IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 5, 2020, accepted July 13, 2020, date of publication July 16, 2020, date of current version July 29, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009470

Detail-Preserving CycleGAN-AdaIN Framework for Image-to-Ink Painting Translation

FENGQUAN ZHANG, HUAMING GAO, AND YUPING LAI, (Member, IEEE)
School of Information Science and Technology, North China University of Technology, Beijing 100144, China

Corresponding author: Fengquan Zhang (fqzhang@ncut.edu.cn)

This work was supported in part by the Humanities and Social Sciences Fund of the Ministry of Education under Grant 19YJC760150, in part by the Beijing Social Science Foundation under Grant 18YTC038, in part by the Beijing Natural Science Foundation under Grant 4182018, Grant 4154067, and Grant 4194076, in part by the National Natural Science Foundation under Grant 61402016, in part by the open funding project of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, under Grant VRLAB2020B10, in part by the Beijing Youth Talent Foundation under Grant 2016000026833ZK09, and in part by the NCUT Foundation under Grant XN018001.

ABSTRACT Image translation tasks based on generative models have become an important research area, such as the general framework for unsupervised image translation-CycleGAN (Cycle-Consistent Generative Adversarial Networks). A typical advantage of CycleGAN is that it can realize the training of two image sets without pairing, but there are still some problems in the preservation of semantic information and the learning of specific features. In this paper, we propose the CycleGAN-AdaIN framework based on the CycleGAN model, which can translate real photos into Chinese ink paintings. In order to retain the content of the image completely, we use one cycle consistency loss to replace two in the structure of the model. To learn the style information of the ink painting, we introduce an AdaIN (Adaptive Instance Normalization) module before the decoding process of the generation network. In addition, to correct the details of the generated image, we add the MS-SSIM (Multi-Scale-Structural Similarity Index) loss in the reconstruction loss to generate a higher quality image. Compared with the existing methods in FID, Kernel MMD, PSNR and SSIM, the experiment results show that our method can accomplish the task of transferring real photos to ink paintings and get better performance than the baseline model.

INDEX TERMS Chinese ink painting, CycleGAN, detail-preserving, style transfer, AdaIN.

I. INTRODUCTION

In recent years, more and more researches have been devoted to the digital protection of traditional art. As a traditional Chinese painting, Chinese ink painting (paintings made by mixing water and ink into different shades of ink) has not only formed many styles and factions, but also has many techniques and great research value after long-term reform and development. Generally, in computer simulation, we extract typical artistic effects that can reflect its characteristics, such as voids, brush strokes and ink diffusion effects, and analyze them to realize their digital definition. Studying and realizing the digital definition and simulation of the traditional painting art are not only helpful to the training of painters' painting skills, but also beneficial to the inheritance and protection of traditional art.

The associate editor coordinating the review of this manuscript and approving it for publication was Inês Domingues.

There have been a lot of works for the study of ink painting. Generally speaking, these methods can be roughly divided into three categories: physical modeling based simulation methods, non-physical based methods, and deep learning based methods. Although methods based on physical modeling perform well, they are computationally complex and inefficient, and it is difficult for untrained users to draw ideal results. Non-photorealistic rendering methods deal with inputs of two-dimensional images, and take a single image with a specific style as the output, however, these methods process the images simply and roughly, and they are limited to a single style, which makes it difficult to generalized to other styles. With the rise of deep learning, the high-level features of images have been effectively used. The image style transfer methods based on convolutional neural network (CNN) can quickly extract the stylized features and content features of the image, and has gradually become the mainstream technology in the field of image style transfer.

In this paper, inspired by a typical unsupervised image translation framework, CycleGAN [1], we implement the image translation from real photos to ink paintings. Only two datasets with different styles are needed to learn the characteristics of the image and then realize the style transfer process. First of all, we found that CycleGAN has a disadvantage in retaining the content of the image, and compared to using two cycle consistency losses to constrain the quality of the generated image, the network structure using only forward one can make the generated image have more complete stroke information and more voids effect while saving a lot of training time. Secondly, although the structure of using only forward cycle consistency loss can better retain the content of an image, the generated image lacks the sense of space, therefore we introduce the AdaIN [2] module to learn the style information of ink painting before the decoding process of the generated network, so as to ensure that some details of the generated image are improved while generating more natural ink painting. Finally, we add the MS-SSIM [3] loss to the reconstruction loss to strengthen the constraints of the generative network and generate higher quality images. Fig.1 shows the pipeline of our proposed framework. Briefly, the contributions of this paper are as follows:

(1) We propose an image translation framework of CycleGAN-AdaIN from real photos to ink paintings, which uses one cycle (X2Y2X) instead of two (X2Y2X + Y2X2Y) of CycleGAN. Compared with the baseline model, our method can not only better retain the content of the generated image, but also save a lot of training time.
(2) An additional AdaIN module is designed to learn the style information of ink painting. We add it before the decoding process of the generator, which can not only keep the spatial structure of the generated image while retaining its content, but also produce more realistic ink diffusion effect.
(3) In order to correct the details of the generated image, we combine cycle consistency loss and MS-SSIM loss to strengthen the constraints on the generative network, so as to generate more realistic and natural ink painting.

The rest of this paper is structured as follows: The related works are introduced in Section 2. Section 3 describes the methods proposed in this paper in detail. Section 4 shows the experiment results and evaluations. Section 5 concludes this paper and describes our future work.

