Research Article

Motion Deblurring in Image Color Enhancement by WGAN

Jiangfan Feng and Shuang Qi

Chongqing University of Posts and Telecommunications, College of Computer Science and Technology, Chongqing, China

Correspondence should be addressed to Jiangfan Feng; fengjf@cqupt.edu.cn

Received 13 November 2019; Revised 8 May 2020; Accepted 28 May 2020; Published 24 June 2020

Academic Editor: Adrian Podoleanu

Copyright © 2020 Jiangfan Feng and Shuang Qi. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Motion deblurring and image enhancement are active research areas over the years. Although the CNN-based model has an advanced state of the art in motion deblurring and image enhancement, it fails to produce multitask results when challenged with the images of challenging illumination conditions. The key idea of this paper is to introduce a novel multitask learning algorithm for image motion deblurring and color enhancement, which enables us to enhance the color effect of an image while eliminating motion blur. To achieve this, we explore the synchronization of processing two tasks for the first time by using the framework of generative adversarial networks (GANs). We add $L1$ loss to the generator loss to simulate the model to match the target image at the pixel level. To make the generated image closer to the target image at the visual level, we also integrate perceptual style loss into generator loss. After a lot of experiments, we get an effective configuration scheme. The best model trained for about one week has achieved state-of-the-art performance in both deblurring and enhancement. Also, its image processing speed is approximately 1.75 times faster than the best competitor.

1. Introduction

With the rapid developments in sensor technology and mobile devices, we have entered an era of consumer photography. As cheap devices are portable, camera shake is unavoidable and may result in blurry light field images. Similarly, moving objects can also cause the images to appear blurry. This has led to an increased interest in the image enhancement and motion deblurring from light fields.

Due to the excellent performance of convolutional neural networks (CNNs) in dealing with computer vision tasks, they have turned out to be good at motion deblurring and image enhancement [1–9]. These approaches work well for tasks in which the visual data provide sufficient information, and data collection can be automated effectively. There are two critical challenges in learning policies to synchronize the two tasks. First, motion information is difficult to automate for large-scale data collection. Second, the mentioned CNN-based methods, all dedicated to single task, suffer from the difficulties of making a multieffect dataset for the high cost of hiring professional photographers to retouch the target images. Therefore, it is desirable to develop more effective methods applied to synchronize the two tasks.

In this work, we propose a learning approach to acquiring tasks jointly. In particular, we propose a GAN-based model to perform sharp enhanced imaging from motion-blurred images by learning nonlinear input-output mappings. The aim is to train two dual tasks jointly: motion deblurring and image enhancement, via double learning, which improves the supervision and the generalization performance of both tasks. Firstly, motivated by the method proposed by Chen et al. [9], we construct a new dataset for multitask training. Then, we train the model based on WGAN-GP [10] framework as its stability characteristic. We integrate $L1$ loss and perceptual style loss as pixel loss into generator loss to make a generated image match the target image at both pixel and visual level. After about one week of training, our model has achieved state-of-the-art test results. Further, the processing performance is approximately 1.75 times faster than the best competitor.

2. Related Works

The transformation of one image into another is defined as an image-to-image translation task by Isola et al. [11] with the basic set of predicting pixels from pixels. In this

direction, CNNs are widely used to tackle these problems by seeking a loss function to narrow the gap between the predicted values and the real values. Figure 1 shows a typical CNN structure for image translation tasks.

As shown in Figure 1, convolutional neural networks with this structure are often treated as encoder-decoder networks. Encoder-decoder networks are active image transformation structures, which are used in many visual tasks [12, 13]. It is an asymmetrical structure. The input image enters the encoder computed by several convolutional layers gradually. The output of each layer is half the size and dual channels of the previous.

