*ICIC Express Letters*
*Part B: Applications*
Volume **10**, Number **11**, November **2019**
 ISSN 2185-2766
pp. 995–1002

# RESEARCH AND IMPLEMENTATION OF IMAGE GENERATION TECHNOLOGY BASED ON GAN

Xizi Wang and Yuping Zhang

School of Automation Engineering
University of Electronic Science and Technology of China
No. 2006, Xiyuan Ave., West Hi-Tech Zone, Chengdu 611731, P. R. China
{ 292476024; 147552408 }@qq.com

Received May 2019; accepted August 2019

Abstract. *This paper introduces some practical applications and research significance of GAN. After analyzing the advantages and disadvantages of GAN, use the improved CycleGAN to carry out the style migration experiment of the picture. Use the objective functions of GAN and WGAN to observe the effects of different objective functions through the generated pictures. Carry out the face generation experiment based on TL-GAN. The dataset used in this article is from Stanford. It contains 6288 landscape images and 1073 Monet's paintings, all of which are 256 × 256 pixel RGB images. We compare and analyze the experimental results, summarize their advantages, limitations and improvements. Then we research and implement image generation technology based on different GAN. Finally summarize different GANs' innovation point and put forward some improvement spaces of GAN in this paper.*
**Keywords:** Deep learning, GAN, Image generation, Style migration

1. **Introduction.** In recent years, with the Internet's data burst and the GPU's computing power increasing, deep learning has been greatly developed. The emergence of the Generative Adversarial Networks (GAN) has pushed deep learning to a new era [1]. GAN provides new techniques and means for the development of computer vision. It generates high-quality samples with unique zero-sum game and confrontation training [2]. And it has more powerful feature learning and feature expression capabilities than traditional machine learning algorithms. At present, it has achieved remarkable success in the field of machine vision, especially in the field of image generation. So it is the hotspot of current research.

GAN's main inspiration comes from the idea of zero-sum game. Applied to deep learning neural network, GAN is through the continuous gaming between network $G$ (Generator) and network $D$ (Discriminator) [3] to let $G$ learn the distribution of the data. When applied to image generation, $G$ can generate a realistic image via a section of random number after the training is completed [4]. The main function of $G$, $D$ is: $G$ is a generative network that receives a random noise $z$ (random number), and generates an image through this noise; $D$ is a discriminant network that discriminates whether the picture is 'real' [5]. During the training process, the goal of $G$ is to generate a real picture as much as possible to deceive the discriminant network $D$. The goal of $D$ is to try to identify the images generated by $G$ is real or fake. Thus, $G$ and $D$ constitute a dynamic "gaming process", and the final equilibrium point is the Nash equilibrium point [6].

This paper mainly makes a simple combining and comparison of the current popular GAN model, and tests the performance of various GAN in the field of face generation and style migration. At the same time, this paper analyzes the results and proposes improved methods. The style migration based on CycleGAN breaks the requirements

DOI: 10.24507/icicelb.10.11.995

for the emergence of training data integration, making the application of style migration more universal. The face generation based on TL-GAN uses the hidden features to customize the face generation, which makes the application of the technology more flexible. Therefore, we choose these two technologies for experimental applications.

In Section 2, GAN, WGAN, CycleGAN and TL-GAN are introduced. GAN model implementation, training and results are stated in Section 3, where the experimental effect is displayed in Section 4. Finally, a conclusion in Section 5 closes the work.

2. **GAN.** GAN is a generation model that can be used to generate images, audio, etc. And the quality of generation increases year by year, as shown in Figure 1. Image generation can also be used to generate a high-definition beautiful character as the protagonist of the poster, save a lot of manpower and material time, and advertising fees for hiring celebrities, etc.


FIGURE 1. GAN generation quality development trend

The quality of the generated images is getting higher and higher, which means that you can use them to repair some of the faces of ancient photos, for example, use old photos of unclear celebrities to recreate their appearance, as shown in Figure 2(a); or to transform the quality of the cartoons that were watched in your childhood from BD to HD, which is also very meaningful.