II. RELEATED WORK

A. INK PAINTING

Traditional ink painting simulation research is mostly based on physical modeling and non-photorealistic rendering. The physical modeling methods are mainly completed by physical analysis of the brush, paper, and ink diffusion effects, etc., and then establishing suitable models, while the non-physical simulation methods mainly refer to the non-photorealistic graph generation algorithms. Wang *et al.* [4] presented a real-time rendering method based on the GPU programmable pipeline for rendering the 3D scene in the ink-wash painting style. They used an ink dispersion model which was defined by referencing the theory of porous media to simulate the dispersion of ink. Gua *et al.* [5] proposed a technique for using geometric buffers to render 3D ink paintings in real time. According to the characteristics of hand-drawn ink paintings, contour lines and coloring area in the ink paintings were stylized by rendering 3D models to 2D texture images. At present, there are few studies on ink paintings based on deep learning. Luo *et al.* [6] proposed a multimodal fusion framework and system to generate traditional Chinese paintings. By selecting the appropriate existing networks for different elements in the traditional works, these networks and elements are integrated to create a complete new painting finally. He *et al.* [7] put forward ChipGAN, which mathematically defined and implemented the voids, brush stroke and ink-diffusion characteristics of ink painting. Zhou *et al.* [8] produced an accelerated version of the transfer for Chinese traditional ink painting style to improve the calculation speed and quality by reducing the size of the rendering network.

B. GENERATIVE ADVERSARIAL NETWORKS

Since Goodfellow *et al.* [9] proposed GAN (Generative Adversarial Network), it has received more and more attentions from academia and industry. GAN regards the generation problem as the confrontation and game between the discriminator and the generator. The generator generates synthetic data from the given noise, and the discriminator distinguishes the generated data from the real data. Since the introduction of GAN, many variants have been produced. LSGAN was proposed in [10], and the loss function of least squares was used to replace the original loss function. WGAN [11], [12] and WGAN-GP [13] stabilized the training of GAN and improved its convergence speed. Radford *et al.* [14] applied a convolutional network to GAN, and provided a good network topology for its training. In order to solve the problem that GAN is too free, a kind of GAN with conditional constraints was proposed in [15], in which a conditional variable was introduced into the modeling of the generator and discriminator to guide the data generation process. As a generative model, GAN is widely used in data generation, the most common one is image generation [16]–[18]. There are also many applications in other fields such as style transfer [19]–[21], and feature extraction [22], [23].

C. STYLE TRANSFER

Style transfer is to transform the image representation of an object into another image representation of the object. Real-time style transfer has attracted a lot of research due to its high speed and little requirement on datasets. AdaIN [2] was proposed to solve the dilemma of flexibility and speed in the transfer process. It is applied between the encoder and decoder of the generative network to match the feature space of the content image and style image, which can not only help transform any style in real time, but also ensure

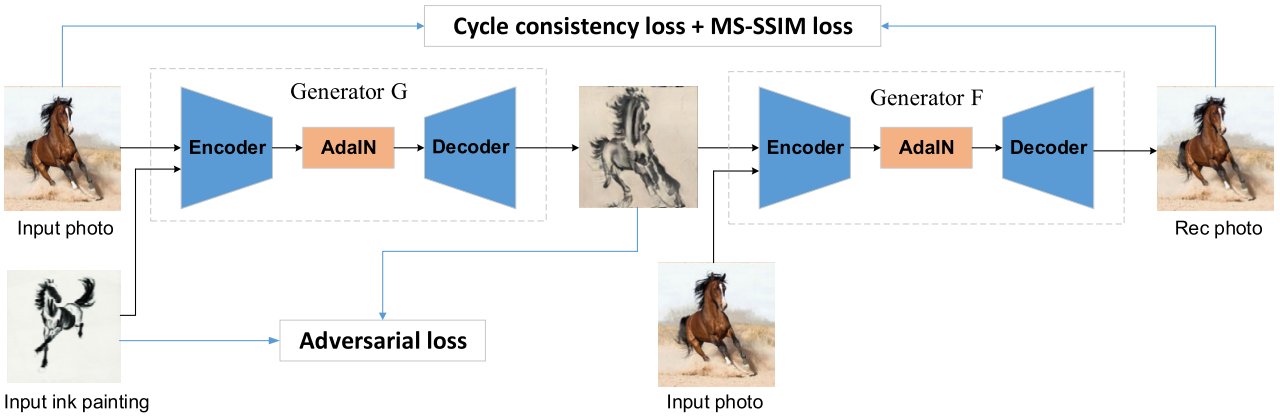

FIGURE 1. The pipeline of our proposed method, in which G and F are generators with the same structure, AdaIN is used between encoder and decoder.

the efficiency of calculation. The GAN-based style transfer is mainly reflected in the image translation between different style datasets, which requires a lot of training time, but is convenient and fast in the subsequent image transformation process. Among them, the supervised model Pix2Pix [24] and the unsupervised model CycleGAN [1] are the most classic. CycleGAN defines two generators and two discriminators, which is essentially a ring network composed of two mirror-symmetric GANs. The generators G and F are respectively responsible for the mapping of the image from the X domain to the Y domain and from the Y domain to the X domain, and the discriminators D_X and D_Y are respectively responsible for distinguishing the generated data and the real data of the X domain and the Y domain. In addition, cycle consistency loss is designed to replace the reconstruction loss. Recently, more and more attention models have been designed to realize image translation, such as Selection GAN [25], CSA [26], etc.

III. THE PROPOSED METHOD

Our method includes two generators and one discriminator. Generator G is used to convert images in domain X(real photos) to domain Y(ink paintings), and generator F is used to convert the generated domain Y images back to domain X. The discriminator is a 0-1 classifier, which is used to judge the generated picture and the real ink picture. The generated picture is represented by 0, and the real ink picture is represented by 1. We carry out back propagation by combining the adversarial loss, cycle consistency loss, identity loss and MS-SSIM loss to calculate the learning parameters.