To benefit gradient propagation and accelerate convergence, skip connections between corresponding layers are conventional means among the proposed methods. A loss function commonly used in this structure is the Euclidean distance. According to [14, 15], it is difficult to get the desired result by using a simple model structure with Euclidean distance. Tao et al. [3] built the SRN-DeblurNet by using this traditional structure. They adopted a multiscale strategy [2] and added ConvLSTM [16] in their model. This undoubtedly increases their training time and processing time. The method based on pure CNN structure requires a lot of effort to explore the appropriate loss function and complex networks to get what we expect. Fortunately, Goodfellow et al. [17] proposed generative adversarial networks (GANs) which enable the method based on this to effectively implement image translation tasks without complex network structure and loss function. Figure 2 shows the structure of GANs.

As shown in Figure 2, GANs include two parts: generator and discriminator. The goal of the generator is to turn noise into an image close to the target image. The purpose of the discriminator is to distinguish the generated image from the target image. During the training of GANs, both of them compete with each other and intensify at the same time, which ultimately makes the generator generate indistinguishable images.

Original GANs turn the noise into the target image. To solve the translation task of the image, Isola et al. [11] proposed conditional GANs (cGANs) based on the GAN framework, which maps specific input to the target image. Ramakrishnan et al. [4] proposed a model for motion deblurring by using this model structure. Because their generator contains dozens of convolutional layers, it takes them about 0.3 seconds to process one image with fixed size 256×256 in their test environment. Based on conditional GANs, Yu et al. [18] proposed a model for fast reconstruction of MRI images from highly undersampled data by integrating the perceptual content loss into adversarial loss and adding a refinement learning procedure to the generator. Yang et al. [19] proposed a better model named DAGAN for CS-MRI image reconstruction based on their previous work [18] with adding frequency domain mean square error loss.

However, as described by Salimans et al. [20], gradient disappearance and gradient explosion are easy to occur in the training process of the model based on the original GAN

structure. In GANs, optimizing the loss function is equal to minimizing the Jensen–Shannon divergence of the generated data distribution and the target data distribution, which was analyzed by Arjovsky et al. [21]. Because there is no overlap between the two distributions, the Jensen–Shannon divergence of them is constant. Finally, when the discriminator is in the optimal state, the generator is challenging to be trained as we want. They proposed to use Wasserstein distance to improve the model, which is known as WGAN. The weight clipping of the discriminator is one of the core skills of WGAN to simulate Wasserstein distance. In the latter study, Gulrajani et al. [10] proposed a new training skill of WGAN named gradient penalty because of the two-stage differentiation in the parameter of the discriminator and the difficulties of adjusting the clipping range. Kupyn et al. [22] proposed a model named DeblurGAN by using the framework of WGAN-GP [10] to realize deblurring. But the deconvolutional layers make them encounter the checkerboard artifacts [23]. Chen et al. [9] proposed a deep enhancer to enhance images by taking advantage of WGAN-GP and CycleGAN [24]. Their model has an excellent performance in image enhancement. Wang et al. [25] proposed a novel GAN-based framework which makes two GANs concatenated, sharing partial parameters of the discriminator and generator between the two GANs for salient object detection.

3. Proposed Method

3.1. The Overall Framework of Our Model. To achieve the simultaneous processing of two tasks, we design a model based on the WGAN-GP structure, which has the characteristics of fast convergence and stable training. As can be seen from Figure 3, our model consists of two parts: generator and discriminator. Firstly, we input the blurred image into generators to generate the fake image and then calculate the pixel loss, which is the combination of the content loss and style loss between the fake image and the real image. Secondly, we input the fake image and the actual image into discriminator to calculate the Wasserstein distance. Wasserstein distance can simulate discriminator to optimize its parameters. Finally, we use the adversarial loss and pixel loss to boost the generator and discriminator simultaneously, which makes the generator confuse discriminator when optimizing. After these three steps of finite iteration training, the generator can make the blurred image sharp and bright.

