industrial Engineering and Applications
doi: 10.18178/wcse.2022.04.057

Depth Separable Convolution Lightweight Blind Motion Deblurring Algorithm

Chi Li [1,2], Weiwei Kong [1,2], Jiawei Xue [1,2] and Ze Wang [1,2]

[1] School of Computer Science and Technology, Xi'an University of Posts and Telecommunications, Xi'an, Shannxi 710121 China

[2] Shaanxi Key Laboratory of Network Data Analysis and Intelligent Processing, Xi'an University of Posts and Telecommunications, Xi'an, Shannxi 710121 China

Abstract. Aiming at the motion blur problem caused by handheld camera shake and object movement in the process of photo acquisition, an image deblur algorithm based on depth separable convolution lightweight generative adversarial network was proposed. Firstly, a depth separable convolution residual block with self-attention mechanism is introduced in the generator. Secondly, Transpose convolution module is replaced by the Pixel-Shuffle up-sampling module. Finally, The combination of perceptual loss and relativistic generative adversarial loss can effectively alleviates the mode collapse problem of traditional GAN and improves the stability of model training. Experimental results show that compared with Deblur-GAN the number of parameters of the proposed algorithm is significantly reduced, the deblurring time of a single image is reduced by 50%, the deblurring image is clearer in subjective vision, and the objective evaluation indexes of image quality such as peak signal-to-noise ratio and structural similarity are also significantly improved.

Keywords: image deblurring, Generative adversarial nets, self-attention, deep learning, depth separable convolution, relativistic average discriminator.

1. Introduction

The task of image deblurring is to recover the clear image from the blurred image. In recent years, it has been applied to medical imaging, criminal case detection, astronomical remote sensing and other fields [1][2], it has become a research hot spot in the field of computer vision and image processing. According to whether the blur kernel is known, it can be generally divided into non-blind deblurring and blind deblurring. Most of the early studies focused on non-blind deblurring, such as Rudin et al. [3] proposed the Total Variation (TV) regularization algorithm, and Xu et al. [4] estimated clear images by piecewise function approximation L0 norm as the image gradient regularization term. Kim et al. [5] proposed TV-L1 model in 2014 and achieved good results.

However, in the actual situation, most of the blur kernel are non-uniformly distributed and unknowable, so it becomes necessary to study the blind area blur algorithm. The traditional blind deblurring algorithm [6][7] is based on the estimation of the blur kernel by statistical algorithm, that is, the prior information of the image is obtained first. The information is modeled and fitted to a function consistent with the prior information distribution of the clear image. The fitted function is used as a blur kernel to estimate the clear image. Michaeli et al. [8] applied an internal patch cross-scale recurrence prior to blind image deblurring. In different scales of clear images, the probability of a certain image block appearing in similar image blocks is much higher than that in blur images. The optimal blur kernel is found by maximizing the cross-scale block similarity. With the rapid development of deep learning technology, more and more deblurring algorithms based on deep learning are proposed. SUN et al. [9] use convolutional neural networks (CNN) to predict the probability distribution of motion blur to estimate the image blur kernel, and recover the clear image according to the blur kernel. But blur kernel estimation takes too long and the result of deblurring depends on blur kernel estimation.

In order to make the image deblurring model more generalized, researchers proposed an end-to-end deblurring algorithm based on neural network. Nah et al. [10] proposed a multi-scale neural network called DeepDeblur, which adopts the end-to-end algorithm from coarse to fine to greatly improve the performance of deblurring algorithm. The model has good deblurring effect and good generalization performance, and can adapt to a variety of blur cases. However, the multi-scale method consumes too much computation power and makes a lot of meaningless calculations. As Goodfellow et al. [11] proposed Generative Adversarial Nets (GAN), GAN has been widely applied in the field of computer vision. Based on GAN and residual network ideas, Kupyn et al. [12] proposed DeblurGAN algorithm, which realizes end-to-end image deblurring based on conditional generative adversarial network. Compared with previous algorithms, the algorithm not only processed images faster, but also achieved better visual effects, higher peak signal-to-noise ratio and structural similarity. Based on DeblurGAN as the basic framework, this paper proposes a lightweight generation adversity-network image deblurring algorithm based on depth separable convolution.

2. The proposed method

The network architecture in this paper adopts generative adversarial network, which was proposed by Goodfellow et al. In 2014. The core idea of generative adversarial network is the game between generator network (G) and discriminator network (D). The generator's job is to take input and produce new data, the discriminator's job is to take real data and generated data, and to differentiate as much as possible. The goal of the generator is to generate enough data to fool the discriminator by constantly updating the network weights, and the discriminator is also updating the network to achieve the optimal discriminant effect.