A. FORWARD CYCLE FOR VOIDS AND STROKES

As we mentioned in the introduction, our method uses only a forward cycle to learn the content of the image. We know that CycleGAN can retain the outline of the image to a certain extent, because the introduction of the cycle consistency loss reduces the possibility of mapping paths from the source domain to the target domain. However, in the ink painting style transfer task, we found that the ink paintings generated using two cycles are not as good as using one cycle in strokes

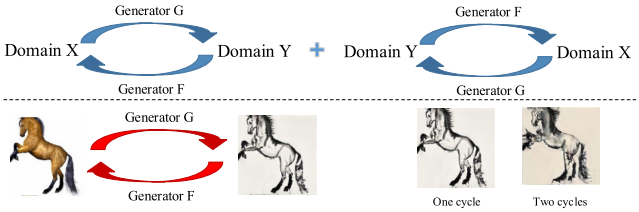

FIGURE 2. Diagram and results of one cycle and two cycles. top: schematic diagram of CycleGAN, bottom-left: schematic diagram of one cycle. bottom-right: the comparison results of one cycle and two cycles.

and voids. We believe that the strong constraint of the two cycle consistency losses enables the model to learn more complex and higher-level features, but also causes the loss of some semantic information that should be retained; and the model's excessive learning of different brightness information in each part of the image brings about the generation of unnatural voids (Void is an abstract feature that is difficult to define mathematically, here we perceive it with a subjective vision). The top of Fig. 2 shows the schematic diagram of CycleGAN's forward and backward cycles, in ink painting learning task, we only use the forward cycle consistency loss (Fig. 2 bottom-left).

B. LEARNING OF INK DIFFUSION

For further generating a more hierarchical image, we need to learn another typical feature of ink painting: ink diffusion, which refers to the edge diffusion effect formed by the movement of ink and water in the paper. To achieve this feature, we input both the content figure and the style figure into the generative network for encoding, and integrate them through AdaIN at the feature map level.

Specifically, AdaIN is similar to style transfer. It aligns the normalized channel-wise mean and variance of the content image to the styled image's, so that the generated picture has the same feature distribution as the ink painting. Different from other normalization methods, such as BN (Batch Normalization) and IN (Instance Normalization), AdaIN has no learnable affine parameters. According to the input style image, it adaptively generates affine parameters. If x and y

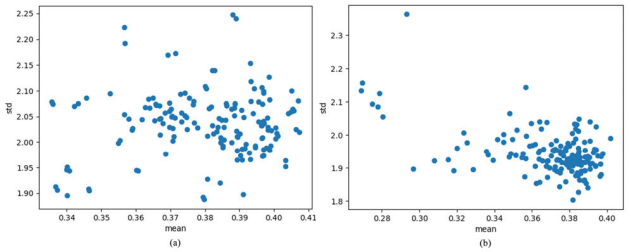

FIGURE 3. Data distribution of feature maps. (a) before the AdaIN module, (b) after the AdaIN module.

represents the feature map of content image and style image respectively, then the calculation of the AdaIN layer is as follows:

$$AdaIN(x, y) = \sigma(y)(\frac{x - \mu(x)}{\sigma(x)}) + \mu(y) \tag{1}$$

The feature map after AdaIN integration will be mapped to the image space through the decoding network. In addition, in order to maintain the data distribution after the calculation of AdaIN layer, we do not use any normalization layer in the decoder as mentioned in [2]. Fig. 3 shows the distribution of feature data before and after the AdaIN module. The horizontal axis represents the mean and the vertical axis represents the variance. We can see that the data points after AdaIN calculation is closer because of learning the similar ink diffusion effect.

C. LOSSES

1) ADVERSARIAL LOSS

Given a set of unpaired data from domain X and domain Y, by calculating the loss of generated image after domain $X \rightarrow Y$ mapping on the discriminator and the loss of real image in domain Y on the discriminator, the adversarial loss is expressed as follows:

$$\ell_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[\log D_Y(y)] + E_{x \sim p_{data}(x)}[\log(1 - D_Y(G(x)))] \tag{2}$$

where x represents the sample from the real photo dataset, y represents the sample from the real ink painting dataset, $G(x)$ represents the generated sample, and D_Y is a 0-1 classifier used to distinguish the generated image from the real ink image. G tries to minimize this loss function, while D_Y tries to maximize it.

2) CYCLE CONSISTENCY LOSS

The process of mapping sample x from domain X to domain Y through G and then back to domain X through F is called reconstruction. The reconstructed image and sample x should be highly similar, that is to say, $F(G(x)) \approx x$. Therefore, the cycle consistency loss is defined as follows:

$$\ell_{cyc}(G, F) = E_{x \sim p_{data}(x)}[||F(G(x)) - x||_1] \tag{3}$$

where $F(G(x))$ is the reconstructed image, we use L_1 norm to calculate the loss between the reconstructed image and the real image.

TABLE 1. Generator size of different methods.

Model	Params size of G (MB)	Layers of G	Estimated Total Size of G (MB)
DistanceGAN [28]	43.44	71	791.19
CycleGAN [1]	43.40	71	979.38
ChipGAN [7]	43.40	91	979.38
Ours	85.37	173	1744.06

3) MS-SSIM LOSS

Structural similarity index (SSIM) is an indicator that measures the similarity of two images. In practical applications, the Gaussian function is generally used to calculate the mean, variance, and covariance of the image, rather than traversing the pixels, in exchange for higher efficiency. Given images x, y, we first calculate the mean and variance of each image and the covariance of the two images:

$$\mu_x = \frac{1}{RC}\sum_{i=1}^{R}\sum_{j=1}^{C} x(i,j) \tag{4}$$

$$\mu_y = \frac{1}{RC}\sum_{i=1}^{R}\sum_{j=1}^{C} y(i,j) \tag{5}$$

$$\sigma_x^2 = \frac{1}{RC-1}\sum_{i=1}^{R}\sum_{j=1}^{C} (x(i,j) - \mu_x) \tag{6}$$

$$\sigma_y^2 = \frac{1}{RC-1}\sum_{i=1}^{R}\sum_{j=1}^{C} (y(i,j) - \mu_y) \tag{7}$$

$$\sigma_{xy} = \frac{1}{RC-1}\sum_{i=1}^{R}\sum_{j=1}^{C} (x(i,j) - \mu_x)(y(i,j) - \mu_y) \tag{8}$$

then the luminance, contrast and structure comparison measures are given as follows:

