Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Image Style Conversion Optimization Method in Animation Design Based on Deep Convolutional Neural Networks

Xu Wang[1,2,✉], Jung Won-joon[2]

[1] *Jilin Animation Institute, Changchun, Jilin, 130012, China*

[2] *Tongmyong University, Busan, 48520, Nouth Korea*

ABSTRACT

With the continuous development and popularization of the animation field, animation style products have received more and more attention, and the traditional style migration algorithm can no longer meet people's needs. In this paper, after researching the style principle of convolutional neural network and generative adversarial network, we propose an improved image style conversion algorithm based on convolutional neural network, introducing depth-separable convolution and cross-layer connection method to realize the information fusion of low-layer convolution and high-layer convolution, and at the same time, combining four different functions for the fitting of loss function. The proposed method in this paper is trained and validated on Facades dataset, Monet2photo dataset, Summer2winter dataset and Apple2orange dataset, and the quantitative experimental results show that the MOS value of the proposed method in this paper is improved on average compared with CycleGAN and StarGAN by about 12.67% and 52.93%, thus it can be concluded that the method proposed in this paper effectively improves the overall texture details and style quality of image style transformation in animation design.

Keywords: convolutional neural network, generative adversarial network, image style transformation, animation design

1. Introduction

Animation is a visual art, it is relying on the strong visual expression to attract the attention of the audience, and the image style in animation design is the key means to truly express its audiovisual characteristics [1-2]. The term style has different points in different literary works. Generally speaking, in animation works, style contains double meanings, which is not only the sensory style from the perspective of audiovisual language, but also the directing style and plot style that combines narrative techniques and directing skills [3-5]. Audio-visual language style and plot style play an equally important role, and it can be said that the plot, characters, art design, camera editing, and any other

✉ Corresponding author.
E-mail address: wangxu00sharon@163.com (X. Wang).
Received 05 January 2024; Revised 12 February 2024; Accepted 15 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-244

work categories have their own style positioning [6-8]. And it is these elements that make up an animation work together to make up the overall style of the animation, so that the animation as a whole presents a representative and unique look.

In recent years, art style drama animation films have thus attracted widespread attention in society. The production of artistic animation requires artists to draw huge amounts of artworks, which are then stitched together to ultimately present a beautiful work [9-11]. Image style transformation using the field of deep learning can convert realistic style images into images of different artists' styles, which can save a lot of human and material resources in the process of animation design and reduce the pressure of art creation [12-15]. Image style transformation is also called image stylized rendering, the core principle of image style transformation is to use algorithms to learn the style of the target image first, get the pre-training model after the training is completed, and then input the image that needs to be converted, and the style of the input image will be converted to the target image style [16-18]. The essence is to separate the content and style features of the image, for different image content and features are reorganized to get the desired target [19-21]. Therefore, this paper investigates some advanced methods developed in the field of image style conversion, aiming to achieve clear and realistic image style conversion results.

Image style transformation techniques have been widely used in the field of animation design. He, J. emphasized the importance of motion style migration in the field of animation design for generating realistic and dynamic human animations, which can automatically create realistic motion samples from existing motion data, replacing the manual creation process in traditional animation design, while the introduction of deep learning algorithms allows the method to thoroughly and accurately generate motion sequences [22]. Kim, S. explored the potential of neural style migration techniques in animated filmmaking, where style migration networks innovate production methods for the animation design process and also provide the impetus to create images with unique aesthetics, contributing to the development of contemporary animation aesthetics [23]. Liu, D. S. M. et al. described the complexity of artificial image style migration by introducing the production process of the dynamic artwork “Qingming Shanghe Tu”, while the automated animation generation method with style migration technology can quickly achieve the effect of animation synthesis and reduce the repetitive work of animation design while maintaining a consistent painting style [24]. Wang, R. pointed out that there are a lot of repetitive manual operations in the current image style transformation process in the process of art creation and animation creation, while the image style migration method based on the deep learning framework is able to purchase the extraction of style features in the images and realize the automatic transformation of styles between images [25].

The distribution of special effects is an important factor in measuring the visual effect of animation works, and image style migration based on deep learning and artificial intelligence technology not only perfectly presents the creator's sense of thinking, but also greatly reduces the cost and improves the work efficiency. Fan, X. discusses the practical value of deep learning framework based on the convolutional neural network model VGG19 in the design of animation pattern special effects, which shows that the image style migration algorithm can produce unique animation special effects artistic effects, thus providing new ideas and contexts for animation pattern special effects design [26]. Li, S. evaluated the effect of animation effects design based on AI style migration algorithm, and the experiments showed that AI style migration algorithm is widely used in the field of 3D animation and computer graphic design, and most of the users are satisfied with its animation effects design [27]. While style migration for video animation requires animators to convert the style of an image frame by frame and subsequently generate a complete video sequence, deep learning image migration methods reduce the duplication of labor in the process, effectively easing the pressure on creators. Ruder, M. et al. proposed two image style migration methods for video sequences, one is to introduce a new initialization method and loss function into the original image style migration technique based on energy minimization, and the other is to construct a deep network architecture and training

procedure, and both methods are able to generate consistent and stable stylized video sequences [28]. Yang, S. et al. described a dynamic migration method for artistic text styles, which uses a bidirectional shape-matching framework to construct shape mappings of multi-scale symbolic styles to migrate the appearance and motion styles of the video content to the target text in order to create high-quality, controllable artistic text animations [29]. In addition, there are many types of animation creation forms, such as sketching, portrait drawing, cartoon drawing, and ink drawing, etc., which require image style migration technology to assist the creation due to the complicated production process, and deep learning technology significantly improves the performance of image style conversion. Zhang, L. et al. investigated a specific style migration task in anime sketch images by combining residual U-net and auxiliary classifier generative adversarial network (AC-GAN) to migrate the style of the content images into grayscale sketches, which is automated, fast and generates high quality [30]. Song, G. et al. showed that the art form of portraiture has evolved from a realist style to a creative style, based on which a reverse consistent migration learning framework, AgileGAN, is proposed, which captures attribute-dependent stylization of face features and achieves good performance in datasets such as 3D cartoons and comics [31]. Zhang, Z. et al. used a variational autoencoder to dynamically encode the style information of a specific cartoon image to complete the process of projecting the style features of the image to the content features, and further introduced the cartoon contrast learning loss to realize the scaled image style migration [32]. Xi, H. et al. outlined the creative idea of realizing the effect of ink animation in the style migration of animated images, and provided technical support and thinking innovation for the practice from the art of ink animation through the work of algorithm design, model optimization and parameter adjustment, providing technical support and thinking innovation for the practice from ink animation art [33].

The article firstly introduces the commonly used VGG structural model for style migration using convolutional neural networks, and how to extract features for content representation and style representation using VGG networks. Then, it conducts related research on generative adversarial networks, and then, it thoroughly describes the image style transformation algorithm based on convolutional neural networks and generative adversarial networks. On the conception of cross-layer connected generator network, the original encoder is replaced by depth-separable convolution, which enables the generator to adaptively learn features with different resolutions and speed up the iteration speed. On the conception of the loss function, the design and optimization of the loss function is combined with four different loss functions. Finally, the algorithm proposed in this paper is trained and validated on multiple datasets, and a variety of classical style migration algorithms are used for comparison experiments, and the subjective qualitative comparison and objective quantitative comparison results verify the superiority of this paper's algorithm in the style migration task. At the same time, the algorithm of this paper is subjected to performance test experiments and qualitative style migration experiments.

