Hindawi
Scientific Programming
Volume 2022, Article ID 5815693, 11 pages
https://doi.org/10.1155/2022/5815693

Research Article

Research on the Generation of Creative Animation Driven by Deep Learning Model

Xiaokun Sang[1] and Lijuan Xu[2]

[1]*College of Art, Qingdao Huanghai University, Qingdao, Shandong 266247, China*
[2]*International Business School, Qingdao Huanghai University, Qingdao, Shandong 266247, China*

Correspondence should be addressed to Xiaokun Sang; sangxk@qdhhc.edu.cn

Received 10 March 2022; Revised 29 March 2022; Accepted 11 April 2022; Published 21 April 2022

Academic Editor: Jie Liu

It is a very interesting and practical task to transform real-world images such as portraits or scenery into creative animation images. Since this concept was put forward, it has aroused extensive research interest in the field of computer vision. The generative adversarial networks (GAN) model is widely used in this field. Depth convolution GAN (DCGAN) and Wasserstein GAN (WGAN) improve the original GAN, but there are still problems existing in creative animation generation such as model collapse. To solve these problems, the Wasserstein distance is introduced to replace the JS divergence in the GAN model to measure the gap between the sample distribution generated by the generator and the real distribution, and the loss function is improved. In order to achieve a better animation generation effect, the training of the model is further optimized through the adjustment of the network model structure and the setting of parameters. Through the comparison with DCGAN and WGAN models in the animation data set and CelebA data set and the quantitative analysis and comparison of the generation effects of different models, the effectiveness and generalization of the improved GAN model are verified.

1. Introduction

It is a very interesting task for computers to be used to generate creative animation images with artistic style. This task is mainly studied through image style transfer. Image style transfer focus uses a computer to stylize the content in an image, presenting a specific artistic style while the original content can be recognized [1, 2]. This is a new research direction in computer vision in recent years. Style transfer technology allows computers to create art "autonomously." Therefore, the concept has attracted people's attention since it was put forward.

Convolutional neural networks (CNN) and generative adversarial networks (GAN) learning models are basically adopted in the study of image style transfer [3, 4]. These models can add an art style to the target image, which has artistic properties. In image style transfer, firstly, CNN or GAN is used to learn style patterns from the specified style images. Then, they are converted into oil paintings, cartoons, Chinese landscape paintings, and other different artistic images, or the transformation of seasons and textures on the image is realized after it is applied to the target image.

GAN model is unstable and difficult to optimize in training. Many style migration efforts improve it from a loss function perspective. In these improved models, DualGAN [5] and CycleGAN [6] can complete the image style transfer work well. However, these GAN models can only migrate either style or content during style migration. In this paper, by improving the generator structure, the model achieves a better balance in the simultaneous transfer of style and content. The improved model is applied to the style transformation from natural images to animation illustrations. The experimental results show that the model can retain the content of the original natural scene and have a very excellent animation illustration style effect.

2. Related Works

As the most widely used generation model in the field of deep learning, GAN is also one of the models with the best visual effect on image generation. It is a new network model proposed by Ian Goodfellow of OpenAI in 2014 [4]. The model has attracted the attention of many scholars. The image quality generated by generating a countermeasure network model is higher than that of traditional generation models, such as variational self-encoder, and the training of the model is faster than that of the autoregressive model. However, the original GAN model also has some disadvantages, such as poor training stability and mode collapse. With the continuous research of scholars, various improved models have emerged one after another, and the performance of the generated countermeasure network has been greatly improved.

In view of the phenomenon that the training speed of the model is slow, the gradient is not updated in time, and even the model collapses when using the original GAN model. Mao et al. proposed the least squares GAN (LSGAN) [7], and the gradient of the model will be reduced to 0 when the data distribution is completely consistent with the real sample. The problem of gradient disappearance in model training caused by using Sigmoid and other functions as activation functions is avoided.

However, the real problem of GAN is how to better measure the gap between generated sample distribution and real sample distribution so that the generator can learn better. This problem did not make a major breakthrough until Wasserstein GAN (WGAN) was put forward.

WGAN was proposed by Arjovsky et al. [8]. The model uses Wasserstein distance instead of JS distance in traditional GAN as the standard to measure the difference between distributions. The common problem of JS divergence is that it is unable to measure the distance between two noncoincident distributions, and the gradient often disappears in the process of model training. Using Wasserstein distance can better measure the gap between the generated sample distribution and the real sample distribution, effectively alleviate the problems of mode collapse and training instability in network model training, and achieve good experimental results without a complex network model structure.

Ishaan Gulrajani et al. proposed WGAN's improved model WGAN-GP on the basis of WGAN [9]. In the model, the gradient penalty method is used to replace the weight clipping in WGAN to achieve the approximate 1-Lipschitz restriction effect on the discriminator network, and the normalization operation is cancelled in the discriminator network. David Berthelot et al. proposed the boundary equilibrium GAN (BEGAN) model and designed a new way to evaluate the generation quality of generators [10]. By estimating the difference between the distribution of distribution errors instead of the traditional generation, they can directly estimate the generation distribution and the real step-by-step errors in the antinetwork model. The model can also be trained stably under the standard GAN structure, and the model can converge quickly. At the same time, a super parameter is added to adjust the quality and diversity of the image generated by the generator.