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{9}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{10}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{11}$$

finally the value of SSIM is calculated by:

$$SSIM(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma} \tag{12}$$

when $C_3 = C_2/2, \alpha = \beta = \gamma = 1$, the calculation of SSIM can be simplified as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{13}$$

MS-SSIM is obtained by calculating the value of SSIM on multiple scales:

$$SSIM(x, y) = [l_M(x, y)]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j} \tag{14}$$

where $M = 1$ represents the original image size, $M = 2$ represents 1/2 of the original image size, and so on., α, β, γ adopt the given default value.

Through the above formulas, we get the MS-SSIM loss between the image of domain X and its reconstructed image:

$$\ell_{MS-SSIM}(G, F) = 1 - SSIM(x, F(G(x))) \tag{15}$$

4) IDENTITY LOSS

In order to ensure that generator G is mapping to the Y domain, the image of Y domain input to G should still generate the image of Y domain. We calculate the loss between the input y and output y':

$$\ell_{identity}(G) = E_{y\sim p_{data}(y)}[||G(y) - y||_1] \tag{16}$$

D. OBJECTIVE FUNCTION

To sum up, our total loss function is defined as follows:

$$\ell(G, F, D_Y) = \ell_{GAN}(G, D_Y, X, Y) + \lambda\ell_{cyc}(G, F) + \beta\ell_{MS-SSIM}(G, F) + \ell_{identity}(G) \tag{17}$$

where the parameters λ and β are used to control the linear combination of these losses. Our goal is to optimize a minimax function:

$$G^*, F^* = \arg\min_{F,G}\max_{D_Y} \ell(G, F, D_Y) \tag{18}$$

IV. EXPERIMENT

In this section, we will prove the effectiveness of our proposed method through several evaluation indicators and comparative experiments. We select the following existing methods for comparison: Neural Style Transfer [27], DistanceGAN [28], CycleGAN [1], ChipGAN [7]. Table 1 shows the size of the generator for different GAN-based models. Our experiments were executed under win10 system, using an Intel(R) Xeon(R) Silver 4110 CPU at 2.10 GHz with 16GB memory and a 11GB NVIDIA GeForce RTX 2080 Ti GPU.

A. DATASETS AND TRAINING DETAILS

1) DATASET

We use the ChipPhi dataset collected in [7]. There are two kinds of ink datasets, one is horse and the other is landscape painting. We conducted experiments on the horse dataset. Domain X contains 1478 training images with different resolutions and 160 test images of $256 * 256$ size. Domain Y contains 822 training images with different resolutions and 90 test images of $256 * 256$ size.

2) TRAINING DETAILS

In our experiments, we initialize the weights of the convolutional layer to a normal distribution with a mean of 0 and a standard deviation of 0.02. We set the batchSize to 1 and the number of epochs to 200. The learning rate is set to 0.0002 in the first 100 epochs, and the next 100 epochs is linearly decayed to 0. Adam optimization algorithm with betas =(0.5, 0.999) is used for generator G, F and discriminator D_Y. In the linear combination of total loss function, the coefficients of identity loss and cycle consistency loss are set to 5.0 and 10.0 respectively, while the coefficients of other loss terms are default to 1.0. In addition, the scale in MS-SSIM loss is set to 5.

B. EVALUATION METRICS

Evaluation indicators are introduced in this section. We selected four evaluation indicators, which can be divided into two categories: FID and Kernel MMD for evaluating GAN networks, PSNR and SSIM for evaluating the quality of generated images.

1) FID

The FID score [29] measures the distance between the real picture and the generated picture at the feature level. By using InceptionV3 to generate $N * 2048$ vectors for the N pictures of the real dataset, the mean value μ_x is obtained, and then InceptionV3 is also used to generate the $M * 2048$ vectors for the generated M pictures, to obtain the mean value μ_y. The FID can be calculated as follows:

$$FID = ||\mu_x - \mu_y||^2 + T_r(\Sigma_x + \Sigma_y - 2(\Sigma_x\Sigma_y)^{1/2}) \tag{19}$$

where Σ_x and Σ_y represent the covariance of the real dataset and the generated dataset respectively, and T_r represents the trace of the matrix.

2) KERNEL MMD

Kernel MMD [30] measures the difference of data distribution between the real dataset and the generated dataset, which can evaluate the quality of the generated image to a certain extent, with low computational cost.

3) PSNR AND SSIM

PSNR and SSIM [31] are widely used image evaluation indicators, which are based on the error between corresponding pixels. Since the visual characteristics of human eyes are not taken into account, there are often inconsistencies between the evaluation results and human subjective feelings. However, due to the universality of its use, we still choose it as a reference. SSIM measures image similarity from brightness, contrast and structure.

C. EXPERIMENT RESULTS

1) COMPARISON WITH EXISTING METHODS

a: QUALITATIVE COMPARISON

Fig. 4 shows the comparison results of our method and the existing methods. It can be seen that our method generates images with higher quality.