3.2. The Generator. The goal of the generator is to generate fake images that cannot be distinguished from the real. We use images with two characteristics of sharp and bright as the learning objects of the generator. As shown in Figure 4, our generator consists of two parts: encoder and decoder. The encoder is responsible for extracting image features, while the decoder accounts for reconstructing and optimizing the image according to the features extracted by the encoder.

3.2.1. Encoder and Decoder. As mentioned above, our encoder is designed for extracting features. Its core component

FIGURE 1: The structure of CNNs for image translation tasks.

FIGURE 2: The structure of GANs.

FIGURE 3: The structure of our model.

is DStairBlock. Each DStairBlock halves the size of feature maps and doubles the channels. This design allows DStairBlock's filters to grow and learn more features. However, there are leading training problems of deep learning networks regarding the disappearance and explosion of the gradient. The corresponding traditional solutions are the initialization and regularization of data. Although these solutions solve the gradient problem and deepen the depth, it brings another problem, that is, the degradation of

network performance. While the residual net [26] is used for solving this problem, it also eases the gradient problem of deep networks. Therefore, we apply the idea of the residual net to our DStairBlock, which helps enhance the encoder's sensitivity of extracting features. As shown in Figure 5, we also add Relu [27] as a nonlinear activation function in DStairBlock to increase the stability of the model. In our model, after the input image passed through the encoder layers, the encoder extracted 128 features. Figure 6 shows 10

FIGURE 4: The structure of the generator.

FIGURE 5: The structure of DStairBlock.

of them randomly selected by us. It can be seen from the figure that encoder extracts the contour features of low exposure areas and sharpens the object edge.

The decoder is responsible for image reconstruction using the features extracted by the encoder. Its structure is symmetrical with the encoder. Referring to [23], we use the combination of resizing and convolution layer instead of deconvolution, which makes the images smoother. As can be seen from Figure 4, the reconstructed feature maps processed by each UStairBlock are becoming more evident and approach the target image.

3.2.2. Loss of Generator. As mentioned earlier, to quantify the proximity between generated image and target image and guide the generator to optimize, the proper loss is also a key factor for the success of the model. There is usually a choice between $L1$ (MAE) loss and $L2$ (MSE) loss as the content loss for a model to be driven. Recently, some methods combine the classical losses with the perceptual content loss [28] as the content loss. The perceptual content loss is $L2$ loss, but the calculated objects are the feature maps of the specific convolutional layer of the VGG model [29]. In the study of style transfer, Gatys et al. [30] proposed style loss, which can show the style gap between images. The key point of style loss is to calculate the correlation of each feature in the feature maps extracted by the VGG model. This feature correlation is called the Gram matrix. Formula

(1) shows the calculation of the Gram matrix, where $V_n(x)_{h,w,ci}$ and $V_n(x)_{h,w,cj}$ denote the i -th and j -th features extracted by the n -th layer of the VGG model. The feature correlation can reflect the style of the picture in detail. Formula (2) shows the calculation of the style loss. In short, it is the squared Frobenius norm of the difference of two Gram matrices. At a certain level, style loss can drive the model to shorten the structural similarity and color similarity between the generated image and the target image. Inspired by the study of style transfer, the goal of our model is to learn the sharp and colorful style of target images. Ultimately, we use the combination of $L1$ loss and style loss as the pixel loss to simulate model training. Formulas (3) and (4) show the calculation of $L1$ and L_{pixel} , respectively. λ and γ in formula (4) represent the proportion of content loss and style loss in the pixel loss, respectively. To make the discriminator and generator fight against each other in training, adversarial loss is an essential element. Mathematically, the adversarial loss is defined as formula (5), where P_g denotes the distribution of the generated data and $D\theta()$ and $G\theta()$ denote the discriminator and generator, respectively. Formula (6) shows the loss of our generator. This loss can simulate the discriminator and generator against each other and shorten the pixel distance of the image in model training. The next section will cover the implementation details of these losses.