2. Method

2.1. Principles of style migration based on convolutional neural networks

2.1.1. VGG network structure. The structure of the VGG network is shown in Fig. 1.The VGG network has 5 convolutional segments, each with 2-3 convolutional layers, and a maximum pooling layer is attached to the end of each segment to reduce the image size. Each segment has the same number of convolutional kernels within it, with the number of kernels increasing the further back you go, with VGG16 and VGG19 being the most commonly used. The structure are multiple identical 3x3 convolutional layers stacked on top of each other to show that two 3x3 convolutions in tandem have the same receptive field as a 5x5 convolution, while three 3x3 convolutions in tandem have the same receptive field as a 7x7 convolution, and this way of designing the structure reduces the number of learning parameters, reduces overfitting, and allows the network to learn the features better, which

makes the selection of the VGG network structure to do feature extraction for style migration has a good advantage.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig. 1. VGG network structure diagram

2.1.2. Content characterization. In the study of VGG network structure, it was found that the complexity of the nonlinear filter banks in each layer of the network increases with the depth of the network. For an input image $\vec{x}$, its feature information will be calculated by the activation function in each layer of the network, if there are N_i different filters in one layer, then N_i feature maps of M_i size will be obtained by the filtering operation, M_i indicates the feature map size size, so the content feature representation in layer L can be expressed as matrix $F^L \in R^{N_i} \times M_i$ [34]. In order to visualize the feature information obtained from the input image in the different layers of the VGG network, a white noise image is optimized by continuous iteration to generate a new image that wants to match the feature information of the original image, the content reconstruction maps and the style reconstruction maps of the different layers are shown in Fig. 2.

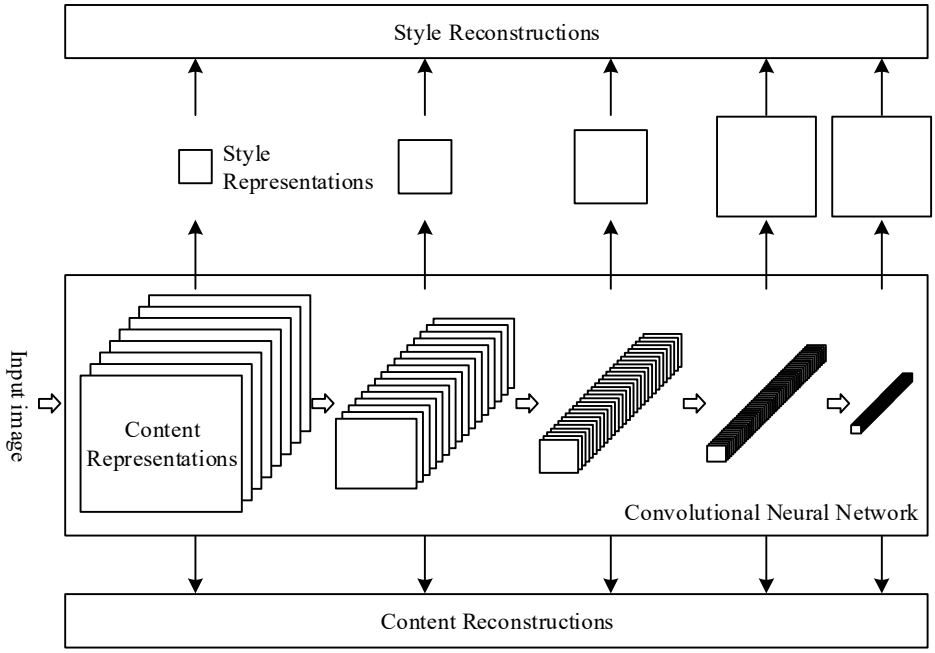


Fig. 2. Different layers of content reconstruction and style reconstruction

From the figure, it can be seen that the reconstruction result of the content image will become more abstract and blurred as the depth of the network deepens, and the content reconstruction result of the lower layers will be clearer, and the texture will be preserved maximally. Assuming that the input image is $\vec{x}$ and the reconstructed image is $\vec{p}$, its content reconstruction loss can be expressed as:

$$Loss_{content}\left(\vec{p},\vec{x},L\right)=\frac{1}{2}\sum_{i,j}\left(F_{ij}^{L}-p_{ij}^{L}\right)^{2} \tag{1}$$

where F_{ij}^{L} denotes the activation value of the i rd convolutional kernel in layer L at position j. Since the equation is derivable everywhere, the partial derivatives can be expressed by continuously optimizing the parameters through backpropagation:

$$\frac{\partial L_{content}}{\partial F_{ij}^{L}}=\begin{cases}\left(F^{L}-P^{L}\right)_{ij} & F_{ij}^{L}>0\\ 0 & F_{ij}^{L}<0\end{cases} \tag{2}$$

Since the feature information of the lower layers of the network can better reconstruct the content of the image, the feature information of the lower layers of the network can be used as a feature representation of the content of the image.

2.1.3. Stylistic characterization. The feature representation of style is different from the content feature representation that uses the values of the feature map directly. Style is something more abstract, style features as the network layers deepen, its reconstruction results can be more rich expression of the abstract style of the image, it does not only depend on the individual convolution kernel output feature information, but also the correlation of different convolution kernel convolution operation results of the composition, and thus expected to express the spatial extension of the style information, the correlation of these information has a transformation of the *Gram* matrix $G^{L}\in R^{N_L}\times N_L$ representation, where G_{ij}^{L} represents the The inner product between the i th feature map and the j th feature map of layer L is represented by the following equation:

$$G_{ij}^{L}=\sum_{k}F_{ik}^{L}F_{ik}^{L} \tag{3}$$

A multiscale representation of an input image can be obtained by covariance correlation containing feature information from multiple layers of the network, which captures only the texture information of the image and not its global information [35]. By reconstructing a given input image matching the stylistic representation, one can visualize the stylistic representation information extracted at different layers of the network. Again using the white noise image as a template, constant optimization iterations are used for loss reduction purpose by minimizing the mean square error between the Gram matrix of the input image and the Gram matrix of the generated image.

Suppose $\vec{a}$ is the input image and A^L is its stylized representation on layer L. $\vec{x}$ is the generated image and G^L is its stylized representation on layer L, then the overall stylistic loss on layer L can be expressed as:

$$E_L = \frac{1}{4N_L^2 M_L^2} \sum_{i,j} \left(G_{ij}^L - A_{ij}^L \right)^2 \tag{4}$$

Then the total style loss can be expressed as:

$$Loss_{style}(\vec{a}, \vec{x}) = \sum_{i,j}^{L} \omega_L E_L \tag{5}$$

where ω_L denotes the weight factor of each layer for the total loss.

2.1.4. Style transformation. In order to migrate the style representation of the style image to the content representation of the content image, we need to minimize the minimum distance between the white noise image of the content feature representation in the lower layers of the convolutional neural network and the style feature representation of the style image in the multiple layers of the convolutional neural network.