For example, in the first row of Fig. 4, our method generates more natural voids effect on the horse's neck and hind legs than ChipGAN, and more realistic stroke information on the horse's hind legs than CycleGAN, while the results generated by DistanceGAN and Neural style transfer are not very good. For the sample in the second row, our method retains

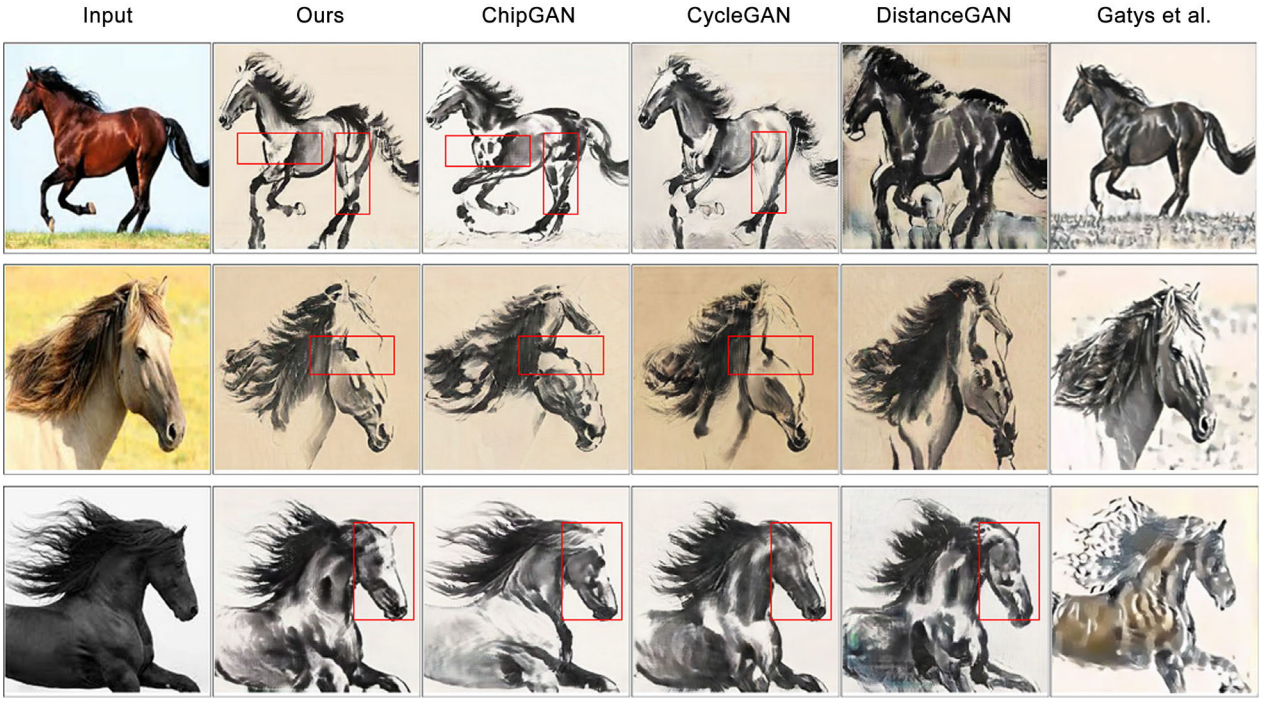

FIGURE 4. Comparison with existing methods on real horse->ink painting. In each case: input image (column 1), result of ours (column 2), result of ChipGAN [7] (column 3), result of CycleGAN [1] (column 4), result of DistanceGAN [28] (column 5), result of Neural Style Transfer [27] (column 6).