FIGURE 6: The extracted feature maps of the encoder.

$$G_n^V(x)_{c_i c_j} = \frac{1}{c_n h_n w_n} \sum_{h=1}^h \sum_{w=1}^w V_n(x)_{h,w,c_i} V_n(x)_{h,w,c_j}, \quad (1)$$

$$L_{\text{style}}(Y, X) = \|G_n^V(Y) - G_n^V(X)\|_F^2, \quad (2)$$

$$L_1(Y, X) = \frac{1}{\text{whc}} \sum_{h=1}^h \sum_{j=1}^j \sum_{k=1}^c |Y_{ijk} - X_{ijk}|, \quad (3)$$

$$L_{\text{pixel}}(Y, X) = \lambda L_1(Y, X) + \gamma L_{\text{style}}(Y, X), \quad (4)$$

$$L_{\text{adv}} = -E_{\tilde{x} \sim P_g} [D_{\theta}(\tilde{x})], \quad (5)$$

$$\tilde{x} = G_{\theta}(X),$$

$$L_g(Y, X) = L_{\text{adv}}(X) + L_{\text{pixel}}(\tilde{Y}, \tilde{X}), \quad (6)$$

$$\tilde{x} = G_{\theta}(X).$$

3.3. The Discriminator. In the vanilla GAN framework, the discriminator is responsible for the simpler binary classification task. The stronger the ability of the discriminator to distinguish real from fake, the closer the generated image is to the target image. But it backfired. Based on WGAN-GP, we use the discriminator of PathGAN [31] to help us enhance our generator. It only consists of five convolutional layers and is conducive to reducing model parameters and shortening training time.

3.3.1. Loss of Discriminator. Wasserstein distance is the key point of WGAN realized by weight clipping. But there are two serious problems, as mentioned above. The gradient penalty is the improvement measure. The loss of the discriminator can be defined as formula (7), where P_t denotes the distribution of the target data, K indicates the range that the weight of discriminator needs to be limited, and β denotes the coefficient of gradient penalty. During our model training, we set K and β to 1 and 10, respectively. χ indicates the distribution of the data generated by the target data and the generated data. Formula (8) shows how $\hat{x}$ is calculated.

$$L_D(Y, X) = E_{\tilde{x} \sim P_g} [D_{\theta}(\tilde{x})] - E_{Y \sim P_t} [D_{\theta}(Y)] + \beta E_{\tilde{x} \sim \chi} \left[\left\| \nabla_{\tilde{x}} D_{\theta}(\tilde{x}) \right\|_p - K \right]^2, \quad (7)$$

$$\hat{x} = \varepsilon X_t + (1 - \varepsilon) x_g, x_t \sim P_t, x_g \sim P_g, \varepsilon \sim \text{Uniform}[0, 1]. \quad (8)$$

4. Experiment and Results

4.1. Dataset. Traditional deblurring methods use uniform or nonuniform blur kernels to convolute sharp images to generate blurry images. However, the blurred images generated by these methods are still different from the images captured by cameras for the simplified blurring kernels. Therefore, the model trained with these datasets often fails to recover sharp images in the real world. To solve this problem, Nah et al. [2] proposed a method to simulate the generation of authentic blurred images and constructs the GoPRO dataset, which

consists of 2013 pairs of training pictures and 1111 pairs of test pictures. This method synthesizes the continuous frames of videos captured by high-speed cameras in exposure time into blurred images. As for image enhancement, MIT-Adobe 5K [32] is a common dataset. It consists of 5,000 pictures, which is adjusted by five professional photographers with an excellent visual appearance. The image enhancement method of [9] trains their model by using this dataset. Their outstanding performance wins public recognition. We reconstruct our dataset based on GoPRO by using their method to get blurry and sharp-enhanced image pairs.

We also use Kohler dataset [33], a benchmark dataset for evaluating motion deblurring algorithms, to assess our model. This dataset contains 48 blurred images, which are processed by 12 different blur kernels.