The model training process is about the minimax game of (1), that is, on the premise that the discriminator can maximize the distinction between true and false data, the distribution gap between generated data and real data is minimized to minimize the value of the whole formula:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_{data(x)}}[\log D(x)] + E_{z\sim p_{z(z)}}[\log(1-D(G(z)))] \quad (1)$$

where E represents mathematical expectation, $X\sim P_{data(x)}$ indicates that data X follows a real data distribution, D(x) represents probability that the discriminator determines that real data is true, $Z\sim P_{data(z)}$ indicates that data Z represents random noise subject to Gaussian distribution, $G(_{Z})$ represents generated data obtained by random noise Z through generator G, and $D(G(_{Z})$ represents probability that the discriminator judges generated data to be true.

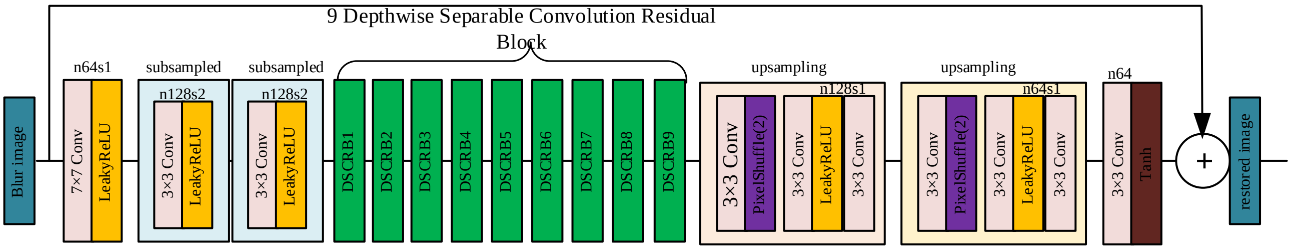

Fig. 1 Text generator network

2.1. Generator Structure

The structure of the generator network in this paper is shown in Fig. 1. The generator inputs blur pictures and outputs the restored pictures after deblurring, and trains the generator network to obtain the optimal weight value by using back propagation, so as to achieve the best deblurring effect of the generator on blur pictures.

First, a 7×7 convolution layer is used to extract low-dimensional features, and then two lower sampling layers with convolution kernels of 3×3 are used for image down-sampling. The activation functions are LeakyReLU and the normalization layer is removed, with the number of convolution kernels of 64, 128, 256 respectively. LeakyReLU function has the advantages of high computational efficiency, back propagation, and fast network convergence. It also solves the problem of neuron death in ReLU function, making the model more stable. In the task of image deblurring [10], it has been proved that removing the normalization layer can improve network performance and reduce computational complexity. BN layers normalize the features using mean and variance in a batch during training and use estimated mean and variance of the whole training datasets during testing. When the statistics of training and testing datasets differ a lot, BN layers tend to introduce unpleasant artifacts and limit the generalization ability. then, 9 depth separable convolution residual blocks (DSCRB) containing the attention layer were used for depth feature extraction with 256 convolution kernels. After that, the feature map passes through two upper sampling layers of PixelShuffles, and the number of convolution kernels is 128, 64 respectively. Finally, the feature map passes through a 7×7 convolution layer and Tan(h) activation function are used to generate a clear image after deblurring. In addition, for the whole network model, residual connection is added between input and output to make the whole generator training better. The following focuses on the DSCRB module.

2.2. DSCRB

Residual blocks adopted in this paper are based on depth separable convolution. Before feature extraction using depth separable convolution, channel attention module and space attention module are added to make the network pay more attention to the specific details of the object in the image and highlight the features of important objects in the image. Resulting in higher quality images. Fig. 2 shows the DSCRB adopted in this paper.

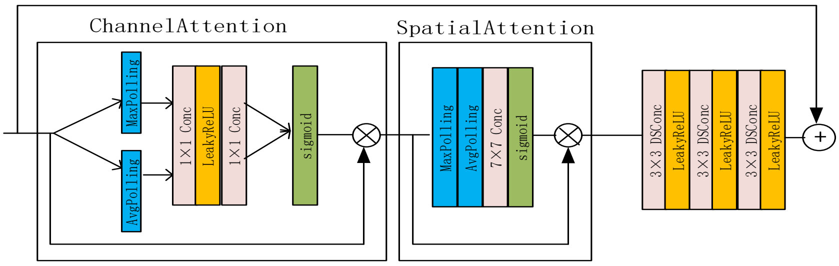

Fig. 2Text DSCRB

Channel attention module aims to focus on which of the input feature maps are meaningful, that is, to judge the importance of the input feature channels, and then allocate appropriate weights so that the network can extract more useful and detailed feature maps. As a supplement of channel module, spatial attention module is designed to focus on the most informative parts of the feature map. And after each module, the generated attention feature map needs to be multiplied by the input feature map of the module for adaptive feature refinement.