(a) From BD to HD (b) CycleGAN plus a smile (c) CycleGAN generates anime portrait

FIGURE 2. GAN's application

We can use CycleGAN to carry out the style migration of pictures, which can transform landscape painting to oil painting, convert horse to zebra, etc. The main contribution of CycleGAN is to provide an unsupervised image translation method. CycleGAN can add a smile to the cool big brother, completely different between the original picture style, as shown in Figure 2(b); convert the face image into a cartoon image, as shown in Figure 2(c); generate the elder faces of one person according to his or her current photo, etc.

Based on these applications of GAN, we implemented two of the most representative applications: style migration and face generation. We use the improved CycleGAN to carry out the image style migration experiment and the objective functions of GAN and WGAN to observe the effects of different objective functions through the generated images to perform face generation experiments based on TL-GAN.

2.1. **Wasserstein GAN.** WGAN introduces the Wasserstein distance instead of JS divergence and KL divergence as an optimization goal [7]. As Wasserstein distance has the superior smoothing characteristics compared with KL divergence and JS divergence, it solves the gradient disappearance problem of the original GAN fundamentally [8]. What is more, the original GAN is difficult to train, the training process is usually heuristic, it requires a well-designed network architecture, which is not universal. The loss of the generator and discriminator cannot indicate the training process and generated samples lack diversity, too.

2.2. **CycleGAN.** The traditional GAN's $G$ is to convert random noise into pictures, but in style migration we need to convert the picture into a picture, so this time we have to use the picture as $G$ input, then $G$ is going to learn a mapping. However, training with a single GAN is unstable and may cause all photos to be mapped to the same image's mode collapse. At this time CycleGAN was proposed to solve this problem. CycleGAN [9] is a combination of two GANs, and the purpose is to achieve the conversion of unpaired images, especially for image style migration. CycleGAN is used to improve stability, and its innovation: it can migrate image content from the source domain to the target domain without paired training data.

2.3. **TL-GAN.** Transparent Latent-space GAN is a new and efficient method for controllable synthesis and editing. It allows users to gradually adjust single or multiple features using a single network. In addition, adding new adjustable features can be done very efficiently in less than an hour. Therefore, we performed a face generation experiment based on TL-GAN to fine tune one or more features of the face.

3. **GAN Model Implementation, Training and Results.** This section mainly introduces the implementation based on CycleGAN and TL-GAN, and elaborates the concrete process of model construction and training.

3.1. **The difficulties of achieving GAN model.**

1) The model is difficult to converge.

(a) Reason: The design of the GAN model is based on the idea of a two-person zero-sum game, which is essentially a maximin game. In game theory, when the discriminator and generator reach the Nash equilibrium, the GAN model reaches convergence, and the following Formula (1) is the state in which the GAN model reaches the optimal moment. Among them, $V(D,G)$ is the objective function of GAN. $E$ represents the mathematical expectation of the real data $x$ and the noise data $z$. $P(z)$ is the noise distribution, generally a Gaussian distribution to obtain a generated data distribution $Pg(x)$. We want $Pg(x)$ to be very close to $Pr(x)$ to fit the approximation of the true distribution.

$$\min_G \max_D V(D,G) = E_{x\sim P_r(x)}[\log D(x)] + E_{z\sim P_z(z)}[\log(1-D(G(z)))] \tag{1}$$

For tasks with non-convex functions, some cost functions do not converge with gradients, resulting in non-convergence of the GAN model. In addition, the GAN model uses the Jensen-Shannon divergence. Due to the asymmetry of the Jensen-Shannon divergence, it causes the difficulty of convergence and instability of the GAN model, as shown in Formula (2). $P(x)$ is the true probability distribution and $Q(x)$ is the fitted probability distribution.

$$D_{KL}(P||Q) = -\sum_{x\in X} P(x)\log\frac{1}{P(x)} + \sum_{x\in X} P(x)\log\frac{1}{Q(x)} = \sum_{x\in X} P(x)\log\frac{P(x)}{Q(x)} \tag{2}$$