4.2. Training Details. We train our model on a server equipped with Intel Core i7-6850K CPU and NVIDIA GTX 1080ti GPU and test on a PC equipped with Core i7-7700HQ CPU and NVIDIA GTX 1050 GPU. We use ADAM [28] optimizer and set the learning rate to a fixed value of 0.0001 for the discriminator and adapt an exponential decay strategy for the learning rate of the generator. In our experiment, the batch size is set to 1, and the input size of the image is 256×256 . After about one week of training, we got satisfactory results.

4.3. Implementation Details. As mentioned earlier, the content loss is used as the loss function in most images to image tasks. To find the most suitable loss function for ours, we first trained our model by setting four different kinds of pixel losses defined as formulas (9) to (12).

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X), \quad (9)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_2(Y, X), \quad (10)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 100 \times \text{Per}(Y, X), \quad (11)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_2(Y, X) + 100 \times \text{Per}(Y, X), \quad (12)$$

where $\text{Per}()$ denotes the perceptual content loss calculated with the feature maps extracted from the third convolution block of VGG-16. In the following description, we use model 1, model 2, model 3, and model 4 to represent the model trained with the pixel loss as listed above. To evaluate the performance of different models, we adopt two standard objective metrics in terms of image quality: PSNR (peak signal to noise ratio) and SSIM (structural similarity index). PSNR reflects the distortion of the image after processing. The larger the PSNR value, the smaller the image distortion. SSIM demonstrates the integrity of image structure information. The larger the SSIM value is, the higher degree of the similarity of structures between images is. Table 1 shows the test results of these models. Except for model 1, the other three models are all trained without gradient penalty because without using the gradient penalty, model 1 gets the results of 30.40 and 0.957, which are also better than the other three models. The images generated by these four models are shown in Figure 7. The images generated by model 1 are sharper and smoother than the others. The results of model 2 have more blurry artifacts but without noise. The images generated by the remaining two models have some noise and are blurrier.

After obtaining the above test results, we add style loss to pixel loss inspired by the study of style transfer. In our experiments, VGG-16 is the perceptual information extracting model. It has 13 convolution layers in five convolution blocks, as shown in Figure 8. We capture the features generated by

TABLE 1: Quantitative deblurring and enhancement performance of different models (1 to 4) on GoPRO.

Model	PSNR	SSIM
Model 1	30.49	0.960
Model 2	30.10	0.955
Model 3	29.57	0.947
Model 4	29.31	0.926

these five blocks. Figure 9 shows the first five features extracted from these five blocks. In the figure, the features of blocks 1 to 5 are shown from top to bottom. It can be seen from the figure that with the increase of the convolution layer, the features become more and more abstract. In terms of the model, the deeper the convolution layer is, the larger the receptive field the layer has. Therefore, a deep convolution layer can extract high-level nonlinear features.

We add the style loss of these five blocks to the pixel loss, respectively. In the following description, we use model 5, model 6, model 7, model 8, and model 9 to represent the model trained with the five-pixel losses shown by formulas (13) to (17).

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_1}(Y, X), \quad (13)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_2}(Y, X), \quad (14)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_3}(Y, X), \quad (15)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_4}(Y, X), \quad (16)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_5}(Y, X). \quad (17)$$

Figure 10 shows the test results of these five models on the GoPRO dataset. As can be seen from the figure, the visual effect of mode l5 and mode l9 is the worst, and the restored images of mode l9 are severely distorted and have point noise because the feature of block 5 is too abstract and disturbs the content loss. The reconstructed images of models 6 to 8 are better but not as good as those of model 1. Table 2 shows their standard test results on the GoPRO dataset. From the table, we can also see the reality reflected by the pictures. The test results of model 7 are the best of the five models, followed by model 8. Therefore, the style feature extracted by the middle layer is more valuable. However, the perceptual information obtained from a single block is too one-sided to show the effect of style loss.