A style image a goes through the whole over network and its computation in each layer is stored. Content image p passes through the whole over network and its content expression p^L in layer 4 is stored, a random white noise image x passes through the whole over network and its style features G^L content features F^L are computed, and once in each layer of the style expression the mean square error between G^L and A^L is computed as a style loss metric while the mean square error between F^L and p^L is computed as a content loss metric, the global Total Loss Los_{stotal} is a linear connection between content loss and style loss, and its total loss formula is expressed as follows:

$$Loss_{total}\left(\vec{p}, \vec{a}, \vec{x}\right) = \alpha Loss_{content}\left(\vec{p}, \vec{x}\right) + \beta Loss_{style}\left(\vec{a}, \vec{x}\right) \tag{6}$$

Where α and β denote the content and style weight factors. Its inverse to the pixel value can be used as the direction of backpropagation, and the optimization method of gradient descent is used to continuously update the image x until the generated image matches both the style image features and the content image features.

Although this method can express the effect of style migration well, it requires a large number of iterative computations each time the image is generated, which reduces the practicality and application effectiveness of the algorithm and also requires a large amount of computational resources.

2.2. Generating Adversarial Networks

1) GAN. The source of Generative Adversarial Networks is actually the idea of Azero-sum game in GameTheory, and to extend this idea to deep learning, especially neural networks, it is in fact through the repeated confrontation and supervision of two neural networks (generative network and discriminative network) in the training process, so that the generative network learns the distribution of data that exists in the real world, and ultimately, it reaches the Nash equilibrium in game theory. Corresponding to the game theory is to achieve the Nash equilibrium. The generative adversarial network is formulated as follows:

$$\min_G \max_D V(G,D) = E_{x\sim P_{data}(x)} \log(D(x)) + E_{x\sim P_z(z)} \log(1-D(G(z))) \tag{7}$$

2) W-GAN. In generative adversarial network, the original generative adversarial network using JS scatter as the optimization function does not reasonably evaluate the distance between the generated false data distribution and the real data distribution. So the main difference between it and the original GAN is that, in terms of loss function, W-GAN uses Wasserstein distance instead of JS scatter for optimizing the generative adversarial network that needs to be trained, so as to alleviate the problem of unstable training of generative adversarial network.

For the true distribution p_r and model distribution p_θ, their 1st-Wasserstein distance is:

$$W^1(p_r, p_\theta) = \inf_{\gamma\sim\Gamma(p_r,p_\theta)} E_{(x,y)\sim\gamma}\left[\|x-y\|\right] \tag{8}$$

where $\Gamma(p_r, p_\theta)$ is the set of all possible joint distributions whose marginal distributions are p_r and p_θ.

When two distributions do not overlap, or overlap very little, the KL scatter between them is positive infinity. At this point the JS scatter is $\log 2$, a value that does not vary with the distance between the two distributions. However, the 1st-Wasserstein distance can still be used at this point to evaluate the distance between the two distributions that do not overlap.

3) Pix2Pix and CycleGAN. The most prominent revelation that Pix2Pix plays on the research of generative adversarial networks lies in the fact that a unified framework is proposed to solve the image transformation problem.

This framework establishes a direct mapping of inputs to outputs in the network, making the overall network framework appear very simple and elegant. However, doing this directly brings about one of the most significant problems in the image generation task, i.e., unclear quality of the generated images. Therefore, Pix2Pix introduces the content of conditional in CGAN in the loss, taking an image as a condition, which is specifically manifested in the network as LlLoss with increased output and labeling information, which is used to constrain the difference between the generated image and the real image. The final Loss function formula for the Pix2Pix network is as follows:

$$G^* = \arg\min_G \max_D L_{cGAN}(G,D) + \lambda L_{L1}(G) \tag{9}$$

CycleGAN, as the successor of Pix2Pix, its biggest contribution is to solve the previous network's input images must be pairs of shortcomings, its unique network architecture makes the network can be in any two sets of images in the style of mutual conversion. The CycleGAN structure is shown in Figure 3.

The goal accomplished by CycleGAN is to get a machine to learn the way to get from source data domain X to target data domain Y in the absence of paired data. It uses an Adversarial Loss Function to learn to map the generative network $G: X \to Y$, making it difficult for the discriminator to differentiate between the image $G(X)$ and the real image Y. Because of not being paired data, the mapping that the network is trying to fit in training is drastically limited, so CycleGAN, as depicted in the name, adds an exact opposite mapping $F: Y \to X$ to the mapping G of the generative network to artificially making the data into pairs, while adding a cyclic consistency loss to the original Loss function to ensure $F(G(X)) \approx X$ (and vice versa).

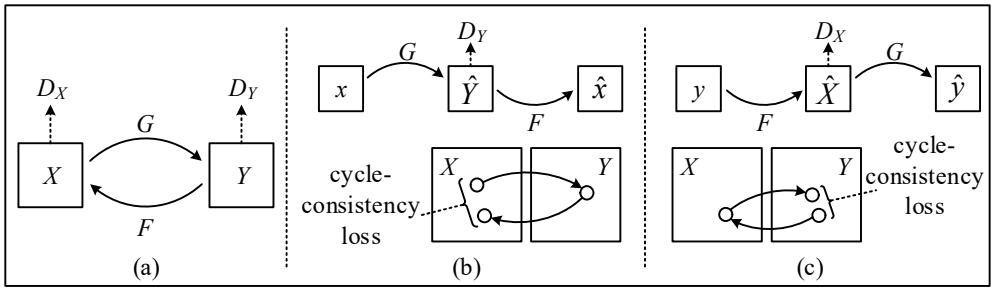


Fig. 3. CycleGAN chart

There are two most important points in the implementation of Cycle GAN. The first one is the double discriminator, two distributions X and Y, generators G and F are maps from X to Y and Y to X respectively, and two discriminators Dx and Dy can discriminate the converted images. The second point is cycle-consistency loss, i.e., coordination consistency loss, the reason for using other maps in the dataset to test the generator is to prevent G and F from overfitting, e.g., if you want to transform a picture of a puppy into Van Gogh's style in an experiment, if there is no cycle-consistencyloss, the generator may produce a real Van Gogh painting to fool the discriminator Dx and thus ignore the input puppy. The formula for cycle-consistency loss is as follows:

$$L(G,F,Dx,Dy) = L_{GAN}(G,Dy,X,Y) + L_{GAN}(F,Dx,Y,X) + \lambda L_{cyc}(G,F) \tag{10}$$

2.3. Image style transformation algorithm based on improved generative adversarial network

2.3.1. Overview of the model structure. The DMI-GAN model mainly considers the statistical laws of the data and the correlation between the features. The correlation refers to the distribution laws of the data of different styles of images and their correlation in different feature dimensions. The distribution pattern of these data and their correlations in different feature dimensions have a decisive impact on the training of the algorithm and the quality of the results. If one wants to convert an image into another style, there needs to be enough sample data to learn the differences and similarities between the two styles. Statistical patterns and correlations between features in these sample data can help the model to recognize which features are relevant to the styles and thus perform the style conversion operation better.

2.3.2. Generator network structure. The input and output of the generator network structure are three-dimensional tensors of size 256 × 256 × 3. The encoder employs a number of depth-separable convolutional combining layers, where m = 8, for a total of eight layers, and the final output is a one-dimensional vector. The size of the convolution kernel is 5 × 5 with a step size of 2, and downsampling is performed after each layer of convolution operation. After the encoder layer, the decoding is performed, using multiple reverse convolution Batch-Normalization-PReLU combination layers, repeating the 5×5 reverse convolution operation for a total of eight layers, which is the reverse process of the encoder, gradually recovering the image size by up-sampling, doubling the length and width of the image with each layer, and ultimately outputting an image of size 256×256×3. The model uses a combination of inverse convolution, batch normalization and PReLU activation functions as the eight layers of the decoder [36]. The network employs a left-right symmetric structure, where the main role of the convolutional operation on the left side is to perform feature extraction from the input image by means of different convolutional kernels. Based on this, the feature maps are normalized by the Batch Normalization layer, while the nonlinear transformations are performed by the ReLU activation function to enhance the expressive power of the model. In the design, the depth-separable convolutional layer, the Batch Normalization layer and the ReLU activation function are regarded as a

whole module and are denoted as one layer, which reduces the number of network layers, reduces the number of parameters, and improves the computational efficiency.