(b) Solution: Replacing the Jensen-Shannon divergence with Wasserstein distance can effectively overcome the difficulty of convergence and instability of the GAN model.

$$W(P_r,P_g) = \inf_{\gamma\sim\prod(P_r,P_g)} E(x,y)\sim\gamma[||x-y||] \tag{3}$$

The superiority of the Wasserstein distance compared to the $KL$ divergence and $JS$ divergence is that even if the two distributions do not overlap, the Wasserstein distance can still reflect their distance. WGAN's book shows it through a simple example [10,11].

2) The gradient disappears.

(a) Reason: When the discriminator $D$ is trained to be optimal, the generator $C(G)$ is:

$$C(G) = 2JSD\left(P_{data}\left|P_g\right.\right) - 2\log 2 \tag{4}$$

Among them,

$$JS\left(P_1\left\|P_2\right.\right) = \frac{1}{2}KL\left(P_1\left\|\frac{P_1+P_2}{2}\right.\right) + \frac{1}{2}KL\left(P_2\left\|\frac{P_1+P_2}{2}\right.\right) \tag{5}$$

When the two distributions are different, as

$$\begin{array}{c} P_1(x) = 0 \text{ and } P_2(x) = 0 \\ P_1(x) \neq 0 \text{ and } P_2(x) \neq 0 \\ P_1(x) = 0 \text{ and } P_2(x) \neq 0 \\ P_1(x) \neq 0 \text{ and } P_2(x) = 0 \end{array} \tag{6}$$

the distribution of the first item and the second item has no reference value. By analyzing the third and the fourth item, the following Formula (7) can be obtained by substituting the third item into the calculation:

$$\log \frac{P_2}{\frac{1}{2}\left(P_2+0\right)} = \log 2 \tag{7}$$

At this time, the $JS$ divergence is $\log 2$, that is, $C(G) = 0$, which leads to the better the GAN is trained in the discriminator, the more likely the generator is to disappear. At this time the condition is that the generating distribution and the real distribution are not coincident, as shown in Figure 3.

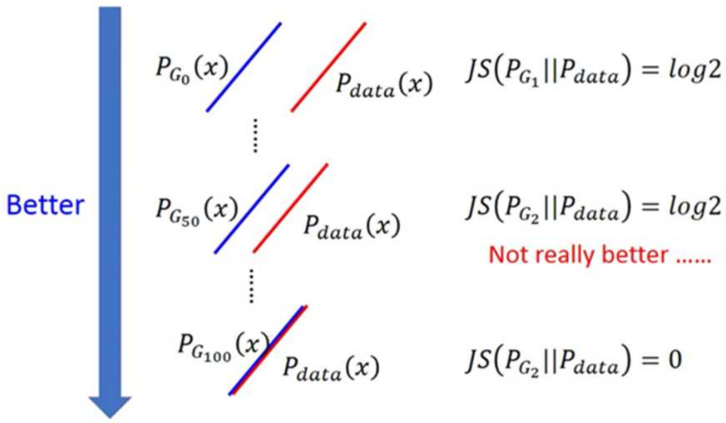



FIGURE 3. $JS$ divergence chart

So the original generator's loss function has no meaning. It is to let the generator minimize the probability discriminator identifies a fake sample it generates, but in fact it will cause the gradient to disappear, as the generator generates pictures badly when starting training at first. The discriminator can identify it easily, so that the discriminator training has no loss. So there is no effective gradient information back to the generator to optimize itself, which causes the gradient will disappear.

(b) Solution: The discriminator wants to widen the score gap of the true and false samples as much as possible. So the larger the gradient, the better the change. So the discriminator adds gradient penalty based on the Wasserstein distance after sufficient training. The improved optimization goals are as follows.

$$E_{x\sim P_{data}}[D(x)] - E_{x\sim P_G}[D(x)] - \lambda E_{\hat{x}\sim P_{\hat{x}}}p\left[\left(\left\|\nabla_{\hat{x}}D\left(\hat{x}\right)\right\|_2 - 1\right)^2\right] \tag{8}$$

At the same time, adding Gaussian white noise during the training process can effectively reduce the gradient disappearance problem.