Based on the above results, we add the style loss generated by multiple blocks to the pixel loss. In the following description, we use model 10, model 11, and model 12 to represent the model trained with the three pixel losses shown by the formulas (18) to (20).

FIGURE 7: The test results of models 1 to 4 on GoPRO. (a) Blurred image. (b) Generated image of model 1. (c) Generated image of model 2. (d) Generated image of model 3. (e) Generated image of model 4. (f) Ground truth.

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_1+B_2+B_3}(Y, X), \quad (18)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_2+B_3+B_4}(Y, X), \quad (19)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_3+B_4+B_5}(Y, X). \quad (20)$$

Figure 11 shows the test results of these three models on the GoPRO dataset. Table 3 shows the standard test results of these three models on the GoPRO dataset. From these two results, model 11 has the best performance. It can be seen from the figure that the overall visual effect of the image generated by model 11 is sharper and smoother, followed by model 12. It indicates that the combination style losses of deep blocks are better than shallow blocks because the deep blocks can extract more perceptual features. However, depending on the deep style loss completely can reduce the

FIGURE 8: The convolution structure of VGG-16.

FIGURE 9: The features extracted by different blocks of VGG-16.

FIGURE 10: The test results of model 1 and models 5 to 9 on GoPRO. (a) Generated image of model 1. (b) Generated image of model 5. (c) Generated image of model 6. (d) Generated image of model 7. (e) Generated image of model 8. (f) Generated image of model 9.

effect of the model for the high-level features that are overhanded. As a result, the best performance of model 11 is due to a combination of deep and shallow style losses.

The proportion of style loss and content loss can also seriously affect the results. To find the appropriate portion, we train four models with different values of λ and γ . In the following description, we use model 13, model 14, model 15, and model 16 to represent the model trained with the four pairs of values shown by formulas (21) to (24).

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 100 \times L_{\text{style}}^{B_2+B_3+B_4}(Y, X), \quad (21)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 1000 \times L_{\text{style}}^{B_2+B_3+B_4}(Y, X), \quad (22)$$

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 10000 \times L_{\text{style}}^{B_2+B_3+B_4}(Y, X), \quad (23)$$

TABLE 2: Quantitative deblurring and enhancement performance of models 5 to 9 on GoPRO.

Model	PSNR	SSIM
Model 5	29.14	0.925
Model 6	29.54	0.934
Model 7	30.04	0.952
Model 8	29.87	0.939
Model 9	29.21	0.916

FIGURE 11: The test results of models 10 to 12 on GoPRO. (a) Generated image of model 10. (b) Generated image of model 11. (c) Generated image of model 12.

TABLE 3: Quantitative deblurring and enhancement performance of models 10 to 12 on GoPRO.

Model	PSNR	SSIM
Model 10	30.12	0.953
Model 11	30.96	0.971
Model 12	30.54	0.959

FIGURE 12: The test results of models 13 to 16 on GoPRO. (a) Generated image of model 13. (b) Generated image of model 14. (c) Generated image of model 15. (d) Generated image of model 16.

TABLE 4: Quantitative deblurring and enhancement performance of models 13 to 16 on GoPRO.

Model	PSNR	SSIM
Model 13	30.17	0.943
Model 14	30.59	0.956
Model 15	30.96	0.971
Model 16	29.21	0.924

FIGURE 13: The test results of model 15 and models 17 to 19 on GoPRO. (a) Blurred image. (b) Generated image of model 17. (c) Generated image of model 18. (d) Generated image of model 19. (e) Generated image of ours. (f) Ground truth.