1) Depth separable convolution. In order to improve the training speed and reduce the amount of operations, the depth separable convolution generator first performs depth separable convolution operations on each channel, sets the number of output channels of each single channel to 1, and sequentially performs convolution operations on these channels to obtain N feature maps with channel 1. These N feature maps are then spliced to form a feature map with N channels. Finally, a point-by-point convolution of size 1 × 1 is used to perform feature fusion on the spliced feature maps, and the final result is output. Unlike grouped convolution, which performs only a single convolution and directly splices the output, depth-separable convolution performs two convolution operations and takes into account both the region and channel information of the image, and thus is able to more fully extract defect feature information. In contrast, the ordinary convolution used for grouped convolution is computationally high, leading to the problem of the model having an excessive amount of parameters.

Suppose the Size of the input feature map is $D_k \times D_k \times M$ and the Size of the convolution kernel is $D_F \times D_F \times M$ and the number of them is N. For a single ordinary convolution, its total computation can be expressed as equation (11):

$$D_k \times D_k \times D_F \times D_F \times M \tag{11}$$

If there is N convolution, the total computation of ordinary convolution can be expressed as equation (12):

$$D_k \times D_k \times D_F \times D_F \times M \times N \tag{12}$$

The total computation of the depth separable convolution can be expressed as equation (13):

$$D_k \times D_k \times D_F \times D_F \times M + N \times M \times D_k \times D_k \tag{13}$$

The ratio of the total computational effort of the deep separable convolution to the normal convolution can be expressed as equation (14):

$$\frac{1}{N} + \frac{1}{D_F^2} \tag{14}$$

2) Cross-layer connectivity generator. In order to realize the sharing of a large amount of information between inputs and outputs in image conversion, as well as to transfer the information directly in the network, this paper introduces the cross-layer connectivity mechanism to design the generator model G. The cross-layer connectivity mechanism, by allowing the current layer to inherit the feature maps of the previous layer, can enable the network to directly utilize the information between the inputs and outputs for conversion processing.

2.3.3. Discriminator network structure:

1) Full domain convolution. In order to ensure the quality of the generated image, the accuracy of the discriminator directly affects the image generation. For this reason, this paper uses full domain convolution to evaluate the performance of the generator to generate images. The determination network uses the whole image input to the discriminator for determination, and the network structure is designed through the discriminator network to achieve this purpose. The final output of the discriminator is a one dimensional vector [37]. To realize this process, the network uses a multilayer downsampling technique and a discriminator layer is added to the last layer.

2) Multi-scale discriminator. The use of multi-scale discriminators avoids overfitting as well as reduces the memory footprint compared to operations that increase the depth of the network or expand the

convolutional kernel. In this paper, we use three different discriminators D_1, D_2 and D_3 respectively, including several downsampling layers and an output classification layer.

2.3.4. Loss function design and analysis. In this paper, four loss functions are used: the WGAN loss, the L_1 loss, the mutual information loss, and the FM loss.The loss function of WGAN-div is based on the method of approximation of Wasserstein distances between probability distributions, and the upper bound of the distances is obtained by calculating the covariance of the segmented linear functions of the respective negative integrals, which is used as the discriminator loss function.The WGAN-div min-max optimization The specific definition of the problem can be expressed as Eq. (15), where z denotes random noise, x denotes real data, $\hat{x}$ denotes sampling of a linear combination of real and generated data, D denotes the discriminator, and G denotes the generator. In the original GAN D reaches the optimal state before G, which causes G the optimization to slow down or stagnate, and the generative distribution P_g is difficult to fit to the real distribution P_{data}, whereas the WGAN-div makes the G and D updates smoother by introducing the Wasserstein scatter.

$$loss_{WGAN} = \sigma^{E}_{G(z)\sim P_x}\left[D(G(z))\right] - x_{x\sim P_r}[D(x)] - k^{E}_{\hat{x}-P_x}\left[\left\|\nabla_x D(\hat{x})\right\|^{p}\right] \tag{15}$$

Thus the WGAN optimization problem can be expressed as Eq. (16):

$$\min_{G}\max_{D_1,D_2,D_3}\sum_{k=1,23} los_{WGAN}\left(D_k, G\right) \tag{16}$$

The task of generator G is to learn the mapping from input image x to target image y, i.e., $G: x \to y$. By optimizing the parameters of generator G, the generated image is made to be as close as possible to the real target image such that it is difficult to distinguish the real from the fake. The task of the discriminator D, on the other hand, is to distinguish the generated image from the real image as accurately as possible. The performance and effectiveness of this framework can be further improved by pix2pix by performing network optimization on Eq. (17).

$$\arg\min_G \max_D \left(loss_{GAN} + \lambda loss_{L1}\right) \tag{17}$$

where $loss_{GAN}$ is the GAN loss function and $loss_{L1}$ is the L_1 loss. λ is the adjustable parameter. The $loss_{GAN}$ definition can be expressed as equation (18):

$$loss_{GAN} = E_{y\sim pdata(y)}\left[\log D(y)\right] + E_{x\sim pdata(x)}\left[\log\left(1 - D\left(G(x)\right)\right)\right] \tag{18}$$

$loss_{L1}$ definition can be expressed as equation (19):

$$\text{loss}_{L1} = E_{x,-1-p\ \text{pata}(x,y)}\left[\left\|y - G(x)\right\|_1\right] \tag{19}$$

In order to enhance the correlation between the generated and target images, this paper introduces the FM loss from the pix2pixHD algorithm to further limit the performance of the discriminator. This loss function calculates the disparity by extracting and summarizing the features output from each layer of the discriminator, learning and matching the shared features between the real and generated images. In this case, the features extracted by layer i discriminator D_k are denoted as $D_k^{(i)}$. Subsequently, the FM loss $L_{FM}\left(G, D_k\right)$ is computed according to Eq. (20).

$$loss_{Fy}\left(G, D_k\right) = E_{(x,y)}\sum_{k=1}^{T}\frac{1}{N}\left[\left\|D_k^{(i)}(x,y) - D_k^{(i)}(x, G(x))\right\|_1\right] \tag{20}$$

where T is the total number of discriminator layers, N_i indicates the number of elements in each layer of the layer.