In DSCRB, the input feature map F is firstly passed through AvgPoolling and MaxPoolling respectively, get two feature maps and input them into the multi-layer perceptron which activation function is LeakReLU, then add the two feature maps output by the perceptron and activate by Sigmoid to generate the final channel attention feature map M_C. M_C multiplied by F to get F' and input F' into the spatial attention module, put F' into global maximum pooling and global average pooling and make channel splicing of the two feature graphs obtained. And then dimension reduction to one channel by a 7×7 convolution operation, after the Sigmoid function get spatial attention characteristic feature map M_S. Finally, multiply M_S and F' to obtain the final feature map F''. F' through three depths separable convolution and LeakyReLU activation function yields the output of a DSCRB.

2.3. Depth Separable Convolution

Fig. 3 shows the specific operation of standard convolution. Assume that the number of input channels be 3 and the size of convolution kernel be 3×3×3. The process of a convolution kernel completing a convolution is that the convolution kernel is convolution with the corresponding channel map of the input layer respectively, and a channel feature map of the output layer is obtained by adding. The number of output channels is consistent with the number of convolution kernels. In Fig. 3, the number of convolution kernels is 4 in order to obtain four output characteristic plots. To extend to the general situation, set the input feature map is $D_I \times D_I \times M$, output feature map is $D_O \times D_O \times N$, and the size of the convolution kernel as, then the calculation formula of the parameter number of the standard convolution is shown in (2). For the standard convolution process shown in Fig. 3, the number of convolutional layer parameters is 3*3*3*4, namely 108.

$$P_1 = D_K \times D_K \times M \times N \tag{2}$$

However, with the development of deep learning, the neural network layers get deeper and deeper, the network is more and more complex, and the number has increased dramatically, computer overhead also becomes more and more big, make training becomes more and more difficult, the depth separable convolution [13] subsequently put forward, under the condition of ensuring the performance of the model, the problem of large number of network parameters is solved, It makes the motion deblurring algorithm of light weight image feasible.

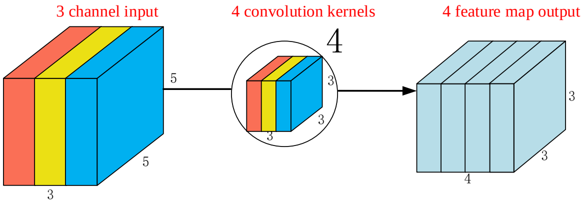

Fig. 3 Schematic diagram of standard convolution

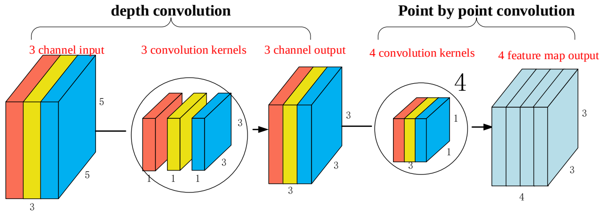

Fig. 4 Schematic diagram of depth separable convolution

The core idea of depth separable convolution is to split standard convolution into deep convolution and point by point convolution. Fig. 4 shows the specific operation of depth separable convolution. Deep convolution operation is to convolution each channel of the input feature map with the convolution kernel of 3×3×1 respectively, and ensure the invariable number of feature graphs while extracting features from the input feature graph. Then, the output of deep convolution is convolved point by point through 1×1×M convolution check, and N feature graphs are finally obtained. The calculation formula of the total amount of depth-separable convolution parameters is shown in (3). For the feature map with input of 5×5×3 as shown in Fig. 4, a feature map of 3×3×4 is generated through depth separable convolution, and the number of parameters of the convolution layer is 3*3*3+1*3*4, equals 39.