FIGURE 14: The test results of model 15 and models 17 to 19 on Kohler. (a) Blurred image. (b) Generated image of model 17. (c) Generated image of model 18. (d) Generated image of model 19. (e) Generated image of ours. (f) Ground truth.

$$L_{\text{pixel}}(Y, X) = 100 \times L_1(Y, X) + 100000 \times L_{\text{style}}^{B_2+B_3+B_4}(Y, X). \quad (24)$$

Figure 12 shows the test results of these four models on GoPRO dataset. Table 4 shows the standard test results of these four models on GoPRO dataset. Due to the small proportion of style loss in model 13, its performance is the worst. A small portion of style loss cannot improve the effectiveness of the model but can interfere with the content loss. However, if there is too much style of loss, the image generated by the model will lose details and be seriously

distorted. From the results, setting 100 and 10000 to the two parameters is a reasonable choice.

4.4. Comparison with the State-of-the-Art Models. Since there is no method similar to ours to recover a sharp enhanced image from the blurred by one model, we combine the three methods of DeblurGAN [22], Deep-DeblurFilter [4], and SRN-DeblurNet [3] with the Deep Enhancer [9], respectively, as the comparable methods, and we refer to them as model 17, model 18, and model 19 in the following description. All these models have recently achieved state-

TABLE 5: Quantitative deblurring and enhancement performance of model 15 and models 17 to 19 on GoPRO.

Model	Time (s)	PSNR	SSIM	MS-SSIM
Model 17	0.129	29.22	0.932	0.943
Model 18	0.590	29.46	0.931	0.946
Model 19	0.183	30.30	0.962	0.975
Ours	0.108	30.96	0.971	0.980

TABLE 6: Quantitative deblurring and enhancement performance of model 15 and models 17 to 19 on Kohler.

Model	PSNR	SSIM	MS-SSIM
Model 17	28.30	0.734	0.618
Model 18	28.59	0.740	0.637
Model 19	28.73	0.726	0.629
Ours	28.98	0.757	0.656

of-the-art results in their related fields. In our comparison experiment, we process the images with fixed size 256×256 . In the first step, we use the three models to get sharp images. Then, we put them into the Deep Enhancer to get sharp and enhanced images. Our best model is model 15. Figures 13 and 14 show some test results on GoPRO and Kohler. According to Figure 13, the image restored by model 17 has obvious checkerboard artifacts. The images restored by model 18 have clear contour but serious hallucination artefacts (the head of the little girl in the second picture). In fact, we also encounter this problem in model 1 without perceptual style loss (as shown in Figure 7). In general, the images generated by our model and model 19 are closer to the target images and have fewer hallucination artefacts. These two models have their own advantages in restoring different details. The overall visual effect of the images generated by our model is better. Therefore, we argue that the perceptual style loss promotes the model to approach the target image on the whole level in terms of structure and color. Although it makes the image lose some details, it makes the image look more comfortable for a smoother and colorful appearance. It also alleviates the hallucination problem caused by $L1$ loss. Tables 5 and 6 show the standard test results on the two datasets. The time is image passing through the generator in each method required. According to Table 5, our test results of PSNR and SSIM are better than the competitors. As shown by Ledig et al. [34], the image generated by the super-resolution model having the highest PSNR does not necessarily have the corresponding perceptual performance. In addition, Wang et al. [35] point out that the single scale is one drawback of SSIM. Therefore, we introduce MS-SSIM as our third metric measurement to evaluate the performance of the previous methods and ours. According to the two tables, our model gets the highest scores of MS-SSIM. What is more obvious is that model 19 takes about 69.4% more time than ours. The time for Deep Enhancer to process the image alone is about 0.027s.

5. Conclusion

In this paper, we propose a model that can enhance the sharpness and color of motion-blurred images at the same

time. It dramatically reduces the time of image processing. The training of the model is based on the WGAN-GP framework. In the generator, we use the residual connection to build three stages of encoder-decoder networks. This structure prevents gradient vanishing and makes transmission of image feature information effective. With the use of the open method, we create a new dataset for our model based on GoPRO. Besides, inspired by the study of style transfer, we integrate perceptual style loss into pixel loss, and the experiments prove that this is an effective measure to improve the visual effect of the generated image.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This project was sponsored by the National Natural Science Foundation of China (41971365 and 41571401) and Natural Science Foundation of Chongqing, China (cstc2019jcyjmsxmX0131).