3.2. **Experimental parameter setting and training process.**

1) CycleGAN style migration experiment: The training parameters are as follows: epoch = 120, batchsize = 1, lr = 0.0002. The size of the input image is $(256, 256, 3)$.

Training process: The training process is mainly reflected by the loss function.

(a) Generator loss function:

$$L\left(G_{AB}, G_{BA}, A, B\right) = E_{a\sim A}\left[\left\|G_{BA}\left(G_{AB}(a)\right) - a\right\|_1\right] \tag{9}$$

(b) Discriminator loss function:

$$L\left(G_{AB}, D_B, A, B\right) = E_{b\sim B}\left[\log D_B(b)\right] + E_{a\sim A}\left[\log\left(1 - D_B\left(G_{AB}(a)\right)\right)\right] \tag{10}$$

(c) Cycle loss function:

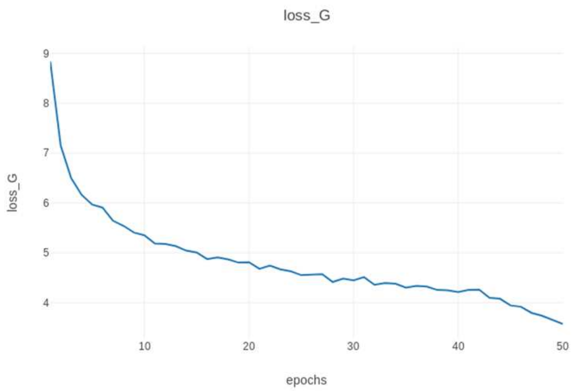
$$L_{cyc}(G, F) = E_{x\sim P_{data}(x)}\left[\,\|F(G(x)) - x\|_1\,\right] + E_{y\sim P_{data}(y)}\left[\,\|G(F(y)) - y\|_1\,\right] \tag{11}$$

(d) Approximate identity mapping loss function:

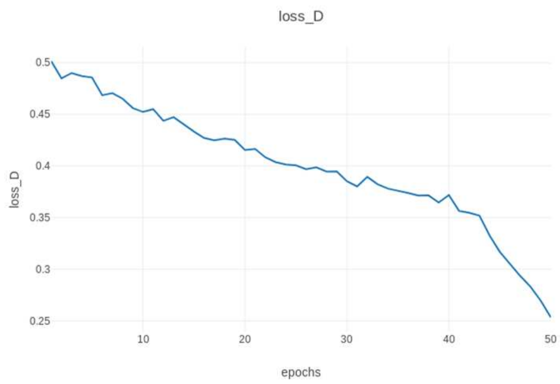
$$L_{cyc}(G, F) = E_{y\sim P_{data}}\left[\,\|G(y) - y\|_1\,\right] + E_{x\sim P_{data}}\left[\,\|F(x) - x\|_1\,\right] \tag{12}$$

In Formulas (9) and (10), $a$ and $b$ is an image obeying $A$ and $B$ distribution. $G_{AB}()$ is a generator network $B$, $G_{BA}()$ is a generator network $A$, $D_B$ is a discriminator network $A$. In Formulas (11) and (12), $x$ is a class $A$ image, $y$ is a class $B$ image, $G$ is a generator network $B$, $F$ is a generator network $A$. $E$ is the expectation of distribution.