Compared with standard convolution, the number of parameters of depth separable convolution is about 1/3 of that of conventional convolution under the same input and output of 4 feature maps. Therefore, on the premise of the same number of parameters, the neural network using depth separable convolution has lower space consumption and is more lightweight.

$$P_2 = D_K \times D_K \times M + 1 \times M \times N \tag{3}$$

The ratio P between depth separable convolution and standard convolution computation is shown in Eq. (4). Using deeply detachable convolution operation can significantly reduce the number of deblurring network parameters and make the network more lightweight.

$$P = \frac{D_K \times D_K \times M + M \times N \times 1}{D_K \times D_K \times M \times N} = \frac{1}{N} + \frac{1}{D_K^2} \tag{4}$$

2.4. Up-Sampling Module

The operation of inputting a low-resolution image and generating a high-resolution image is called the up-sampling operation. Transposed convolution is used for up-sampling in DeblurGAN. Although this algorithm generates better images than the traditional UnPooling up-sampling algorithm, the conventional Transposed convolution operation is likely to cause uneven overlap between pixels in two-dimensional images, and the most typical result is the occurrence of high checkerboard artifacts in recovered images. PixelShuffle [14] up-sampling algorithm adopted in this paper can reduce such artifacts to some extent. In this algorithm, the low-resolution feature image is obtained by convolution and multi-channel recombination. The operation process is shown in Fig. 5.

In the Fig. 5, the first three images from left to right are used for feature extraction of low-resolution images with input W×H to generate a feature map with channel number R^2 and size W×H. R is the amplification factor of up-sampling. In this paper, R equals 2. Then divide a low-resolution pixel into R^2.

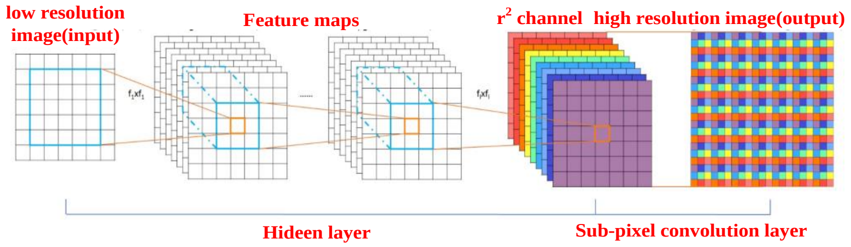
low resolution image(input)
Feature maps
r^2 channel
high resolution image(output)
Hideen layer
Sub-pixel convolution layer

Fig. 5 Operation procedure of PiexlShuffle

Small cells, and fill these small cells according to certain rules. Divide and fill each low-resolution pixel with this algorithm, which completes the process of pixel reorganization. Since the neural network is learnable, the weights of R^2 recombination channels can be adjusted during the recombination to further optimize the generated results. So as to produce higher quality high-resolution images with width W×R and height H×R.

2.5. Discriminator Structure

The discriminator outputs the probability value of the real clear image and the image generated by the generator to judge the authenticity of the image and optimize the generator according to the probability value of the output. ISOLA *et al*. [15]proposed the PatchGAN discriminator. The basic idea is to randomly cut the input image into multiple N×N small images (patches), and the result of each Patch X_{ij} represents the discrimination result of a certain area in the original image. The discriminator averages the results of each Patch to get the final output result. This training makes the model pay more attention to the details of the image, which helps the generator to produce better details and sharper edges. The specific structure of the discriminator in this paper is consists of five 4×4 convolution layers. The first layer is followed by the LeakyReLU activation function, the next three layers have been added with an Instance Norm layer and the LeakyReLU activation function. The last layer adds Sigmoid function activation as the final output.

2.6. Relative Discriminator Loss

Standard GAN still has many problems, such as gradient disappearance and mode collapse [17]. Usually, in the fight against network trained, the generator and discrimination network is a kind of interval training mode, when training generator, fixed his discriminant network parameter optimization, on the other hand, the discriminant unit training, fixed generator network parameter optimization, the interval training way makes it hard for traditional generated against network training, It is difficult to determine when the generator and discriminator reach dynamic equilibrium, making training very unstable.

An improvement on GAN is Relativistic GAN [18]. It uses a relativistic discriminator to calculate the probability that a given real data is more real than the fake data generated by the generator. In the Standard GAN, the function of the discriminator is to score the authenticity of the data and determine the probability that the input image is real, which will lead to the high score of the real data. According to GAN, while discriminators are trained to improve the probability of false data being true, they should also reduce the probability of true data being true, which can explain a prior knowledge that half of the images input by discriminators are false.