The improvement methods mentioned above are to improve the original generated countermeasure network model from the loss function of the model. In terms of the structural improvement of the generated countermeasure network, the earliest is the deep convolution GAN (DCGAN) [11] proposed by Alec Radford et al. The model combines the powerful convolution neural network with the generator and discriminator that generates the countermeasure network, replaces the pool layer in the original generated countermeasure network with the convolution layer with step size, and uses the batch normalization operation [12] in the generator network and discriminator network to cancel the full connection layer in the network so that the generator can better learn the characteristic information of the image. The generated image has higher quality. Zhang et al. proposed self-attention GAN (SAGAN) [13] and added a self-control module to the model structure of the generation countermeasure network. The self-attention module can well deal with the long-range and multilayer dependence of image information. When generating the image, it can coordinate the details of each position and the details of the far end. At the same time, spectral regularization is added to the discriminator, which has achieved good results in the field of image generation. Andrew et al. proposed the BigGAN [14] model, which has achieved a major breakthrough in the field of image generation. The model not only increases the batch size but also increases the number of filters in each layer of the network. Through the shared embedding between network levels, the random noise and input conditions are spliced and input to each batch normalization layer of the generator network model, which greatly improves the quality of the generated image.

In order to apply the GAN model to a wider range of fields, Mirza and Osindero proposed conditional GAN (CGAN) [15]. By adding additional label information conditions to the generation network and discrimination network to guide the data generation process, the network can generate specific image samples according to the additional condition information. Phillip et al. proposed the pix2pix model [16] on the basis of CGAN and applied GAN to the field of image style migration. The model adds a U-net structure [17] to the generator network and an L1 regularization term to the loss function to realize the image translation task. In the image style conversion, the most popular is the cycle neural network GAN (cycle GAN) [18] proposed by Zhu et al. It realizes the conversion between images of two different styles (such as the conversion from horse to zebra) and the conversion between different painting styles. Different from the pix2pix model, the cycle GAN model can be trained in non-paired data set, while the training data and data of the pix2pix model must be paired. Yunjey et al. proposed the star GAN [19] model to realize the conversion between multiple different style fields through fewer generators. Star GAN realizes the image cross style conversion under different data sets with less training cost by adding one hot condition feature and mask vector to the model.

In the field of more widely used image restoration and image super-resolution reconstruction, there is the Deblur GAN [20] model proposed by Orest Kupyn et al. Based on conditional GAN structure and content loss, it realizes image deblurring through end-to-end learning.

In the practical application of image super-resolution reconstruction, the more popular is the SRGAN model [21] proposed by Christian et al. The generation network model of SRGAN adopts the deep-seated residual network as the network structure and adds the perception loss based on the VGG network [50] to the loss function so that GAN has more real details in generating super-resolution images and faster training speed.

3. Principles of GAN

GAN is also a generative model. But it does not have to explicitly express the probability distribution of the sample. It is the idea of adversarial learning. The intrinsic distribution of data is implicitly learned through a zero-sum game between generator and discriminator [22–25]. When the generator and discriminator reach a Nash equilibrium state, the data generated by the generator can have the same inherent properties as the real data. This allows using generators to get real data [25].

The GAN model consists of two basic modules: generator G and discriminator D. Generator G and discriminator D can be any learning model with generative and discriminant capabilities. Compared with the traditional shallow machine learning model, the deep model has richer parameters and stronger learning ability. In particular, CNN has unique advantages in processing image data. Therefore, CNN is generally used as a discriminator in the GAN model. CNN with transpose convolution structure is used as a generator.

When using the GAN model to generate image data, random noise vector Z needs to be input for generator G. The output result $G(z)$ is obtained by transposing convolution – upsampling – non-linear activation – batch normalization. G(z) has the same structure and size as real training data. They will be fed into discriminator D along with real training data X. Their labels are usually separated by zeros and ones. If the input sample is $G(z)$, then discriminator D should determine its category label as 0. If the input is true sample X, the category label of it is judged to be 1. In the training process, discriminator D needs to maximize the accuracy of label prediction for X and $G(z)$. Generator G, meanwhile, tries to make the generated $G(z)$ indistinguishable from the x from the real training set. Thus, generator G and discriminator D will constantly play against each other throughout the training process. The generative and discriminant abilities of both will be improved continuously. The output of the final generator will have the same appearance as the real data. Judge D will not be able to distinguish the true source of the data. The classification probability of both the real sample and the "false" data generated by G will approach 1/2. At this point, you can assume that generator G has learned the inherent distribution of real data. The "fake" data it generates already has the same properties as the real data. This enables data distribution without explicitly expressing it. The goal of the intrinsic distribution of training data is obtained through adversarial learning.

The learning objectives of the GAN model can use the following form of expression:

$$\min_{G}\max_{D} L_{\mathrm{GAN}}(D,G) = E_{x\sim p_{\mathrm{data}}}[\log D(x)] + E_{z\sim p_z}[\log(1-D(G(z)))], \tag{1}$$

where p_{data} is the probability distribution obeyed by real training sample x. $p_z(z)$ is the probability distribution that noise z obeys. $E_{x\sim p_{\mathrm{data}}}[\cdot]$ and $E_{z\sim p_z}[\cdot]$ are the mathematical expectation of x and z classification probability output by discriminator D, respectively.

4. Animation Style Migration Model

A deep convolution generated countermeasure network (DCGAN) is an improved model of generating countermeasure network GAN. Its principle is consistent with that of GAN. The biggest improvement is the perfect combination of convolution neural network, which is most widely used in image processing, and generated countermeasure network. Deep convolution generated countermeasure network uses convolution neural network structure for both generator and discriminator in the model, At the same time, some changes are made to the structure of the added convolutional neural network to improve the performance of the network model.

4.1. Improvement of Loss Function.

Since GAN was proposed in 2014, although it has been widely used in the field of machine vision and achieved good results, the initial GAN model often has problems such as training difficulties, unbalanced training between generator and discriminator, and insufficient diversity of samples generated by the generator. In the original GAN model, KL (Kullback–Leibler divergence) is used to measure the gap between the sample distribution generated by the generator and the real distribution. The loss function used in the standard generation countermeasure network model is shown in formula (1).