$I(X;Y)$ measures the random variable Y about the random variable X and Y overlap part of the data distribution, $I(X;Y)$ is the mutual information between X and Y, by observing the mutual information of the implied coding and generating distributions, so that the implied coding and generating distributions mutual information regularization, mutual information is essentially the difference between the two information entropy, so that the generating distributions are more quickly leaning to the real distribution, mutual information specific definition can be expressed as equation (21):

$$I(X;Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) \tag{21}$$

$H(X \mid Y)$ and $H(Y \mid X)$ denote the entropy of the posterior distribution, $H(X)$ and $H(X)$ denote the entropy of the prior distribution, assuming that X and Y are independent of each other, then $I(X;Y)$, and vice versa X and Y can be related by an invertible law, which results in the maximum mutual information. Mutual information combined with Wasserstein scatter, the introduction of mutual information discrimination results as a regulatory parameter, the closer the hidden variables are to the real image the higher the weight can be obtained, λ_1 and λ_2 with the increase in the learning rate increases, combined with the formula to obtain the DMI loss function, the loss function can be expressed as equation (22):

$$\begin{aligned} loss_{DMI}(D,G,c,z) = \arg\min_{G}\max_{D}\Big(loss_{WGAN}(D_k,G) \\ +\lambda_1 D(G(z,c))I(c;G(z,c)) + \lambda_2 loss_{L1} + \lambda_3 \sum_{k=1,23} L_{FM}(D_k,G)\Big) \end{aligned} \tag{22}$$

3. Results and discussion

3.1. Evaluation of the effect of image style transformation in animation design

3.1.1. Introduction to evaluation indicators. This section focuses on using the objective MS-SSIM evaluation metric with the subjective Mean Opinion Score (MOS) as an evaluation criterion for the quality of image generation in animation design.MS-SSIM is a multiscale structural similarity metric, which is a multiscale version of the SSIM metric, with scores ranging from 0 to 1. The larger the result, the greater the similarity between the two images. The process of calculating the MS-SSIM score involves scaling the two input images from small to large and calculating the similarity between the multiscale images. First, the images are scaled according to the image scaling factor M. The width and height are reduced by a factor of 21M-. When M=1, it indicates the original image size. When M=2, it indicates that the original image is reduced by half, and so on. After that, same as SSIM, the brightness (mean), color contrast (variance) and structure (covariance) of the image are calculated, and the formulas for the three features are shown below.

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{23}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{24}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{25}$$

Where $l(x,y)$ represents the luminance feature of the image, $c(x,y)$ represents the color contrast feature of the image, and $s(x,y)$ represents the structural feature of the image. μ_x and μ_y denote the mean of x and y respectively, σ_x and σ_y denote the variance of x and y respectively, σ_{xy} denotes the covariance of x and y, and C_1, C_2, and C_3 are constants. Finally the MS-SSIM

scores under multi-scale images are calculated by cumulatively multiplying the variance and covariance of the two images in each scale as in the following equation:

$$MS-SSIM(x,y)=\left[l_M(x,y)\right]^{\alpha_M}\cdot\prod_{j=1}^{M}\left[c_j(x,y)\right]^{\beta_j}\left[s_j(x,y)\right]^{\gamma_j} \tag{26}$$

where $c_j(x,y)$ and $s_j(x,y)$ represent the contrast as well as structural information of the image at the jrd scale, respectively, and $l_M(x,y)$ represents the luminance information at the Mth scale.

Mean Subjective Score Experiment (MOS) is a visual evaluation experiment based entirely on human subjectivity, and the whole experiment is conducted by 10 researchers in the direction of image quality related to the subjective evaluation of the results of each method. Before implementing the visual evaluation experiments in this section, 50 images in X-domain and Y-domain from each of the four datasets are first randomly selected as the test set, and input DiscoGAN, CycleGAN, and StarGAN with the methods in this paper to generate the corresponding result images. The target domain images are then shown to the testers to ensure that the 10 testers have some understanding of the basic target domain style, after which the generated results are shown one by one in sequence, with the testers not knowing which method the presented images belong to, and scoring each image. The scores range from 0 to 5, where 0 indicates the worst quality and 5 the best quality (closest to the target domain image style while preserving the source domain image features). Finally, each method is summed and averaged based on the image scores in each domain of the dataset to obtain the subjective image quality score of each method on each dataset. The specific formula for calculating the score is shown below.

$$MOS=\frac{\sum_{i=1}^{k}N_iS_i}{\sum_{i=1}^{k}N_i} \tag{27}$$

In the above formula, k denotes the rating of the image to be tested, S_i denotes the evaluation score corresponding to each rating, and N_i denotes the total number of people with each evaluation score.

3.1.2. Comparison and analysis of evaluation indicator results. Comparison of MS-SSIM values for each method is shown in Table 1. Here in this section the results are evaluated using the referenced evaluation metric MS-SSIM, which calculates the structural consistency of the input image from a multi-scale perspective. In order to verify the correlation between the generated results and the styles of the source and target domains, this section uses interpolation metrics to compute the evaluation metrics in a comprehensive way. From the results of MS-SSIM metrics on the four datasets, it can be concluded that the method in this chapter has an average improvement of 8.53%, 14.34%, and 22.88% over CycleGAN, StarGAN, and DiscoGAN, respectively, and the method in this paper has the best metrics results. The quantitative results of Facades dataset are analyzed here. Since the MS-SSIM metric calculates the similarity from the brightness, color and structure of the image, and the results of training and validation using Facades dataset, only the structural feature locations have some similarity between the generated result map and the input map, while the overall texture features and color information are not similar, so the The results of the metrics are low.

Table 1. Comparison of MS-SSIM values of all parties

Data set	Style direction	DiscoGAN	StarGAN	CycleGAN	Ours
Facades	Source domain->Target domain	0.2304	0.2368	0.212	0.2714

	Target domain-> Source domain	0.2716	0.2092	0.2555	0.2754
Monet2photo	Source domain->Target domain	0.5327	0.7798	0.7455	0.8423
	Target domain-> Source domain	0.5744	0.8554	0.8722	0.9451
Apple2orange	Source domain->Target domain	0.7254	0.7852	0.7854	0.8123
	Target domain-> Source domain	0.6986	0.7964	0.7919	0.821
Summer2winter	Source domain->Target domain	0.6729	0.8135	0.9582	0.9757
	Target domain-> Source domain	0.6395	0.7834	0.9469	0.9676

Meanwhile, this section uses the subjective opinion score metric, i.e., Mean Opinion Score (MOS), to support the validity of the experiment, which is the most representative subjective evaluation method of image quality, and judges the image quality by the evaluation score of the observer. The Mean Opinion Score (MOS) of each method is shown in Table 2, and the MOS value of this paper's method is improved by about 12.67% and 52.93% on average compared to CycleGAN and StarGAN, respectively. The final subjective mean opinion score proves that the method of this paper effectively improves the quality and visualization of image style conversion.

Table 2. Average opinion score(MOS)

Data set	Style direction	DiscoGAN	StarGAN	CycleGAN	Ours
Facades	Source domain->Target domain	0.78	3.02	4.08	4.59
	Target domain-> Source domain	0.84	2.96	4.2	4.72
Monet2photo	Source domain->Target domain	0.78	3.1	4.06	4.54
	Target domain-> Source domain	0.79	3	4.13	4.5
Apple2orange	Source domain->Target domain	0.81	3.03	4.09	4.54
	Target domain-> Source domain	0.87	2.9	4.02	4.52
Summer2winter	Source domain->Target domain	0.76	2.97	3.95	4.68
	Target domain-> Source domain	0.86	3.08	4.12	4.69

3.1.3. Analysis of evaluation indicator results. The image style migration task is a task that changes the overall stylistic characteristics of an input image, and evaluating the degree of stylization of its experimental results requires a highly subjective human visual sense. Therefore, in this chapter, the degree of image stylization is evaluated using subjective evaluation, and the first evaluation metric chosen is the mean opinion score (MOS). Specifically, the experiment begins by assembling 10 testers, each of whom rates the degree of stylization of the images they see based on their own opinion of the degree of stylization of the images they see, with rating scores ranging from 0 to 5, where a score of 0

means that the degree of stylization is very low or the image is of poor quality, whereas a score of 5 means that the image is highly stylized and stylistically similar to a given stylized image. Before the experiment starts, the input content image and the stylized image are first shown to the testers at the same time, so that the testers can be familiar with the overall feature distribution and the stylized effect that needs to be converted, and after that the stylized result images of the different methods are shown to the testers in turn, and the testers score the result images of each method according to their own subjective feelings. In this chapter, five sets of result images are selected for evaluation, and finally the scores of the result images of each method are summarized and evaluated. The experimental results show that the mean opinion score can effectively evaluate the degree of image stylization. The Mean Opinion Score (MOS) of each method is shown in Table 3. After averaging the testers' scores for each group of image results, the average score of this paper's method is the highest at 4.72, which shows that the degree of stylization of the results of this paper's method is most in line with the subjective feelings of human beings and achieves the best visual effect.