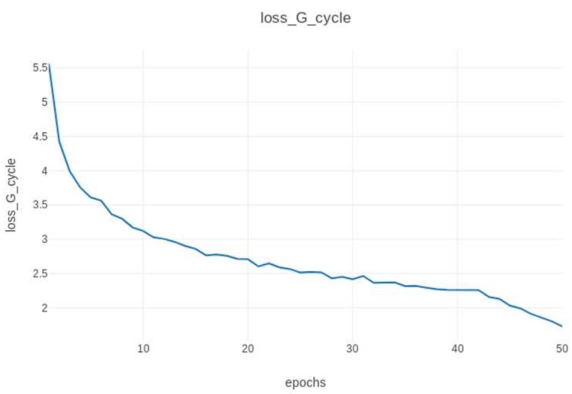
The loss function curve is shown in Figure 4: the ordinate is the value of the generator loss function, and the abscissa is the times of training iterations.



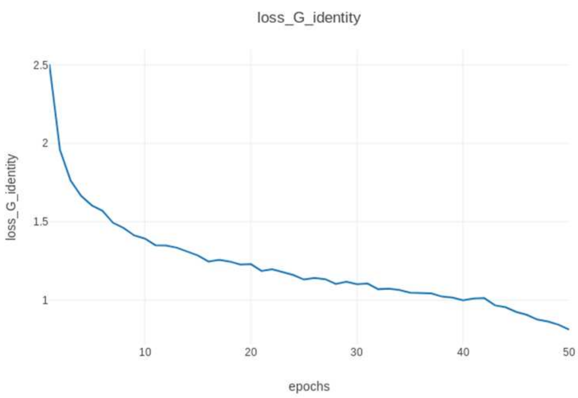

(a) Generator loss




(b) Discriminator loss




(c) Cycle loss




(d) Approximate identity mapping loss

Figure 4. Loss function curve

2) TL-GAN face generation experiment: The training parameters of the TL-GAN model are as follows: epoch = 1000, batchsize = 16, lr = 0.001.

Training process: The loss function used in the experiment is the Wasserstein distance.

(a) Generator loss function:

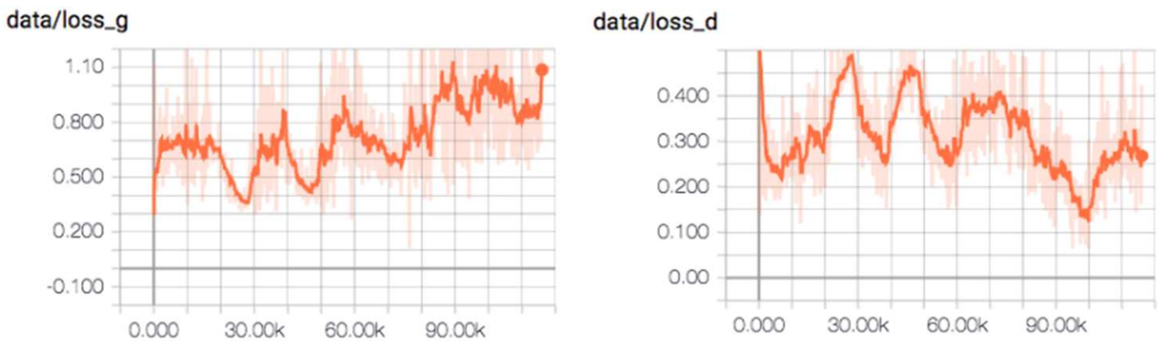
$$L_G^{WGAN} = E[D(G(z))] \tag{13}$$

(b) Discriminator loss function:

$$L_D^{WGAN\_GP} = L_D^{WGAN} + \lambda E\left[(|\nabla D(\alpha x - (1 - \alpha G(z)))| - 1)^2\right] \tag{14}$$

In Formulas (13) and (14), $z$ is the $n$-dimensional noise, $G()$ is the generator network, $D()$ is the discriminator network, $E$ is the distribution expectation, $\lambda$ is the weighting factor, and $\alpha$ is the generator coefficient.

The loss function curve simulation is shown in Figure 5: the ordinate is the value of the discriminator loss function, and the abscissa is the number of training iterations.