In this paper, a Relativistic Average Discriminator (RaD) proposed in relative GAN is used to replace WGAN-GP [16] in DeblurGAN, which makes model training more stable and efficient. The formula of relative average discriminator is shown as follows:

$$D_{Ra}(X_r, X_f) = \sigma(C(X_r) - E_{Xf}[C(X_f)]) \quad (5)$$

where X_r represents the real clear image, X_f represents the generated deblurring image, and σ represents the sigmoid function, E_{xf} denotes the operation to find the mean of all generated images of a batch. The result of the above formula represents the probability that a given real data is truer than a random sample of fake data.

According to the definition of relative average discriminator, the generator and discriminator loss function are modified. The loss function of the generator and discriminator after modification is given below, and the calculation formulas are shown in (6) and (7) respectively:

$$L_D^{Ra} = -E_{Xr}[\log(D_{Ra}(x_r, x_f))] - E_{Xf}[\log(1 - D_{Ra}(x_f, x_r))] \quad (6)$$

$$L_G^{Ra} = -E_{Xr}[\log(1-D_{Ra}(x_r, x_f))] - E_{Xf}[\log(D_{Ra}(x_f, x_r))] + \lambda L_{\mathrm{per}} \tag{7}$$

where L_D^{Ra} is the discriminator loss function, L_G^{Ra} is the generator loss function, where, X_r represents the real clear image, X_f represents the generated deblurring image, L_{per} represents the operation of finding the mean value of the generated image of a batch. Is the content loss function. In this experiment, λ is 100.

2.7. Content loss

The content loss in this paper abandons the traditional L_1 and L_2 loss. The L1 loss is the mean absolute error (MAE), which is the sum of the absolute values of the difference between the target value and the predicted value. The L_2 loss, or mean square error (MSE), is the sum of squares of the difference between the predicted value and the target value. However, because these two kinds of losses are averaged in pixel space, using these two kinds of losses as the only optimization will lead to a higher degree of blur in details of the image. This paper uses perceptual loss [19] as content loss. Make the restored image better in detail. Perceived loss is defined as the difference in feature graph between the image generated by generator and the real clear image after passing through VGG-19 network Conv3.3 layer. Perceived loss is defined as follows:

$$L_{per} = \frac{1}{W_{ij}H_{ij}} \sum_{x=1}^{W_{ij}} \sum_{y=1}^{W_{ij}} (\phi_{ij}(I_s)_{xy} - \phi_{ij}(G(I_s))_{xy})^2 \tag{8}$$

3. Experiment and result analysis

In order to verify the efficiency of the algorithm in this paper, this section discusses the experimental data set, experimental settings and result analysis, and compares the subjective visual level and objective evaluation indicators. The blurred image into the generator network, output to the blur image and calculate the generator losses, then the generated image and clear images of real incoming test apparatus, and calculate the discriminant loss and update for the generator and discriminant network optimization, the final training model to achieve a sense of stability, generation to blur the good image.

3.1. Experimental Data

The selection of training data set is the key factor to determine whether the model has good deblurring performance. The blur image of the early training set is obtained by convolution the blur kernel with the clear image. However, the blur image scene generated by this algorithm is too single, resulting in poor generalization ability of the model. Nah et al. [10] proposed a new image generation algorithm and then generated GOPRO data set. They used high-speed cameras to shoot videos and extracted connected short exposure frames for average, so as to obtain blur images. The algorithm can simulate complex camera shake and object motion, and produce images that are closer to the real scene. From this. The experiment in this paper adopts GOPRO training set for training, which consists of 2103 blur and clear image pairs with a resolution of 720p, and uses 1100 test sets to test images.

3.2. The Experimental Setup

The Pytorch open-source framework for deep learning was used in this experiment. The CPU was configured as Intel Core I7-7700HQ, and GPU is NVIDIA GeForce GTX 1080Ti, and the memory size was 12GB. The algorithm model is trained on Ubntu18.04 platform. The input data was GOPRO training set and randomly cropped to 256×256 images. Adam optimizer was used as the optimizer, and the exponential decay rate β1 of the first-order moment estimation of momentum parameters was set as 0.5, making the proportion of momentum and gradient the same. The exponential decay rate β2 of the second-order moment estimation was set as 0.999. A total of 500 epochs were trained, and batchsize set to 1. Because the learning rate setting is too large, it is difficult to find the optimal solution, and the setting is too small, which makes the training too slow. Therefore, in the experiment of this paper, the learning rate is set as 0.0001 for the first 150 times, after which the learning rate linearly decayed to 0. In order to reduce the over-fitting of the model, the Dropout [20] strategy is adopted in the training process, that is, during the training process, some neurons are randomly discarded according to a certain probability. The Dropout in this paper is set to 0.5, that is, half of neurons are randomly discarded for training.