Table 3. Average opinion score(MOS)

method	Style Swap	Artflow	AdaIN	EFDM	Ours
Group 1	1.66	0.63	4.49	4.76	4.85
Group 2	1.5	0.38	4.73	4.75	4.84
Group 3	1.34	0.48	4.28	4.42	4.64
Group 4	1.8	0.48	4.49	4.64	4.69
Group 5	3.47	0.53	4.6	4.61	4.6

After that, this paper takes another subjective evaluation experiment, in this round of experiments, remove the five groups of image results used in the above experiments, and re-select 5 content images and 15 style images randomly and group them, the ratio of the number of content images to the number of style images in each group is 1:3, and select one of the styles in the style images to be trained with the content images, and get the stylized result images to become a group with the content images and the style images. The testers will make subjective judgment on the displayed stylized result images to determine which style of the result images belongs to which style among the three style maps in each group, and select the algorithm with the best results. Finally, the results of each method are judged to get the accuracy rate of image classification and algorithm preference, and the judgment accuracy rate and algorithm preference rate of each comparison method are shown in Table 4. As can be seen from the results, this paper's method in the degree of image stylization is closer to the input style map overall color, style characteristics, the quality of the resultant map obtained from the experiment is also the most in line with the testers' favor, the testers of this paper's algorithm's preference rate reached 55%.

Table 4. The accuracy of the comparison method and the rate of algorithm preference

Method	Accuracy of judgment	Algorithm preference
Style Swap	50%	0%
Artflow	10%	0%
AdaIN	88%	25%
EFDM	90%	35%
Ours	95%	55%

Finally, the length of training time required for the experimentation of several algorithms was compared with the length of testing time for each test image, and the training time and testing time for each compared method are shown in Table 5. Where the training time is the total time spent from the beginning of the network training time to the end of the training, and the testing time is the total time spent on all test images divided by the number of test images. From the table, it can be seen that the method in this paper increases the training and testing time compared to the benchmark method, but spending less time can get better quality, and to a certain extent, it also improves the network's generalization ability for image feature extraction and style learning, which verifies the effectiveness of the proposed method in this chapter.

Table 5. Training and testing times for different comparison methods

Method	Style Swap	Artflow	AdaIN	EFDM	Ours
Training time(h)	15.9	10.2	12.5	12.8	13.7
Test time(ms)	25	22.6	4.2	4.2	4.9

3.2. Algorithm Performance Experiments

3.2.1. Comparison of data under different experimental indicators. The PSNR and SSIM of each species under different models are shown in Table 6 and Table 7. The data in the table demonstrates the PSNR and SSIM values of different models after inputting different categories of image migration, from the data in the table, it can be seen that the images migrated based on the method of this paper have the highest PSNR and SSIM values, and the four methods have the highest PSNR and SSIM values on human images, and compared with the Cartoon GAN model, the migrated based on this paper's method images its PSNR value is improved by 1.89% overall and SSIM value is improved by 3.33% overall.

Table 6. The PSNR of each species in different models

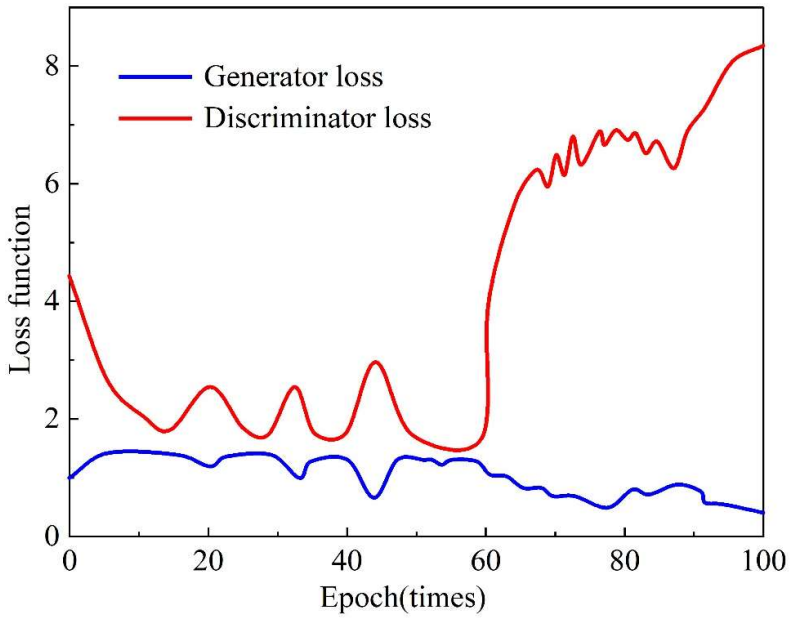
Species	Cartoongan	Cyclegan	Stylegan	Ours
Architecture	14.7361	13.8319	14.9476	15.0628
Plants	16.7452	15.898	16.9287	17.2904
Car	15.8714	14.9906	16.1042	16.0687
Man	18.1029	17.4985	18.2208	18.2538

Table 7. SSIM of each species under different models

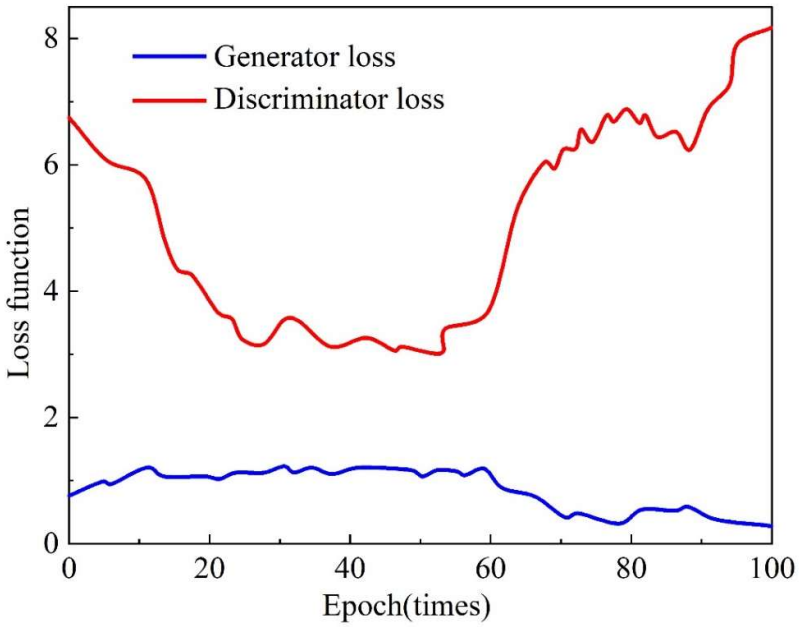
Species	Cartoongan	Cyclegan	Stylegan	Ours
Architecture	0.5813	0.5916	0.6037	0.6128
Plants	0.7506	0.627	0.688	0.7661
Car	0.7419	0.5781	0.6317	0.7657
Man	0.7682	0.6368	0.8045	0.7885