From formula (1), it can be calculated that when the parameters of generator g are fixed, it is the optimal discriminator D. The calculation process is as follows:

$$\min_{G}\max_{D} V(D,G) = E_{x\sim p_r(x)}[\log D(x)] + E_{x\sim p_g(x)}[\log(1-D(x))], \tag{2}$$

where $E_{x\sim p_r(x)}[\log D(x)]$ represents the probability distribution that sample x belongs to real data and $E_{x\sim p_g(x)}[\log(1-D(x))]$ represents the probability distribution that sample x belongs to the sample data generated by the generator.

Then the contribution of x to the loss function is

$$\mathrm{contribution}(x) = p_r \log D(x) + p_g \log(1-D(x)). \tag{3}$$

By taking the derivative of formula (3) to $D(x)$ and making its derivative value 0, it can be obtained that

$$\frac{p_r}{\log D(x)} + \frac{p_g}{\log(1 - D(x))} = 0. \tag{4}$$

By simplification, the best discriminator can be obtained as follows:

$$D(x) = \frac{p_r}{p_r + p_g}. \tag{5}$$

The above results show that the task of the discriminator is to judge the possibility that the input sample x comes from real data and generated data. When $p_r(x) = 0$ and $p_g(x) \neq 0$, the discriminator can easily determine the source of x. When $p_r(x) = p_g(x)$, it indicates that the probability that the sample belongs to the real sample is equal to that of the generated sample. At this time, the output of the optimal discriminator is 0.5, which means that the generator and the discriminator have reached Nash equilibrium. The sample generated by the generator is enough to confuse the true with the false so that the discriminator cannot make a correct judgment on the input sample. If the obtained optimal discriminator is replaced back into the loss formula of the original GAN, the following can be obtained:

$$\begin{aligned}
&E_{x\sim p_r(x)}\left[\log\frac{p_r}{p_r + p_g}\right] + E_{x\sim p_g(x)}\left[\log\left(1 - \log\frac{p_r}{p_r + p_g}\right)\right] \\
&= \int_{x\in X} p_r \log\frac{p_r}{p_r + p_g}\mathrm{d}x + \int_{x\in X} p_g \log\frac{p_r}{p_r + p_g}\mathrm{d}x \\
&= \int_{x\in X} p_r \log\frac{2p_r}{p_r + p_g}\mathrm{d}x + \int_{x\in X} p_g \log\frac{2p_r}{p_r + p_g}\mathrm{d}x - 2\log 2 \\
&= KL\left(p_r \| \frac{p_r + p_g}{2}\right) + KL\left(p_g \| \frac{p_r + p_g}{2}\right) - 2\log 2 \\
&= 2JS\left(p_r \| p_g\right) - 2\log 2.
\end{aligned} \tag{6}$$

It can be seen from the above results that the form of the optimal discriminator can be obtained according to the loss function in the original generated countermeasure network. When the discriminator is in the optimal state, the generator loss defined by the original generation countermeasure network can be equivalent to minimizing the Jensen–Shannon (JS) divergence between the real sample data distribution and the generated data sample distribution.

However, there is often a problem when optimizing JS divergence. No matter whether the two data distributions are very close or far apart, as long as there is no overlap between the two data distributions or the overlap can be ignored, JS divergence will not be updated and will always be the fixed value log2. This also means that the gradient vanishing problem will occur when using the loss function of the original GAN for training.

In the process of generating countermeasure network training, especially at the beginning of training, the input of the generator is random noise, so there is no intersection between the large probability of the sample distribution generated by the generator and the real sample distribution. As a result, the JS divergence is fixed at the constant log2, and the gradient is 0, so the gradient descent method cannot be used to train the network parameters. At this time, for the generator network, there will be no gradient information fed back from the discriminator network, which leads to the disappearance of the gradient in the network training. Therefore, the training instability and model collapse often occur in the original generated countermeasure network. On the one hand, when the discriminator network is trained too well, the gradient fed back to the generator network will disappear, and the generator network cannot be updated and optimized. On the other hand, when the discriminator network is not trained well, the correct gradient cannot be fed back to the generator network to guide the generator network to optimize better.

Therefore, we use the loss function of the WGAN model to replace the loss function of the original generated countermeasure network and use Wasserstein distance instead of JS divergence to measure the distance between two data distributions, which effectively reduces the instability of model training. Wasserstein distance is also called earth-mover (EM) distance, which is defined as follows:

$$W\left(p_r, p_g\right) = \inf_{\gamma\sim\prod\left(p_r, p_g\right)} E_{(x,y)}[\|x - y\|]. \tag{7}$$

The advantage of Wasserstein distance over KL divergence and JS divergence in the original generated countermeasure network is that it can reflect the distance between the two data distributions without overlapping.

In order to add Wasserstein distance to the loss function, a constrained discriminator is proposed, that is, it satisfies 1-Lipschitz continuity, and the Lipschitz continuity condition limits the maximum local variation amplitude of a continuous function. In order to meet the constraints, the weight updated during backpropagation is forcibly trimmed to the specified range by weight clipping, and then the $V(G,D)$ is maximized to realize the training of the model.

The proposed loss function is

$$V(G, D) = \max_{D\in 1\_\text{Lipschits}} E_{x\sim P_{\text{data}}}[D(x)] - E_{x\sim P_G}[D(x)]. \tag{8}$$

The larger the value of the proposed loss function, the closer the generated data distribution is to the real data distribution, and the better the network training.