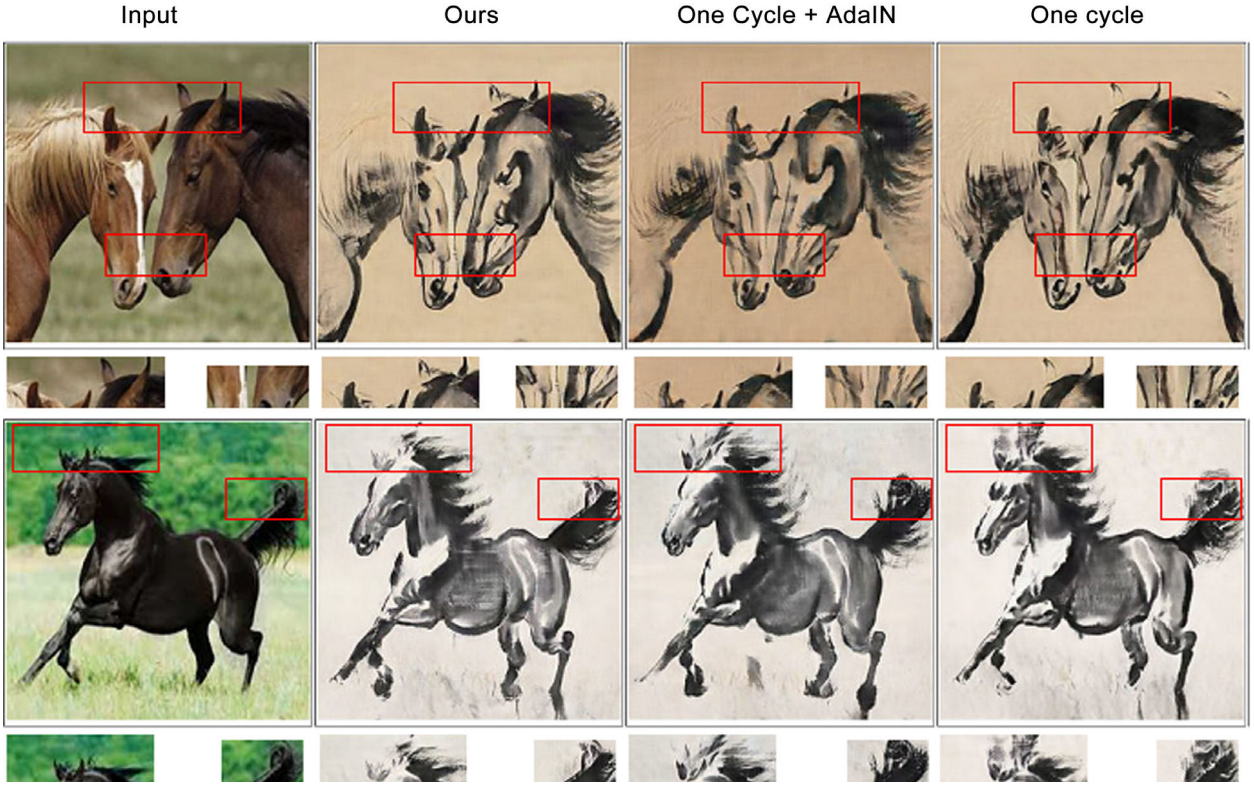

FIGURE 5. The image quality with different variables for real horse->ink horse task. From left to right: Input, Our proposed method (One cycle + AdaIN + MS-SSIM loss), One cycle + AdaIN, One cycle.

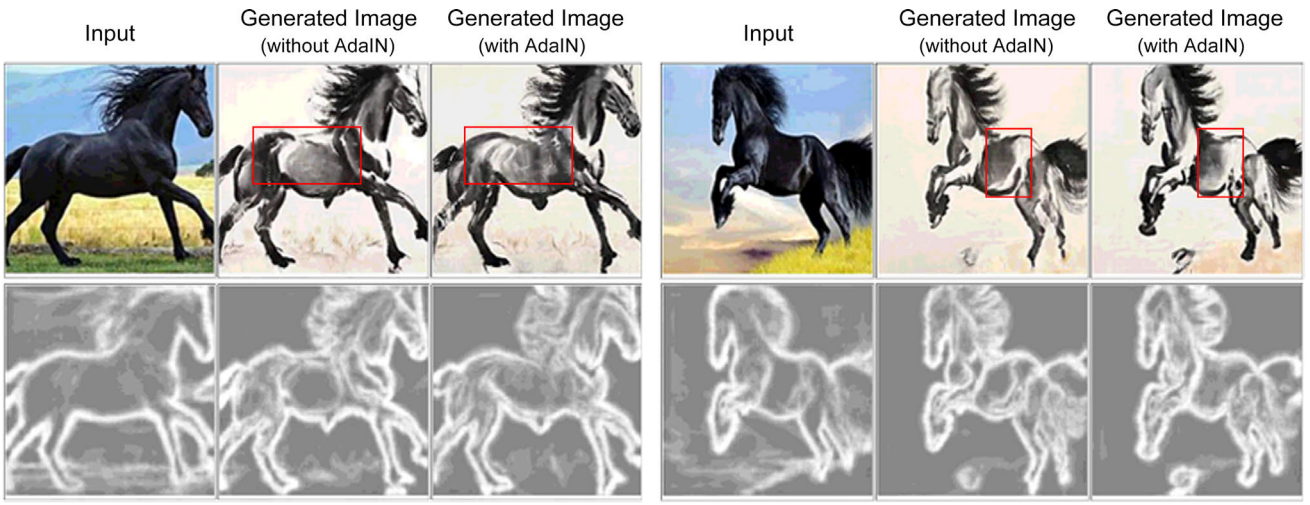

FIGURE 6. The effect of AdaIN module for real horse->ink horse task. In each case: The first row shows the original image (left), generated image without AdaIN (middle), generated image with AdaIN (right), and the second row shows the corresponding edge images.

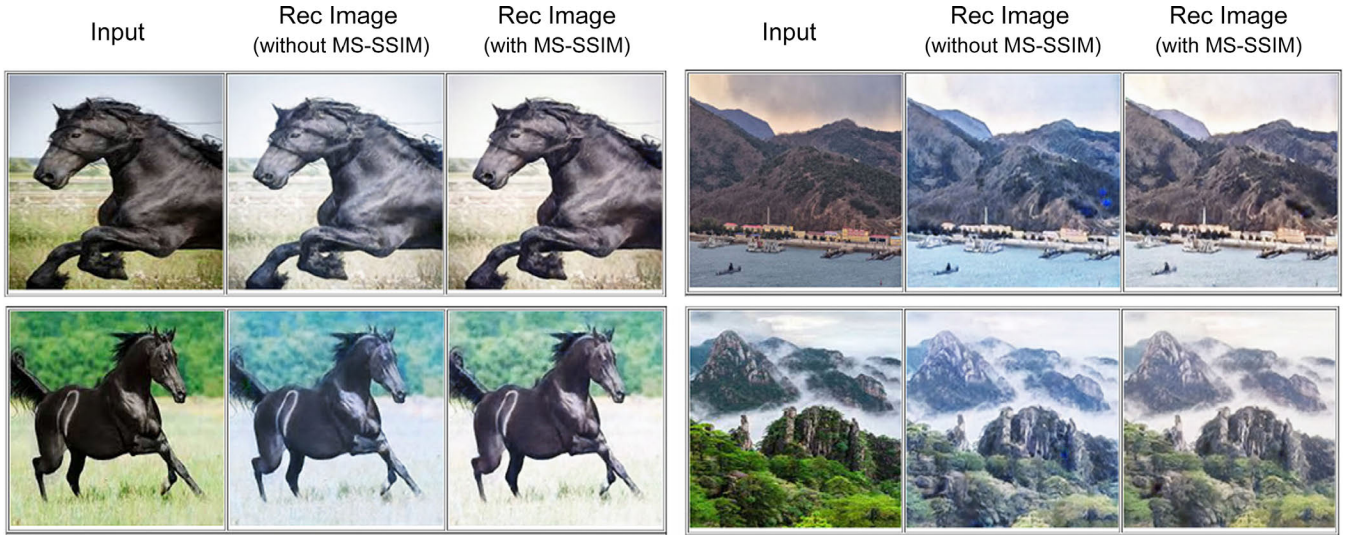

FIGURE 7. The improvement of reconstructed image quality made by MS-SSIM loss. In each case: Input (left), reconstructed image without MS-SSIM loss (middle), reconstructed image with MS-SSIM loss (right).