3.3. GOPRO Data Set Comparison

For the task of blind image deblurring, this paper selects two blur images from GOPRO data set and uses the algorithm proposed in this paper and other algorithms to conduct image deblurring experiments. The comparison results of objective evaluation indexes of different algorithms are shown in Table I, and the subjective effects are shown in Fig. 6. Among them, Xu [4] and Kim et al [5]. based on the non-blind deblurring algorithm. Although such algorithms have certain effects, they have poor processing of image edges. CNN-Deblur [9] algorithm uses convolutional neural network to estimate the probability distribution of motion blur and estimate the blur kernel for deblurring. However, the algorithm cannot accurately estimate most of the blur kernel on the GOPRO data set, and the deblurring effect is poor. DeepDeblur [10] adopts end-to-end multi-scale algorithm. Although the blur removal effect is significantly better than the previous algorithms, details are relatively blur. Deblur-GAN [12] algorithm and GR-Deblurgan [21] algorithm are both based on the end-to-end Deblur of generative adversarial nets, which are superior to other algorithms in terms of overall and details. Compared with Deblur-GAN, the proposed algorithm is 0.92 higher in PSNR index, 0.011 higher in SSIM index, and better than other algorithms in image details processing, such as font display on the wall and license plate number. It also shows that the proposed deblurring model has obvious advantages over other algorithms in image deblurring task.

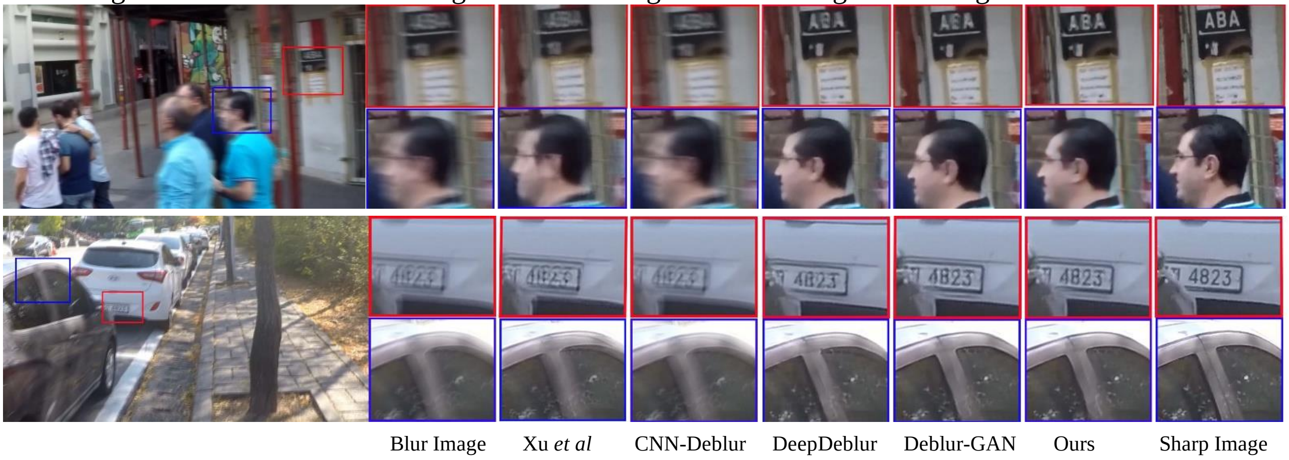

Fig. 6 Comparison of deblurring results on the GOPRO test set

TABLE I. Deblurring performance evaluation on gopro data set

	Xu et al	Kim et al	CNN-Deblur	Deep Deblur	Deblur-GAN	GR- DeblurGAN	Ours
PSNR	24.81	23.64	24.64	29.08	28.42	27.7	**29.34**
SSIM	0.884	0.823	0.842	0.913	0.917	0.923	**0.928**

3.4. Comparison on the Kohler Datasets

Kohler data set [22] consists of 48 images, including 4 scenes, and each scene is processed by 12 different convolution kernels, which is used to evaluate the benchmark data set of blind deblurring algorithm. The results of different algorithms are shown in Table II. As the blur kernel of Xu et al. and other algorithms is known, the PSNR is higher than that of other algorithms, followed by the algorithm in this paper, which is higher than Deblur-GAN 0.22. However, in terms of SSIM, the algorithm in this paper is the best, which is higher than Deblur-GAN 0.007. It can be seen that the algorithm in this paper has excellent overall performance in objective evaluation indexes. The subjective image is shown in Fig. 7. This paper selects two different scenes and compares them with DeepDeblur algorithm. The algorithm in this paper has a strong ability of blurring blurred images both in terms of the whole and details.