References

- [1] L. Li, J. Pan, W. S. Lai, C. Gao, N. Sang, and M. H. Yang, "Learning a discriminative prior for blind image deblurring," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.
- [2] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.
- [3] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.
- [4] S. Ramakrishnan, S. Pachori, A. Gangopadhyay, and S. Raman, "Deep generative filter for motion deblurring," in *Proceedings of the 2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, Venice, Italy, October 2017.
- [5] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu, "Automatic photo adjustment using deep neural networks," *ACM Transactions on Graphics*, vol. 35, no. 2, pp. 1-15, 2016.
- [6] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1-12, 2017.
- [7] A. Ignatov, N. Kobyshev, R. Timofte, K. Vanhoey, and L. Van Gool, "DSLR-quality photos on mobile devices with deep convolutional networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

- [8] Q. Chen, J. Xu, and V. Koltun, "Fast image processing with fully-convolutional networks," in *Proceedings of the 2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, Manchester, UK, November 2019.
- [9] Y. S. Chen, Y. C. Wang, M. H. Kao, and Y. Y. Chuang, "Deep photo enhancer: unpaired learning for image enhancement from photographs with gans," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2016.
- [11] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, Coimbatore, India, June 2017.
- [12] X. Mao, C. Shen, and Y. B. Yang, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2016.
- [13] N. Xu, B. Price, S. Cohen, and T. Huang, "Deep image matting," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea, October 2019.
- [14] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: feature learning by inpainting," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.
- [15] R. Zhang, P. Isola, and A. A. Efros, "Colorful image colorization," *Computer Vision-ECCV 2016*, Springer, Berlin, Germany, pp. 649–666, 2016.
- [16] S. H. I. Xingjian, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: a machine learning approach for precipitation nowcasting," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2016.
- [17] I. Goodfellow, J. Pouget-Abadie, M. Mirza et al., "Generative adversarial nets," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2016.
- [18] S. Yu, H. Dong, G. Yang et al., *Deep de-aliasing for fast compressive sensing mri*, 2017, <http://arxiv.org/abs/1705.07137>.
- [19] G. Yang, S. Yu, H. Dong et al., "Deep de-aliasing generative adversarial networks for fast compressed sensing MRI reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1310–1321, 2017.
- [20] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, December 2016.
- [21] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017, <http://arxiv.org/abs/1701.07875>.
- [22] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [23] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, vol. 1, no. 10, p. e3, 2016.
- [24] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.
- [25] C. Wang, S. Dong, X. Zhao et al., "Deep learning semi-supervised salient object detection in the fog of iot," in *Proceedings of the IEEE Transactions on Industrial Informatics*, Piscataway, NJ, USA, October 2019.
- [26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, June 2018.
- [27] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *Proceedings of ICML*, vol. 30, no. 1, p. 3, 2013.
- [28] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *Computer Vision-ECCV 2016*, Springer, Berlin, Germany, pp. 694–711, 2016.
- [29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale Image recognition," 2014, <http://arxiv.org/abs/1409.1556>.
- [30] L. A. Gatys, A. S. Ecker, and M. Bethge, *A neural algorithm of artistic style*, 2015, <http://arxiv.org/abs/1508.06576>.
- [31] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Amsterdam, Netherlands, October 2016.
- [32] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, June 2011.
- [33] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling, "Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database," *Computer Vision-ECCV 2016*, Springer, Berlin, Germany, pp. 27–40, 2012.
- [34] C. Ledig, L. Theis, F. Huszár et al., "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, June 2017.
- [35] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proceedings of the Thirty-Seventh Asilomar Conference on Signals, Systems & Computers*, Pacific Grove, CA, USA, November 2003.