more complete details on the face and ears of the horse than ChipGAN and CycleGAN, and the generation of DistanceGAN in the mouth has collapsed, the Neural Style Transfer method only learned the color of ink painting. For sample 3, our method produces more realistic visual effects on the face and ears. In summary, our method produces more satisfactory results in most cases. ChipGAN is slightly inferior to the voids effect and the constraints of some details. CycleGAN produces unsatisfactory results in the retention of image content. DistanceGAN is prone to generate poor quality data, and the Neural Style Transfer basically only learns the color information.

b: QUANTITATIVE COMPARISON

We evaluate our model on five indicators: FID (smaller is better), Kernel MMD (smaller is better), PSNR (larger is better), SSIM (larger is better), and training time (smaller is better). Table 2 shows the comparison results of our experiments. It can be seen that our method shows better results on most indicators and requires less training time. It shows that our method can not only retain the semantic information of the image, but also learn the style features of ink painting well.

2) ABLATION STUDY

In Fig. 5, we analyze our model through different variables. It can be seen that in the structure using only one cycle (the last column in Fig. 5), although more content is reserved than the baseline model, in sample 1 (first row), the horse's ears still lack some details, and the faces of the two horses are fused together, the ear of the horse in sample 2 (second row) is poorly generated, and the tail and background are mixed together resulting in a redundant part. The structure

TABLE 2. Comparison with existing methods on the five indicators of FID, Kernel MMD, PSNR, SSIM, and Training time.

Model	FID	Kernel MMD	PSNR	SSIM	Training time
DistanceGAN [28]	239.567	1.0375	7.2046	0.8482	55h
CycleGAN [1]	219.0869	0.9850	**8.7929**	0.9101	54.4h
ChipGAN [7]	242.1771	1.1263	8.4180	0.9021	51.5h
Ours	**209.8527**	**0.9581**	8.4756	**0.9173**	**40h**

TABLE 3. Performance on PSNR and SSIM before and after using AdaIN.

Evaluation index	Before using AdaIN	After using AdaIN	The enhancement of generated image quality.
PSNR	8.7948	8.9134	11.86%
SSIM	0.9103	0.9190	0.87%

of one cycle + AdaIN (the third column of Fig. 5) has been improved in detail. In the middle of the horse's face in sample 1 (first row), due to the filtering out of some background information, the faces of the two horses can be separated, but some information at the ears is still lost. In sample 2 (second row), the ears of the horse are improved and the lines of the tail are clearer. Our method (one cycle + AdaIN + MS-SSIM loss, the second column of Fig. 5) further improves the image generation quality. The ears in sample 1 (first row) are preserved well. As the stroke information of the face is more detailed, the two horses are completely separated, the generated image looks more spatial. The ear and tail parts of sample 2 (second row) are also closer to the input image.

By comparing generated images and extracting edge information, we analyze the effect of the AdaIN module. The first row of Fig. 6 shows the original images and generated ink images before and after using AdaIN of the two samples, and the second row shows the edge images of corresponding images in the first row. We observe that AdaIN's learning of ink diffusion effect mainly focuses on the ink color transition of horse's body. The uniform and reasonable transition can improve the hard color block phenomenon, thus adding more texture details to the generated ink image. It can also be clearly observed from the comparison of the edge images that AdaIN removes the extra strokes and replaced them with ink transitions on the corresponding horse' body. Table 3 shows the qualitative comparison results on PSNR and SSIM, which are increased by 11.86% and 0.87% respectively.

Fig. 7 clearly shows the improvement in visual quality of the reconstructed image after using MS-SSIM loss through comparative experiments. We select two types of test data for horses and landscapes, and find that the generated reconstructed images tend to be green without using MS-SSIM loss. We believe that one possible reason is the influence of background color. For example, the background of most training images in the horse dataset is grass, and the main

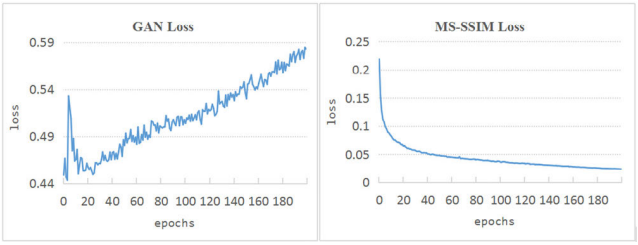

FIGURE 8. The curve of the GAN loss and MS-SSIM loss as epoch increases.

body of the landscape dataset is basically trees. However, MS-SSIM loss restricts the reconstructed image in terms of luminance, contrast, etc., thereby improving this problem and maintaining the consistency of the reconstructed image and the original image.

3) MODEL ANALYSIS

In this section, we analyze the proposed method and evaluate its generalization ability on different datasets.

Fig. 8 shows the curves of the adversarial loss and MS-SSIM loss of the training process. It can be seen that GAN loss is constantly fluctuating, indicating that our network is well trained, and MS-SSIM loss continues to decline until it stabilizes, indicating that our reconstructed image continues to be close to the real image as the number of training epoches increases.

Fig. 9 shows the generalization results of our model on different datasets. In order to ensure the diversity of the generalized data, we selected a variety of generalized datasets, including landscape and flower datasets which involve more brush strokes and texture details, apple datasets which represent simple geometric shapes, Western paintings such as Van Gogh's Paintings, and animal datasets which contain complex shapes such as birds, cats, and dogs. Results show that our method has good generalization ability on different datasets.