4.2. Structure of the Migration Model for Creative Animation Generation. Similar to the general GAN model, the animation illustration style transfer model proposed in this paper is also composed of generators and discriminators. The internal distribution of data can be obtained by leaning against each other. In order to better retain the original content of images and achieve the transfer of artistic styles in animation illustration style transfer, we design the generator structure of the deep learning-driven migration model for creative animation generation as shown in Figure 1. Taking ResNet-18 as the basic model, the generator structure

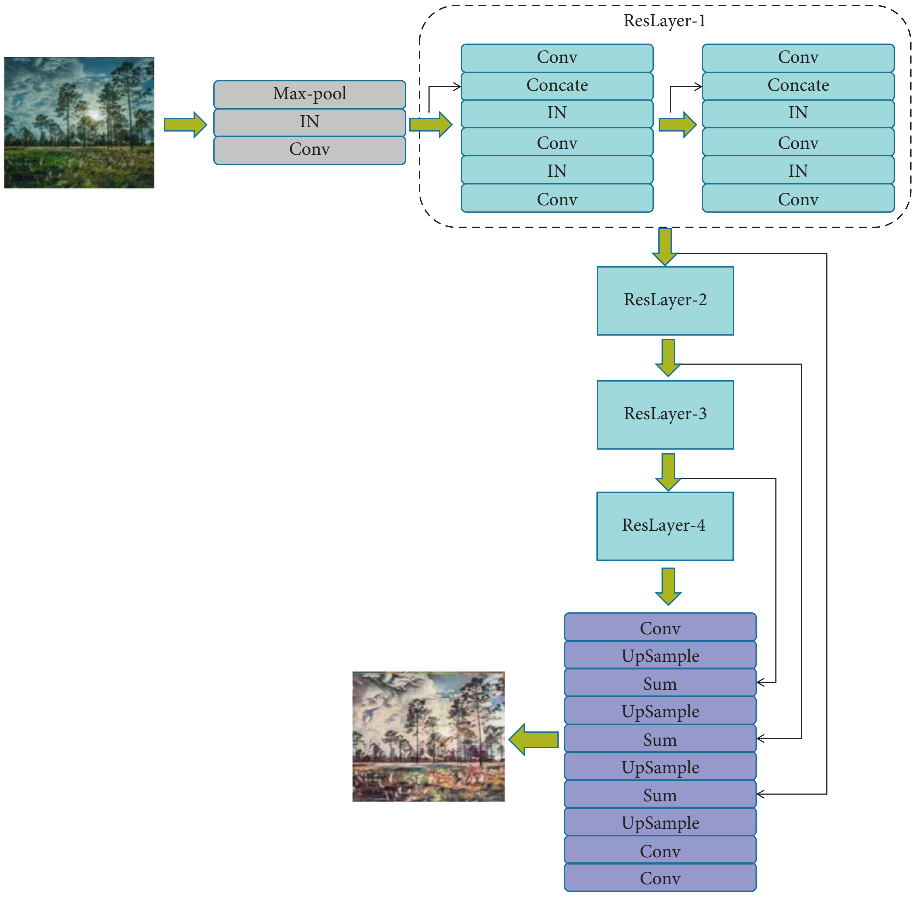

Figure 1: The generator structure of the deep learning-driven migration model for creative animation generation.

divides image generation into two parts, including downsampling and upsampling.

The downsampling part of the image is composed of a basic convolution module and four residual layers (reslayer-1–reslayer-4). However, the basic convolution module consists of four layers: Conv – Instance Norm (IN) – ReLU – MaxPooling. The convolution kernel used is 7 × 7. This makes the convolution operation have a relatively large receptive field. In the four residual layers, each residual layer contains two residual blocks. Their internal structure is Conv-IN-ReLU-Conv-IN. The output of the latter IN layer will concatenate with the input of the entire residual layer. And then it goes through another convolution layer. The entire residual block has the same short-circuit connection structure as BottelNeck in ResNet. This allows the input of the entire residual layer to be reused. It can effectively improve the gradient propagation performance of network optimization. In the downsampling part, each convolution layer uses a 3 × 3 convolution kernel. After each residual layer, 1/2 horizontal and vertical downsampling is done.

After the basic convolution module and four residual layers, the size of the feature map will be 1/16 of the original image. It will then be upsampled. First, the feature graph output by reslayer-4 is convolved. It is then upsampled to restore the feature image to 1/8 of the original image. As shown in Figure 1, add it to the output of reslayer-3 for upsampling. This operation is then repeated until the output of reslayer-1 is added. Such a short circuit connection makes the details of the feature map of the previous processing better preserved. This avoids damage to the content of the image when migrating styles later. After a short circuit connection and addition, the feature graph passes through two convolution layers. It will be transformed back into a three-channel image again. In the convolution operation of the upsampling part, the convolution kernel of the first two convolution layers is 1 × 1. The convolution kernel at the last layer is set to 7 × 7. Tanh activation function is used before conversion to a three-channel image.

PatchGAN discriminator structure of 70 × 70 was used in the discriminator in this paper. Compared with the general convolutional neural network structure, the PatchGAN discriminator has fewer parameters and can receive images of arbitrary size. The PatchGAN discriminator contains three convolution blocks. Each block contains two convolution layers. In terms of the number of channels in the convolution kernel, this paper sets the output channel of the first convolution layer as 64. The number of channels is doubled in each subsequent block.