(a) Generator loss function (b) Discriminator loss function

FIGURE 5. Loss function curve simulation

## 4. Experimental Effect Display.

### 4.1. CycleGAN experimental effect display.

1) Migrate Apple's style to Orange's style, as shown in Figure 6(a).
2) Migrate Orange's style to Apple's style, as shown in Figure 6(b).
3) Migrate Summer's style to Winter's style, as shown in Figure 6(c).
4) Migrate Winter's style to Summer's style, as shown in Figure 6(d).

### 4.2. TL-GAN experimental effect display.

1) Randomly generate a 40-dimensional Gaussian white noise, a network generated face image, which is shown in Figure 7.

2) Modify the Young property to increase the age. The woman had a lot of wrinkles on her face, which is shown in Figure 8(a).

3) Modify the Male property to make the resulting image more masculine, which is shown in Figure 8(b).

4) Modify the Sideburns property to make the temples more prominent. The color of the man's temples has deepened and become more apparent, which is shown in Figure 8(c).

5) Modify the big nose property and the nose becomes bigger, as shown in Figure 8(d).

**4.3. Summary.** This section discusses the implementation process and analysis of style migration and face generation experiments used by deep learning framework in detail. Based on PyTorch, train CycleGAN on apple to orange and summer to winter data sets. TL-GAN training was completed on the CelebFaces Attributes Dataset, and the training model was applied to the customized test data set. Get good results finally.

(a) Apple to orange (b) Orange to apple

(c) Summer to Winter (d) Winter to Summer

FIGURE 6. CycleGAN experimental effect display

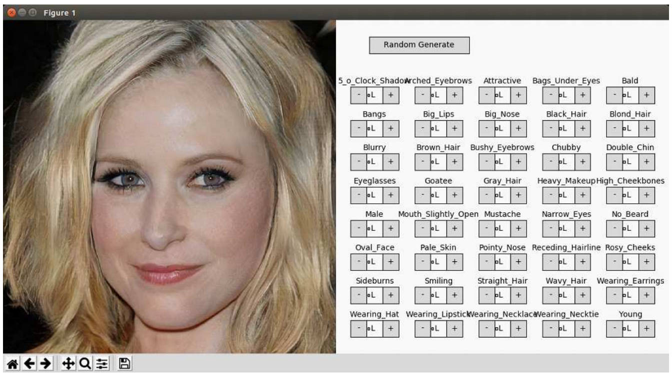
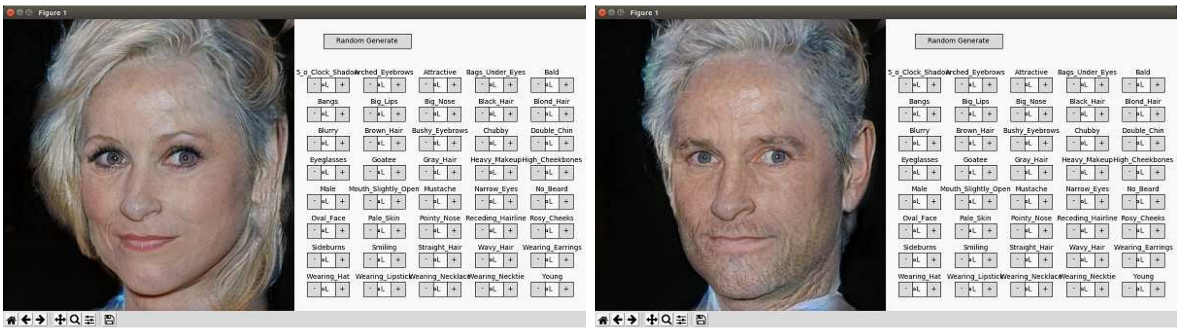
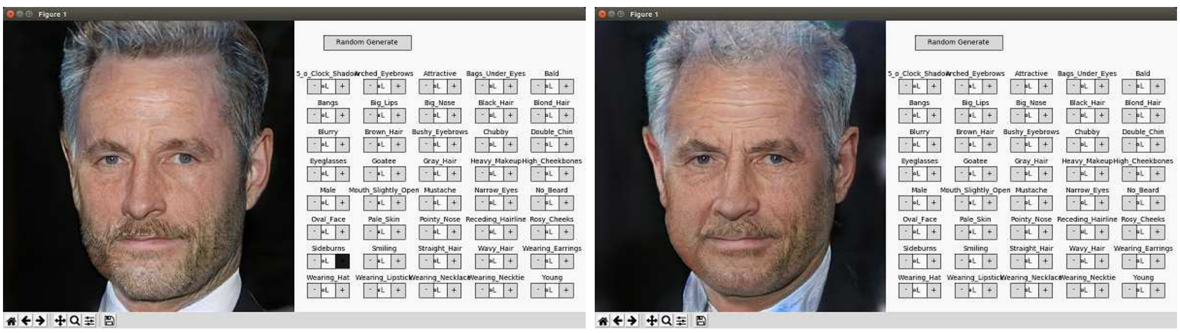