3.2.2. Qualitative experimental results of image style migration in animation design. The generator and discriminator loss functions are shown in Figure 4. Fig. 4(a) demonstrates that the generator loss function rises before the first 5 trainings during the training process, at which time the discriminator loss function decreases, and during the 10th to 50th trainings, the generator has three trough values, and the corresponding discriminator reaches three peak values. The generator loss function decreases after 60 times of training, at which time the discriminator loss function rises, Fig. 4(b) demonstrates that the generator loss function rises before 20 times of training, at which time the discriminator loss function decreases, and the generator loss function decreases after 60 times of training, at which time the discriminator loss function rises. Fig. 4(c) demonstrates that the generator loss function in the training process decreases before 10 training sessions, at which time the discriminator loss function increases, the generator loss function increases after 15 training sessions, at which time the discriminator loss function decreases, Fig. 4(d) demonstrates that the generator loss function in the training process increases before the first 2 training sessions, at which time the discriminator loss function decreases, and the generator loss function in the training process between 2 and 100 training sessions shows a decreasing trend, which contains multiple spikes peaks, at this time the discriminator loss function shows a rising trend, corresponding to the generator's multiple spikes appeared in the corresponding trough value, which can be seen that the model's generator and discriminator against each other, and the model training is stable.

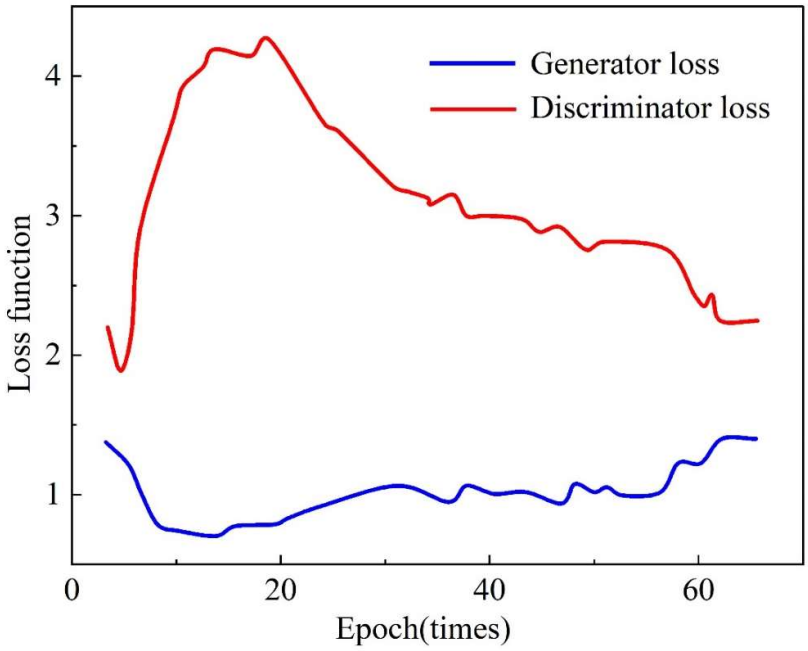
The methods in this chapter have similar properties with CycleGAN and CartoonGAN. However, the training time of the method in this chapter is shorter than CycleGAN and similar to that of CartoonGAN.CycleGAN and CartoonGAN take about 2735s and 1320s, respectively, while the method in this chapter takes about 1770s.CycleGAN spends more time on bidirectional training and the method in this chapter only consumes 180s more than the CartoonGAN consumes 180s more. To get better results, increase the training time a little. The method in this paper can learn the art style better and handle the salient regions of the animation design kind effectively.



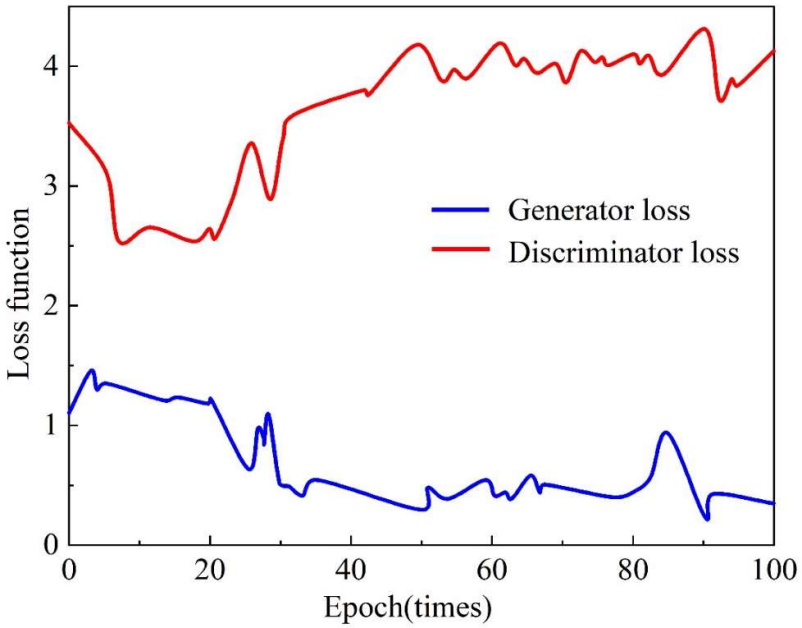
(a) Generator loss function rises before 5 training



(b) Generator loss function rises before 20 training



(c) Generator loss functions fall before 10 training



(d) Generator loss function rises before 2 training

Fig. 4. Loss functions for generators and discriminators

4. Conclusion

The traditional style migration method has obvious effect on image style migration in animation design, but with the development of artificial intelligence technology, the traditional style migration method can not meet people's needs. Based on this, this paper proposes an image conversion algorithm based on deep convolutional neural network to improve the generation of adversarial network model, to realize the conversion of animation design image style.

Through the experimental results of the effect of comparison, found that the method in this paper in the degree of image stylization closer to the input style map of the overall color, style characteristics,

image quality is also the most in line with the favor of the testers, testers on the use of this paper's algorithm for the style of the image conversion of the preference rate reached 55%. The qualitative and quantitative results show that the method proposed in this paper effectively improves the quality of image style migration generation.

Compared with the Cartoon GAN model, the PSNR and SSIM values of the migrated images by this paper's method are improved by 1.89% and 3.33% overall, respectively. Experimentally, it is proved that the use of this paper's method for animation design image style migration has a very good style conversion effect.

References

[1] Kong, F., Pu, Y., Lee, I., Nie, R., Zhao, Z., Xu, D., ... & Liang, H. (2023). Unpaired Artistic Portrait Style Transfer via Asymmetric Double-Stream GAN. IEEE Transactions on Neural Networks and Learning Systems.

[2] Khowaja, S. A., Nkenyereye, L., Mujtaba, G., Lee, I. H., Fortino, G., & Dev, K. (2024). FISTNet: FusIon of STyle-path generative Networks for facial style transfer. Information Fusion, 112, 102572.