In the loss function design of the model, this paper adopts the same cyclic consistency loss as that in CycleGAN. For image transformation G and F, CycleLoss means that the result of source image transformation after $x \longrightarrow G(x) \longrightarrow$

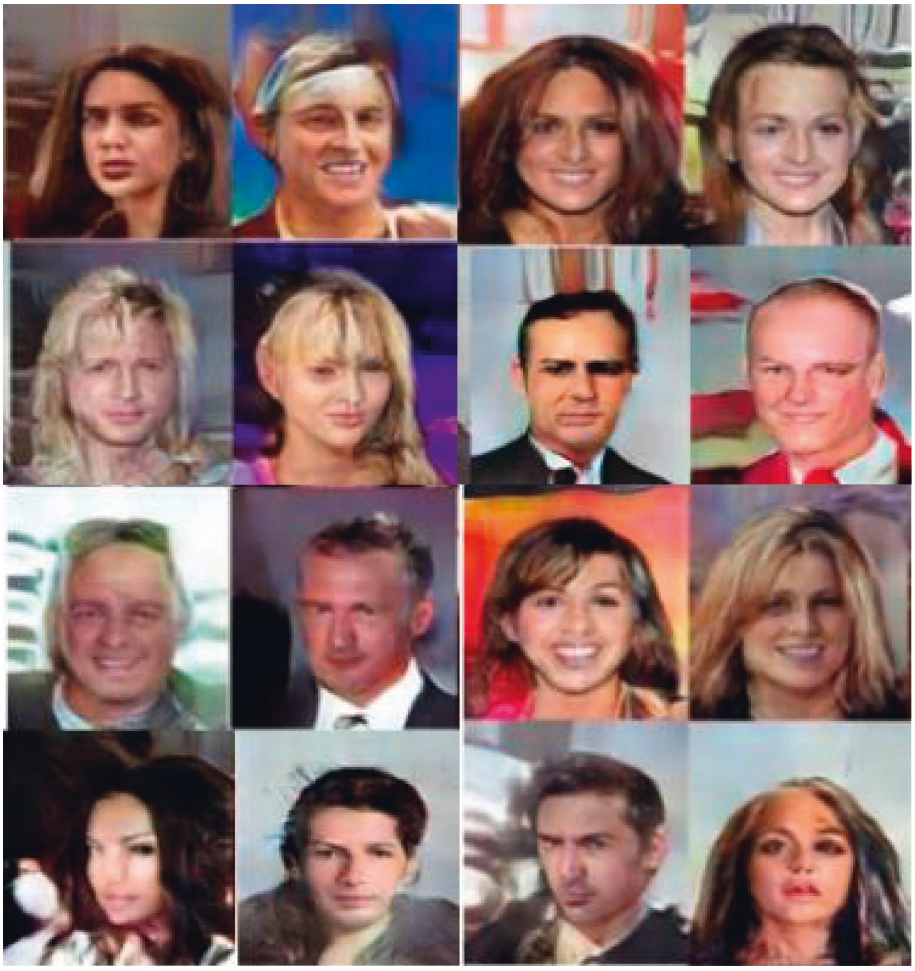

Figure 2: The original representative partial face image.

$F(G(x))$ should have the attribute of $F(G(x)) = x$. Similarly, for the target image through $y \longrightarrow F(y) \longrightarrow G(F(y))$ should also be $G(F(y)) = y$. When L1 distance is used to measure the difference between the result after cyclic transformation and the original image, CycleLoss can be expressed as follows:

$$L_{\text{cyc}}(G,F) = E_{x\sim p_{\text{data}}}\left[\left\|F(G(x)-x)_1\right\|\right] + E_{y\sim p_{\text{data}}}\left[\left\|F(G(x)-y)_1\right\|\right]. \tag{9}$$

When using the CycleLoss training model, we need to consider not only the cyclic consistency loss from source image to target image to source image but also the backtracking CycleLoss from target image to source image to target image. Therefore, two pairs of generators and discriminators need to be set up simultaneously in the model. It was also trained with CycleLoss in formula (2). The final loss function can be expressed as follows:

$$\begin{aligned} L\left(G,F,D_x,D_y\right) = L_{\text{GAN}}\left(G,D_x\right) + L_{\text{GAN}}\left(F,D_y\right) \\ + \lambda L_{\text{cyc}}(G,F), \end{aligned} \tag{10}$$

where D_{x} and D_{y} refer to the discriminators of source image x and target image y, respectively. λ is the equilibrium parameter set based on experience.

4.3. Image Evaluation Index. In the image generation task, the evaluation of the result quality of the generated image not only can rely on the subjective judgment of human vision but also need to analyze the generated image quantitatively. It is mainly considered from two aspects: (1) the quality of the generated image itself, that is, whether the image content is realistic and whether the image details are clear, and (2) for the diversity of generated images, a good generation should generate a variety of images rather than a fixed number of similar types of images. At present, in the field of image generation, the evaluation indicators are is IS (inception score) and FID (Fréchet inception distance).

4.3.1. IS. It uses the pretrained inception neural network as the classifier, inputs the image samples generated by the generator into the classifier, and statistically analyzes the output value of the classifier. Its calculation is

$$IS(G) = \exp\left(E_{x\sim p_g} KL\left(p(y|x)\| \, p(y)\right)\right), \tag{11}$$

where $x \sim p_g$ means that x is the image sample generated from pg, $KL(p(y|x)\| \, p(y))$ means that KL divergence is used to measure the distance between two distributions, $p(y|x)$ represents the probability that the image sample x is