In summary, the recovery performance of the proposed algorithm is superior to other algorithms on Kohler datasets, which proves the effectiveness of the proposed algorithm.

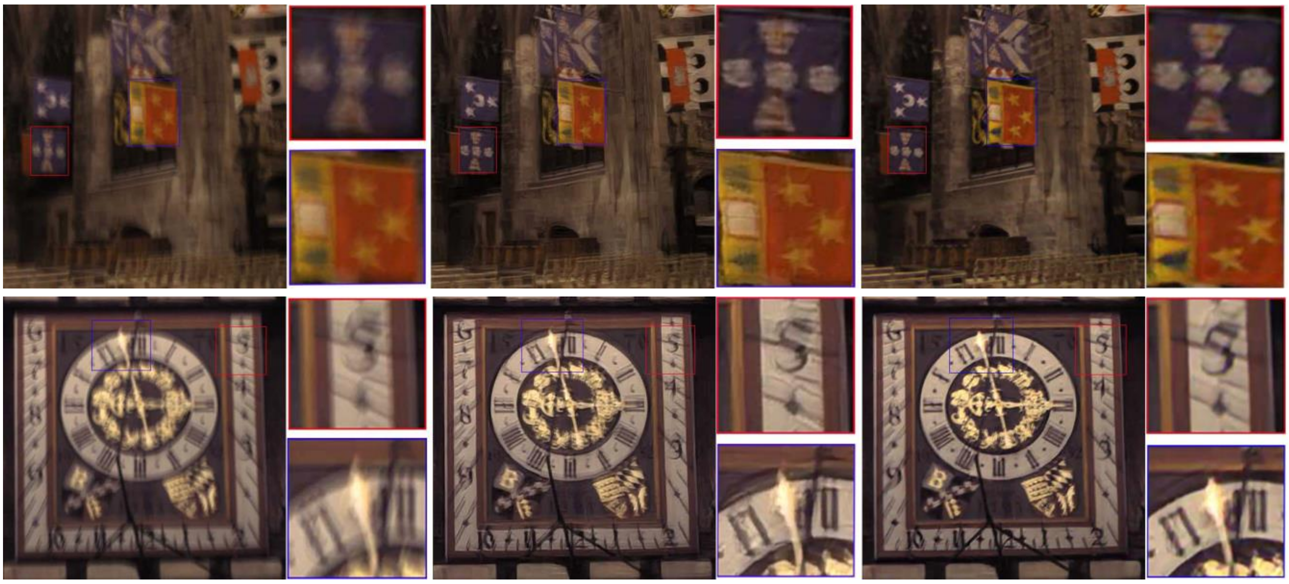

Fig. 7 Comparison of deblurring results on the Kohler test set

TABLE II. Deblurring performance evaluation on gopro data set

	Xu et al	CNN- Deblur	Deep Deblur	Deblur- GAN	Ours
PSNR	27.41	25.22	26.21	26.10	**26.32**
SSIM	0.750	0.773	0.807	0.815	**0.822**

3.5. Calculate Efficiency and Number of Parameters

In addition to objective indicators and subjective effects, the running time of each model applied to a single image and the number of model parameters are also important indicators to measure the computational efficiency of an algorithm model. The shorter the running time, the smaller the number of parameters, the smaller the algorithm complexity of the model. In this paper, the above algorithms are selected to test the running time of each model for a single image on GOPRO data set and Kohler data set respectively. All the tests are based on GPU-accelerated test on the same platform. The results are shown in Table III.

TABLE III. Running time and parameter number of different algorithms on two datasets

	Xu et al	CNN- Deblur	Deep Deblur	Deblur- GAN	Ours
GOPRO times(S)	13.41	1200	4.3	0.91	**0.48**
Kohler times(S)	12.59	1200	4.3	0.62	**0.35**
Parameter (MB)	37	52	303	23.2	**11.9**

It can be seen from Table III that Xu *et al* and other algorithms based on non-blind blur removal do not need to estimate the blur kernel, and the time and space consumption are small. The CNN-Deblur algorithm proposed by Sun and other algorithms need to estimate the blur probability distribution to estimate the blur kernel, and then realize deconvolution through deconvolution. Although the space consumption is not large, the time consumption of a single image is too large which cannot meet the practical application. The time consumption of DeepDeblur algorithm is greatly reduced due to the end-to-end approach. However, as the model adopts a coarse-to-fine multi-scale model, the number of parameters is large. Deblur-GAN algorithm has a significant improvement in running time and parameter number compared with previous algorithms. In this paper, depth-separable convolution is used to replace traditional convolution and attention module is added, so the algorithm in this paper is only half of Deblur-GAN in both running time and model parameters. It fully shows that the algorithm in this paper has short running time, small number of parameters and small algorithm complexity.