FIGURE 7. Randomly generate a face image



(a) Age increase (b) More masculine



(c) More prominent temples (d) Bigger nose

FIGURE 8. TL-GAN experimental effect display

5. **Conclusions.** This paper analyzes GAN's different models and their applications in the computer vision field. Based on extensive research literature, especially the latest developments of GAN, the basic ideas and method characteristics of various methods (GAN, WGAN, CycleGAN) are analyzed and summarized: WGAN introduces new innovation models, innovates lose function, utilizes the new loss function's continuity to solve the crash problem; CycleGAN can migrate the image content from the source domain to the target domain without paired training data. And summarize the advantages and disadvantages of GAN. This paper also shows the results of GAN in image style migration and face generation, using experiments to demonstrate GAN-based image generation technology.

The work of this paper has certain effects, but there is still some room for improvement. For example, GAN still has a lot of room for improvement in the photo authenticity, that is, the generated image meets our expected results in time, but as human beings, most of us can still see the authenticity of the picture at a glance, even though GAN already thinks the picture is real. The second point is overfitting. All GANs can be said to be overfitting and they only generate composite images inside the dataset. The above is where we think this paper needs to be improved.

**Acknowledgment.** This work is partially supported by the Xinjiang Uygur Autonomous Region Natural Science Foundation (Grant No. 2018D01A49). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

## REFERENCES

[1] K. Chen, Image recognition and generation based on artificial intelligence algorithm, *Computer Knowledge and Technology*, vol.13, pp.173-175, 2018.
[2] Y. Lin, X. Dai, L. Li, X. Wang and F. Wang, The new frontier of artificial intelligence research: Generative confrontation network, *Acta Automatica Sinica*, vol.5, pp.775-792, 2018.
[3] I. J. Goodfellow, Generative adversarial networks, *arXiv: 1701.00160*, 2016.
[4] Y. Xu, Overview of the generation of confrontation network theory models and applications, *Journal of Jinhua Vocational and Technical College*, vol.3, pp.81-88, 2017.
[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza et al., Generative adversarial nets, *International Conference on Neural Information Processing Systems*, pp.2672-2680, 2014.
[6] X. Cai, *Research on Image Generation Technology Based on DCGAN Algorithm*, Master Thesis, Qingdao Technological University, 2017.
[7] M. Arjovsky and L. Bottou, Towards principled methods for training generative adversarial networks, *International Conference on Learning Representations*, 2017.
[8] S. Huang, *Research on Super-Resolution Image Reconstruction Algorithm Based on Deep Learning*, Master Thesis, Taiyuan University of Technology, 2018.
[9] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, *arXiv: 1703.10593*, NIPS 2017 tutorial, 2017.
[10] T. Dong, *Design and Implementation of Surveillance Image HD Reconstruction System*, Master Thesis, Huazhong University of Science and Technology, 2018.
[11] Y. Xiong, *Research and Implementation of Sparse Flow Field Processing Based on Deep Learning*, Master Thesis, Beijing University of Posts and Telecommunications, 2018.