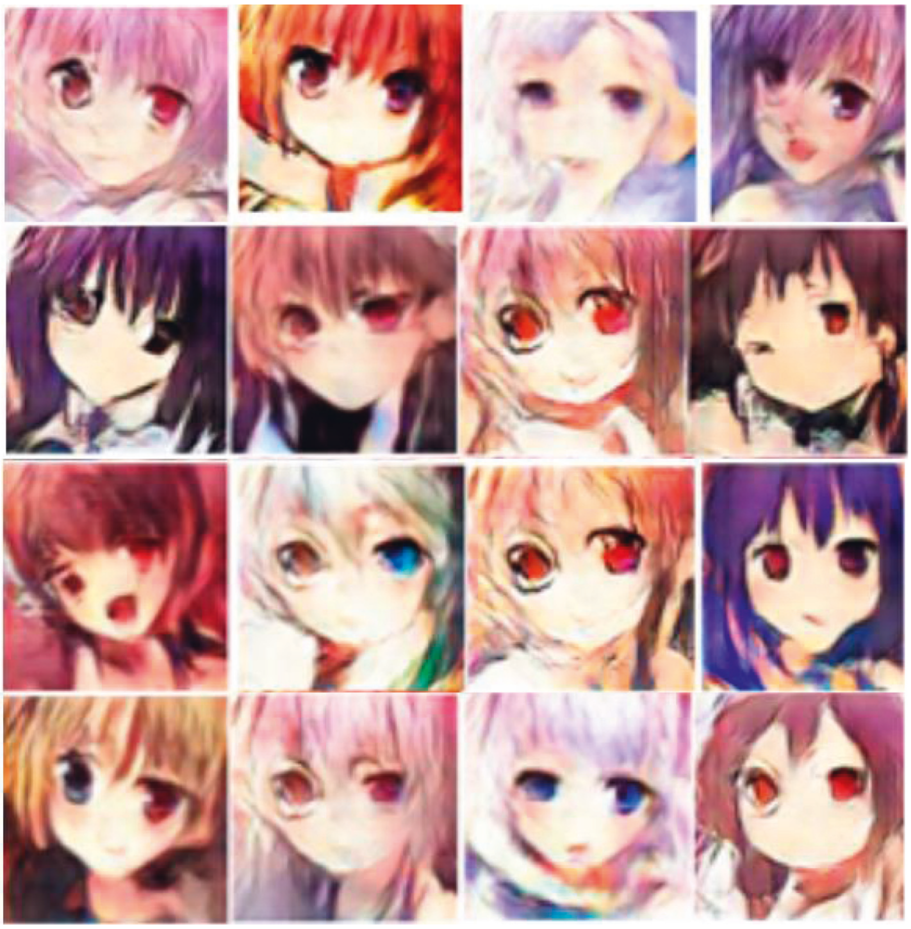

Figure 3: The animation generated by the original DCGAN model.

classified as y, and $p(y) = \int_x p(y|x)p_g(x)$ represents the edge distribution of all categories of images.

The larger the IS value, the better the image generated by the generator model.

4.3.2. FID. It is a method to evaluate the image quality by calculating the distance between the feature vector of the real image and the generated image, and the feature vector of the image is extracted after removing the last layer of the network through the perception neural network. The calculation is

$$\text{FID}\left(P_r, P_g\right) = \left\|\mu_r - \mu_g\right\|^2 + Tr\left(\sum r + \sum g - 2\sqrt{\sum r \sum g}\right), \tag{12}$$

where μ represents the mean vector of the real image and the generated image in the feature space, $\sum$ represents the covariance matrix of the real image and the generated image in the feature space, and Tr represents the trace of the matrix.

On the contrary to IS, if the FID value is smaller, it means that the similarity between the generated image and the real image is higher, indicating that the generation effect of the model is better.

5. Experimental Results and Analysis

5.1. Experimental Data. The animation avatar data set used in the model training in this paper is randomly crawled from the animation material website SafeBooru through the web crawler and screened it. Finally, 150,000 animation images are obtained; 60,000 images are randomly selected as the training samples; and the image size is uniformly processed to 96 × 96 for the experiment.

In order to verify the generalization of the improved model, experiments were carried out on CelebFace data set. A total of 10,200 samples and 202,677 face data were collected in this data set, and the face styles in the images were quite different. All the image sizes were 178 × 218. However, if all the data sets were used as training data, the training time of the model would be too long, so 100,000 images are used as training data in this experiment. At the same time, the size of the original image is 178 × 218, which is not conducive to the construction of the neural network model. It is necessary to preprocess the original image and change it to the size of 128 × 160, which not only can ensure the simplicity of the network model but also can ensure the image proportion.

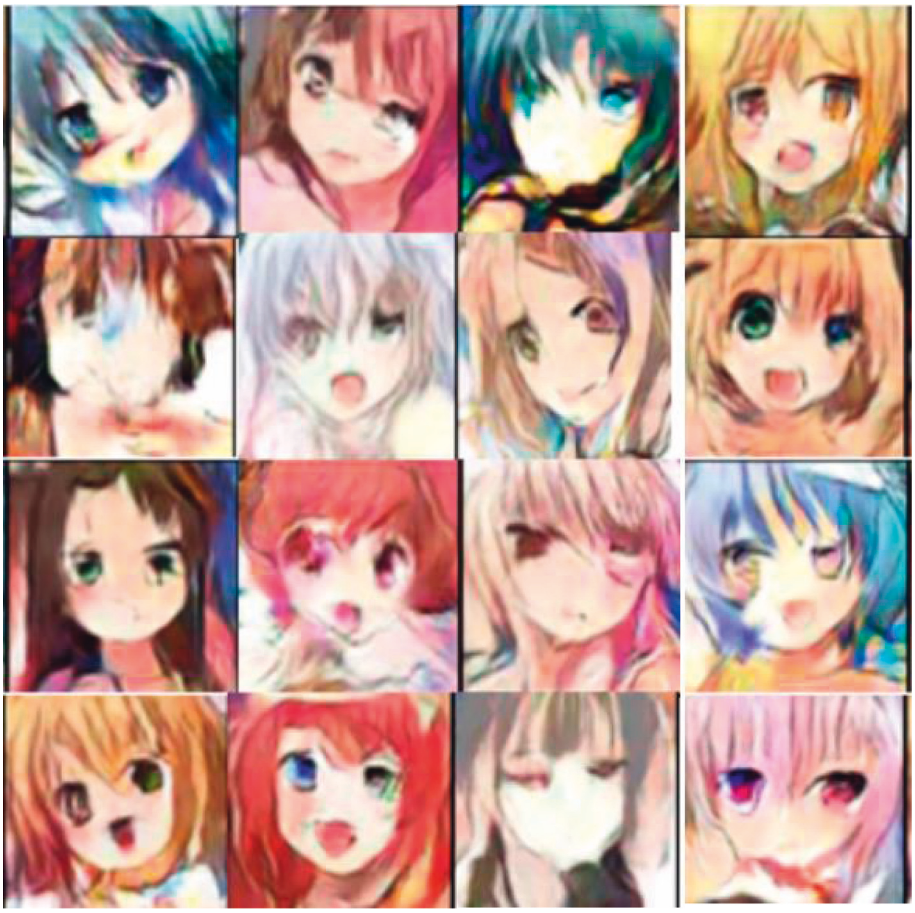

Figure 4: The animation generated by the original WGAN model.