FIGURE 9. Generalization experiments of our proposed method on different datasets.

V. CONCLUSION

In this paper, we proposed an image translation framework for a real photos to ink paintings task. To resolve the loss of image content information in the baseline model CycleGAN, we used only forward cycle consistency loss to replace the forward and backward combination method, which effectively retains the semantic information and is capable of generating natural voids. In order to solve the problem that the generated images are not stereoscopic enough, we used the AdaIN module to learn the ink diffusion effect of real ink paintings, so that the generated images look more realistic and natural. For the correction of image details, we added the MS-SSIM loss to the reconstruction loss, thereby effectively improves the quality of the generated images. We conducted comparative experiments with existing methods on multiple evaluation indicators, and analyzed the generalization ability, proved the effectiveness of our method. In the future, we will study the real-time style transfer of ink paintings, hoping to learn the typical characteristics of ink paintings under real-time conditions.

REFERENCES

[1] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," 2017, *arXiv:1703.10593*. [Online]. Available: http://arxiv.org/abs/1703.10593

[2] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proc. ICCV*, Oct. 2017, pp. 1501–1510.

[3] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Nov. 2003, vol. 2, no. 1, pp. 1398–1402.

[4] Y. Wang, W. Li, and Q. Zhu, "Ink wash painting style rendering with physically-based ink dispersion model," *J. Phys., Conf. Ser.*, vol. 1004, Apr. 2018, Art. no. 012026.

[5] Y. Gua, N. Ono, and K. Urahama, "Real-time rendering of 3D ink-wash painting based on geometry buffers," *J. Inst. Ind. Appl. Eng.*, vol. 5, no. 2, pp. 65–70, Apr. 2017, doi: 10.12792/jiiae.5.65.

[6] S. Luo, S. Liu, J. Han, and T. Guo, "Multimodal fusion for traditional Chinese painting generation," in *Advances in Multimedia Information Processing—PCM* (Lecture Notes in Computer Science), vol. 11166, R. Hong, W. H. Cheng, T. Yamasaki, M. Wang, and C. W. Ngo, Eds. Cham, Switzerland: Springer, Cham, 2018.
[7] B. He, F. Gao, D. Ma, B. Shi, and L.-Y. Duan, "ChipGAN: A generative adversarial network for Chinese ink wash painting style transfer," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*. New York, NY, USA: ACM, 2018, pp. 1172–1180.
[8] R. Zhou, J. H. Han, H. S. Yang, W. Jeong, and Y. S. Moon, "Fast style transfer for chinese tranditional ink painting," in *Proc. IEEE 9th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Beijing, China, Jul. 2019, pp. 586–588, doi: 10.1109/ICEIEC.2019.8784632.
[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[10] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," 2016, *arXiv:1611.04076*. [Online]. Available: https://arxiv.org/abs/1611.04076
[11] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017, *arXiv:1701.04862*. [Online]. Available: https://arxiv.org/abs/1701.04862
[12] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, NSW, Australia, Aug. 2017, pp. 214–223, 2017.
[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of Wasserstein GANs," 2017, *arXiv:1704.00028*. [Online]. Available: https://arxiv.org/abs/1704.00028
[14] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: https://arxiv.org/abs/1511.06434
[15] M. Mirza and S. Osindero, "Conditional generative adversarial nets," Nov. 2014, *arXiv:1411.1784*. [Online]. Available: https://arxiv.org/abs/1411.1784
[16] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a Laplacian pyramid of adversarial networks," 2015, *arXiv:1506.05751*. [Online]. Available: https://arxiv.org/abs/1506.05751
[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," 2017, *arXiv:1710.10196*. [Online]. Available: https://arxiv.org/abs/1710.10196
[18] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4401–4410.
[19] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8789–8797.
[20] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, "Diverse image-to-image translation via disentangled representations," in *Proc. ECCV*, 2018, pp. 35–41.
[21] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. ECCV*, 2018, pp. 172–189.
[22] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.
[23] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," 2016, *arXiv:1605.09782*. [Online]. Available: https://arxiv.org/abs/1605.09782
[24] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.
[25] H. Tang, D. Xu, N. Sebe, Y. Wang, J. J. Corso, and Y. Yan, "Multi-channel attention selection GAN with cascaded semantic guidance for cross-view image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2417–2426.
[26] H. Liu, B. Jiang, Y. Xiao, and C. Yang, "Coherent semantic attention for image inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4170–4179.
[27] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015, *arXiv:1508.06576*. [Online]. Available: https://arxiv.org/abs/1508.06576
[28] S. Benaim and L. Wolf, "One-sided unsupervised domain mapping," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 752–762.
[29] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "GANs trained by a two time-scale update rule converge to a local Nash equilibrium," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6626–6637.
[30] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, Mar. 2012.
[31] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

FENGQUAN ZHANG received the Ph.D. degree in computer science from the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing, China, in 2013. He is currently an Associate Professor with the School of Information Science and Technology, North China University of Technology, Beijing, China. His research interests concern on artificial intelligence, computer vision, computer graphic, computer animation, parallel computing, and mobile computing.

HUAMING GAO received the bachelor's degree in digital media technology from the North China University of Technology, in 2018. She is currently pursuing the master's degree in computer science. Her researches focus on computer vision and deep learning.

YUPING LAI (Member, IEEE) received the Ph.D. degree in information security from the Beijing University of Posts and Telecommunications, Beijing, China, in 2014. He has been an Associate Professor with the North China University of Technology, China, since 2018. His research interests include information security, computer vision, pattern recognition, machine learning, and data mining.