4. Conclusion

To solve the problem of image blur, this paper proposes an image deblur algorithm based on depth separable convolution lightweight generative adversarial network. It makes the network training more stable and ensures that the model is more lightweight, while generating clearer details and more realistic pictures overall. Experimental results prove that the algorithm presented in this paper has significant deblurring effect in different data sets and consumes less time and space.

However, the algorithm is not good for large size blur kernel processing, and the generalization ability and recovery details of the algorithm need to be improved, which is also the focus of future work.

5. References

[1] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman. "Non-local sparse models for image restoration." *IEEE I nternational Conference On Computer Vision*, pp. 2272-2279, 2009.

[2] A. Kour, V. K. Yada, V. Maheshwari. "A review on image processing." *International Journal of Electronics Com munication and Computer Engineering*, vol. 4, pp. 270-275, 2013.

[3] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms." *Physica D: nonline ar phenomena*, vol. 60, pp. 1-4, 1992.

[4] L. Xu, S. Zheng, and J. Jia, "Unnatural L0 Sparse Representation for Natural Image Deblurring." *Proceedings of t he IEEE Conference On Computer Vision And Pattern Recognition*, 2013.

[5] T. H. Kim, K. M. Lee, "Segmentation-Free Dynamic Scene Deblurring." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 2766-2773, 2014.

[6] J. Pan, Z. Hu, Z. Su and M. Yang, Deblurring Text Images via "L0-Regularized Intensity and Gradient Prior." *Pro ceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 2901-2908, 2014.

[7] L. Li, J. Pan, W. Lai, C. Gao, N. Sang and M. Yang, "Learning a Discriminative Prior for Blind Image Deblurrig." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 6616-6625, 2018.

[8] T. Michaeli, M. Irani, "Blind Deblurring Using Internal Patch Recurrence." *In European conference on computer vision*, vol. 8691, pp. 783-798, 2014.

[9] J. Sun, Wenfei Cao, Zongben Xu and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 769-777, 2 015.

[10] S. Nah, T. H. Kim and K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurrin g." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 257-265, 2017.

[11] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, et al, "Generative adversarial nets." *Commun. ACM*, pp. 267 2-2681, June 2014.

[12] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin and J. Matas, "DeblurGAN: Blind Motion Deblurring Using C onditional Adversarial Networks." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recogni tion*, pp. 8183-8192, 2018.

[13] A. G. Howard, M. L. Zhu, B. Chen, D. Kalenichenko, et al, "MobileNets: Efficient Convolutional Neural Network s for Mobile Vision Applications." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recogn ition*, April 2017.

[14] W. Shi, J. Caballero1, F. Huszar, J. Totz. et al, "Real-Time Single Image and Video Super-Resolution Using an Ef ficient Sub-Pixel Convolutional Neural Network." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 1874-1883, 2016.

[15] P. Isola, J. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks." *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition*, pp. 5967-5976, 2017.

[16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. "Improved training of wasserstein gans." *Ad vances in neural information processing systems*, pp. 5767–5777, 2017.

[17] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Rad-ford, and X. Chen, "Improved Techniques for Traini ng GANs," *Advances in neural information processing systems*, pp. 2234–2242, December 2016.

[18] A. Jolicoeur-Martineau, “The relativistic discriminator:a key element missing from standard gan,” *arXiv preprint a rXiv:1807.00734*, 2018.

[19] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken and A. Tejani, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network,” *Proceedings of the IEEE Conference O n Computer Vision And Pattern Recognition*, pp. 105-114, 2017.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, L. Sutskever and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *The journal of machine learning research*, Vol. 15, pp. 1929–1958, January 2 014.

[21] Q. Chen, X. Sui, Y. Guan, J. Fan, X. Deng and J. Wang, “Based on the multi-scale residuals generated against the single image blind to motion blur method of the network,” *Computer application research*, vol. 38, pp. 919-922, 2 021.

[22] R. Kohler, M. Hirsch, B. Mohler, B. Schölkopf and S. Harmeling, “Recording and playback of camera shake: Ben chmarking blind deconvolution with a real-world database,” *European conference on computer vision*. Berlin, 201 2.