5.2. Experimental Configuration. PyTorch deep learning framework was used in the Ubuntu 18.04 environment. It uses an NVIDIA-1080 GPU with CUDA10 for acceleration. An SGD optimizer with a learning rate of 0.002 was used to optimize the model as 200 Epochs. It was then used to test the generation of animation illustration-style images.

5.3. Results and Discussion. Figure 2 is the original representative partial face image, which shows the training results obtained from the CelebA data set. Due to the excessive number of original sample sets, 100,000 face images are randomly selected as training samples in order to reduce training time.

In order to verify the performance of the improved model proposed in this paper in animation generation, comparative experiments are carried out on the original DCGAN model and the original WGAN model in the 100,000 face image data set shown in Figure 2. The comparative experimental results are shown in Figures 3–5. Through the steps of facial emotion recognition and information aggregation, Figures 3–5 show the effects of three different GAN methods in the animation data set. Figure 3 shows the animation generated by the original DCGAN model; Figure 4 shows the generation results of the WGAN model; and Figure 5 shows the animation effect generated by the proposed algorithm.

It is not difficult to see that although the images generated by the three different methods have good identifiability.

However, compared with other methods, the animation image content generated by traditional DCGAN lacks authenticity, and the facial details of the generated animation characters are seriously lost, which gives people a sense of disharmony, and the whole image appears the phenomenon of information collapse.

For the animation generated by the WGAN model, it performs well in the color brightness of the whole image, but the image quality is significantly lower than the other two methods, and the facial features of the animation avatar in the generated image are not clear.

Compared with these two methods, the improved GAN model designed in this paper combines the respective advantages of the original DCGAN and WGAN. The generated animation avatar is closer to the real sample; the details generated on the image are clearer; and the color saturation is high and has stronger authenticity.

The above only analyzes and compares the generation effects of the original DCGAN model, WGAN model, and the algorithm model in this paper on the same data set from

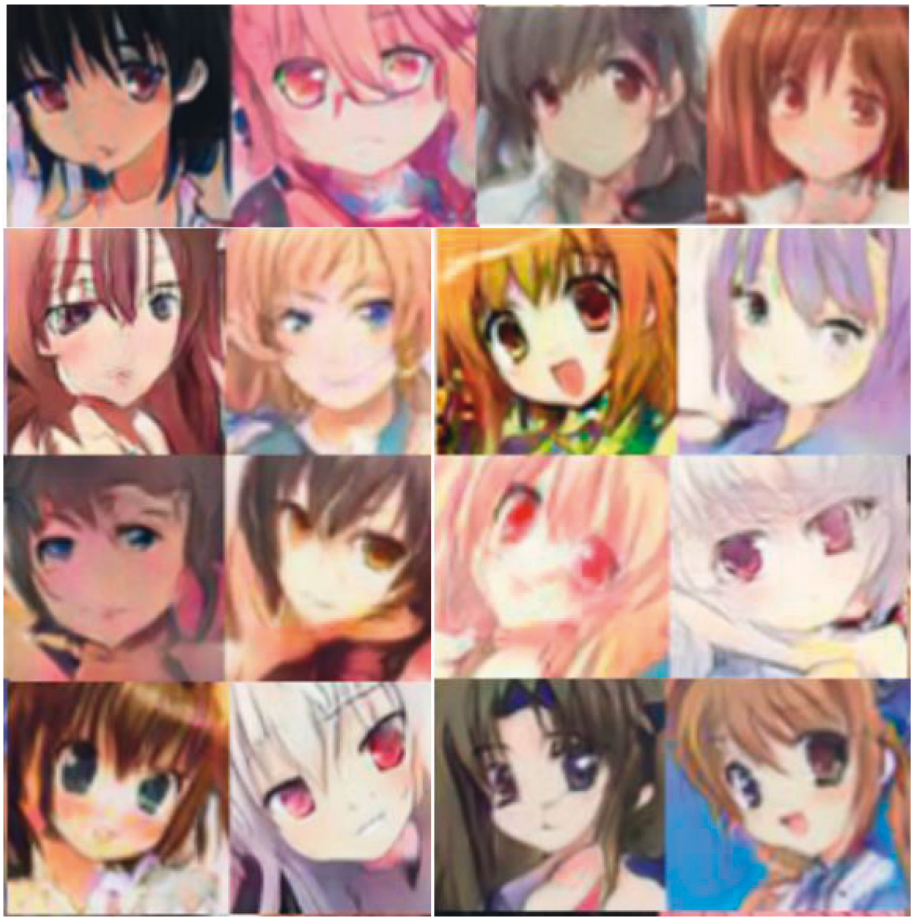

Figure 5: The animation generated by the proposed GAN model.

Table 1: The evaluation results of three models.