[3] Cui, J., Liu, Y. Q., Lu, H. J., Cai, Q. Q., Tang, M. X., Qi, M., & Gu, Z. Y. (2021). PortraitNET: Photo-realistic portrait cartoon style transfer with self-supervised semantic supervision. Neurocomputing, 465, 114-127.

[4] Yang, S., Jiang, L., Liu, Z., & Loy, C. C. (2022). Vtoonify: Controllable high-resolution portrait video style transfer. ACM Transactions on Graphics (TOG), 41(6), 1-15.

[5] Wang, L. (2022). Cartoon - Style Image Rendering Transfer Based on Neural Networks. Computational Intelligence and Neuroscience, 2022(1), 2958338.

[6] Karmakar, A. (2021). Investigation Of Artistic Styles For Effective Storytelling In Animation. NVEO-NATURAL VOLATILES & ESSENTIAL OILS Journal| NVEO, 200-209.

[7] Lyu, Y., Lin, C. L., Lin, P. H., & Lin, R. (2021). The cognition of audience to artistic style transfer. Applied Sciences, 11(7), 3290.

[8] Chen, J., Liu, G., & Chen, X. (2020). AnimeGAN: A novel lightweight GAN for photo animation. In Artificial Intelligence Algorithms and Applications: 11th International Symposium, ISICA 2019, Guangzhou, China, November 16–17, 2019, Revised Selected Papers 11 (pp. 242-256). Springer Singapore.

[9] Lu, Z., Zhou, Y., & Chen, A. (2024). Enhancing Photo Animation: Augmented Stylistic Modules and Prior Knowledge Integration. In Proceedings of the Asian Conference on Computer Vision (pp. 1470-1485).

[10] Fišer, J., Jamriška, O., Simons, D., Shechtman, E., Lu, J., Asente, P., ... & Sýkora, D. (2017). Example-based synthesis of stylized facial animations. ACM Transactions on Graphics (TOG), 36(4), 1-11.

[11] Dushkoff, M., McLaughlin, R., & Ptucha, R. (2016, December). A temporally coherent neural algorithm for artistic style transfer. In 2016 23rd International Conference on Pattern Recognition (ICPR) (pp. 3288-3293). IEEE.

[12] Zheng, J. (2022). Design and Application of Intelligent Processing Technology for Animation Images Based on Deep Learning. Mobile Information Systems, 2022(1), 9438086.

[13] Han, X., Wu, Y., & Wan, R. (2023). A method for style transfer from artistic images based on depth extraction generative adversarial network. Applied Sciences, 13(2), 867.

[14] Ou, Y., & Xu, J. (2024). WDANet: exploring stylized animation via diffusion model for woodcut-style design. Authorea Preprints.

[15] Li, H. A., Wang, L., & Liu, J. (2024). A review of deep learning-based image style transfer research. The Imaging Science Journal, 1-23.

[16] Ho, S. T., Nguyen, V. T., & Ngo, T. D. (2020, October). Interpolation based anime face style transfer. In 2020

International Conference on Multimedia Analysis and Pattern Recognition (MAPR) (pp. 1-6). IEEE.

[17] Chen, Y., Lai, Y. K., & Liu, Y. J. (2018). Cartoongan: Generative adversarial networks for photo cartoonization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 9465-9474).

[18] Ren, S., & Sheng, Y. (2022, February). Image style transfer using deep learning methods. In 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA) (pp. 1190-1195). IEEE.

[19] Su, X., & Kim, H. G. (2024). Automatic Stylized Action Generation in Animation Using Deep Learning. IEEE Access.

[20] Way, D. L., Chang, W. C., & Shih, Z. C. (2019, November). Deep learning for anime style transfer. In Proceedings of the 2019 3rd international conference on advances in image processing (pp. 139-143).

[21] Shi, J. (2023). Artificial Intelligence for Art Creation with Image Style. Highlights in Science, Engineering and Technology, 44, 67-74.

[22] He, J. (2024). Exploring style transfer algorithms in Animation: Enhancing visual. Entertainment Computing, 49, 100625.

[23] Kim, S. (2023). Reimagining Animation Making through Style Transfer. In SIGGRAPH Asia 2023 Art Papers (pp. 1-6).

[24] Liu, D. S. M., & Tu, N. (2021). Video cloning for paintings via artistic style transfer. Signal, Image and Video Processing, 15(1), 111-119.

[25] Wang, R. (2019). Research on image generation and style transfer algorithm based on deep learning. Open Journal of Applied Sciences, 9(08), 661.

[26] Fan, X. (2021, April). Application of style transfer algorithm in the design of animation pattern special effect. In Journal of Physics: Conference Series (Vol. 1852, No. 2, p. 022054). IOP Publishing.

[27] Li, S. (2023). Application of artificial intelligence-based style transfer algorithm in animation special effects design. Open Computer Science, 13(1), 20220255.

[28] Ruder, M., Dosovitskiy, A., & Brox, T. (2018). Artistic style transfer for videos and spherical images. International Journal of Computer Vision, 126(11), 1199-1219.

[29] Yang, S., Wang, Z., & Liu, J. (2021). Shape-matching GAN++: Scale controllable dynamic artistic text style transfer. IEEE Transactions on Pattern Analysis and Machine Intelligence, 44(7), 3807-3820.

[30] Zhang, L., Ji, Y., Lin, X., & Liu, C. (2017, November). Style transfer for anime sketches with enhanced residual u-net and auxiliary classifier gan. In 2017 4th IAPR Asian conference on pattern recognition (ACPR) (pp. 506-511). IEEE.

[31] Song, G., Luo, L., Liu, J., Ma, W. C., Lai, C., Zheng, C., & Cham, T. J. (2021). Agilegan: stylizing portraits by inversion-consistent transfer learning. ACM Transactions on Graphics (TOG), 40(4), 1-13.

[32] Zhang, Z., Sun, J., Chen, J., Zhao, L., Ji, B., Lan, Z., ... & Xu, D. (2023). Caster: Cartoon style transfer via dynamic cartoon style casting. Neurocomputing, 556, 126654.

[33] Xi, H., & Zhi, H. (2023, January). The Application of Ink Animation Based on Artificial Neural Network Style Transfer. In International Conference on Innovative Computing (pp. 499-504). Singapore: Springer Nature Singapore.

[34] Huimin Han,Bouba oumarou Aboubakar,Mughair Bhatti,Bandeh Ali Talpur,Yasser A. Ali,Muna Al Razgan & Yazeed Yasid Ghadi. (2024). Optimizing image captioning: The effectiveness of vision transformers and VGG networks for remote sensing. Big Data Research100477-100477.

[35] Venu Allapakam & Yepuganti Karuna. (2024). An ensemble deep learning model for medical image fusion

with Siamese neural networks and VGG-19.. PloS one(10),e0309651.

[36] Gurusubramani S & Latha B. (2024). Deep Convolutional Generative Adversarial Network for Improved Cardiac Image Classification in Heart Disease Diagnosis.. Journal of imaging informatics in medicine.

[37] M. Diviya,A. Karmel,R. Utthirakumari & M. Subramanian. (2024). Enhancing low-illumination imagery using a Deep Convolutional Generative Adversarial Network with weight regularization (DCGAN-WR) and Zero-Reference Deep Curve Estimation (DCE).Discover Computing(1),48-48.