Model	Evaluation indicators			
	IS		FID	
	CelebA data set	Animation	CelebA data set	Animation
DCGAN	6.22 ± 0.13	6.56 ± 0.25	43.22 ± 0.22	39.52 ± 0.12
WGAN	7.01 ± 0.15	7.12 ± 0.18	40.55 ± 0.10	36.36 ± 0.22
The proposed GAN model	7.45 ± 0.22	7.88 ± 0.24	35.69 ± 0.23	31.28 ± 0.26

the perspective of human visual intuition, which has a certain subjectivity. In order to evaluate these three models more objectively, we use IS and FID as quantitative evaluation indicators of the image effect generated by the model. These two evaluation indicators are the most widely used evaluation indexes in the GAN model at present. Through these two indicators, the animation generated by three different network models is evaluated, and the evaluation results are shown in Table 1.

For these two evaluation indicators, the larger the value of IS, the better the quality and diversity of the generated animation image, while the smaller the value of FID, the better the diversity and quality of the generated animation image. From the data in Table 1, it can be concluded that the proposed GAN model has better performance in generating images than the original DCGAN model and WGAN model. The score of the same model in the CelebA data set is lower than that in animation data set due to the influence of character background information. It can be seen that in the training of depth model, the quality of the data set also has a great impact on the final training results of the model.

6. Conclusion

This paper studies the style transfer of animation based on the GAN model. A new generator network is designed to allow simultaneous migration of image style and content. After training using natural world images as source domain data and art illustration images as target domain data. This method can generate animation images with excellent visual quality. Compared with the images generated by DCGAN and WGAN models, the proposed method achieves a better balance between the image style and the original image content. Aiming at the problem of model collapse that often occurs in the process of network model training, in order to avoid this problem, this paper uses Wasserstein distance instead of JS divergence as the measurement standard. In order to make the weight meet the constraints, the model adopts the method of weight forced cutting, which is not conducive to the learning of the network model. In the next work, we will consider using gradient punishment instead of weight forced cutting to make the weight meet the constraints.

Data Availability

The data set can be accessed upon request to the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Special Subject of Art Education in Shandong Province in 2021 “Teaching Exploration and Practical Research on the Form of Image Expression in the Creation of Red Culture Theme Oil Painting--Taking Yimeng Spirit as an Example” (L2021Y10290453) and 2022 National Art Fund Communication, Exchange and Promotion Project “River Surging: Yellow River Culture and Art Exhibition.”

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July2016.

[2] Y. Jing, Y. Yang, Z. Feng, J. Ye, Y. Yu, and M. Song, “Neural style transfer: a review,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 26, no. 11, pp. 3365–3385, 2020.

[3] S. Ji, W. Xu, M. Yang, and K. Yu, “3D convolutional neural networks for human action recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[4] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Isgum, “Generative adversarial networks for noise reduction in low-dose CT,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2536–2545, 2017.

[5] Z. Yi, Z. Hao, and P. Gong, “DualGAN: unsupervised dual learning for image-to-image translation,” in *Proceedings of the IEEE International Conference on Computer Vision*, October 2017.

[6] C. Bo, Q. Zhang, S. Pan, and L. Meng, “Generating Handwritten Chinese Characters Using CycleGAN,” in *Proceedings of the 2018 Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, IEEE Computer Society, Lake Tahoe, NV, USA, March 2018.

[7] X. Mao, Q. Li, H. Xie, L. Raymond, and Z. Wang, “Least squares generative adversarial networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802, Venice, Italy, October 2017.

[8] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning*, Sydney, Australia, August 2017.

[9] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, “Improved training of Wasserstein GANs,” 2017, https://arxiv.org/abs/1704.00028.

[10] D. Berthelot, T. Schumm, and L. Metz, “BEGAN: boundary equilibrium generative adversarial networks,” 2017, https://arxiv.org/abs/1703.10717.

[11] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” 2015, https://arxiv.org/abs/1511.06434.

[12] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Lille, France, July 2015.

[13] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, “Self-attention generative adversarial networks,” 2018, https://arxiv.org/abs/1805.08318.

[14] A. Brock, J. Donahue, and K. Simonyan, “Large scale GAN training for high fidelity natural image synthesis,” 2018, https://arxiv.org/abs/1809.11096.

[15] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” 2014, https://arxiv.org/abs/1411.1784.

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” 2017, https://arxiv.org/abs/1611.07004.

[17] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: convolutional networks for biomedical image segmentation,” 2015, https://arxiv.org/abs/1505.04597.

[18] J. Y. Zhu, T. Park, P. Isola, and A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of the IEEE international conference on computer vision*, pp. 2223–2232, October 2017.

[19] Y. Choi, M. Choi, M. Kim, J.-W. Ha, and S. Kim, “Stargan: unified generative adversarial networks for multi-domain imageto-image translation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8789–8797, Salt Lake City, UT, USA, June 2018.

[20] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and Ji Matas, “Deblurgan: blind motion deblurring using conditional adversarial networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8183–8192, Salt Lake City, UT, USA, June2018.

[21] C. Ledig, L. Theis, F. Huszár, and J. Caballero, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4681–4690, Honolulu, HW, USA, July 2017.

[22] C. Li and M. Wand, “Precomputed real-time texture synthesis with markovian generative adversarial networks,” in *Proceedings of the European Conference on Computer Vision*, October2016.

[23] A. Kim, T. Jang, and O. K. Chang, “A run-to-run controller for a chemical mechanical planarization process using least

squares generative adversarial networks," *Journal of Intelligent Manufacturing*, vol. 32, pp. 1–14, 2020.

[24] Q. Creswell, Q. White, Y. Dumoulin, M. Liu, B. Sengupta, and A. A Bharath, "Sketch simplification based on conditional random field and least squares generative adversarial networks," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.

[25] Q. Lu, Q. Tao, Y. Zhao, and M Liu, "Sketch simplification based on conditional random field and least squares generative adversarial networks," *Neurocomputing*, vol. 316, pp. 178–189, 2018.