Journal of Quantum Computing
DOI:10.32604/jqc.2021.017251
Article

Tech Science Press

Implementation of Art Pictures Style Conversion with GAN

Xinlong Wu[1], Desheng Zheng[1,*], Kexin Zhang[1], Yanling Lai[1], Zhifeng Liu[1] and Zhihong Zhang[2]

[1]School of Computer Science, Southwest Petroleum University, Chengdu, 610000, China
[2]AECC Sichuan Gas Turbine Establishment, Mianyang, 621700, China
[*]Corresponding Author: Desheng Zheng. Email: zheng_de_sheng@163.com

Received: 20 July 2021; Accepted: 21 September 2021

Abstract Image conversion refers to converting an image from one style to another and ensuring that the content of the image remains unchanged. Using Generative Adversarial Networks (GAN) for image conversion can achieve good results. However, if there are enough samples, any image in the target domain can be mapped to the same set of inputs. On this basis, the Cycle Consistency Generative Adversarial Network (CycleGAN) was developed. This article verifies and discusses the advantages and disadvantages of the CycleGAN model in image style conversion. CycleGAN uses two generator networks and two discriminator networks. The purpose is to learn the mapping relationship and inverse mapping relationship between the source domain and the target domain. It can reduce the mapping and improve the quality of the generated image. Through the idea of loop, the loss of information in image style conversion is reduced. When evaluating the results of the experiment, the degree of retention of the input image content will be judged. Through the experimental results, CycleGAN can understand the artist's overall artistic style and successfully convert real landscape paintings. The advantage is that most of the content of the original picture can be retained, and only the texture line of the picture is changed to a level similar to the artist's style.

Keywords: Generative adversary network; deep learning; image style conversion; convolutional neural network; adversary learning

1 Introduction

Network information, especially media image data, has shown very exaggerated growth and great value. Image conversion refers to converting an image from one style to another and ensuring that the content of the image remains unchanged. In image style conversion, image processing involves many problems in computer vision, iconology and other fields. For example, image coloring [1–2], image inpainting [3], image high resolution [4–6], image style migration [7], and so on. Machine learning research can be applied in many ways [8–9], especially in the field of image. At present, image research has become an indispensable content in the field of computer vision.

Image conversion using GAN can be divided into two aspects: one is input data type, the other is output diversity. If there are matching data, conditional GAN (cGAN) [10] can solve this problem well. The generator and discriminator are the class labels used to generate the condition information of a specific class of images. The conditional information can make the learning of generator more advanced. The pix2pix [11] model based on cGAN can solve many problems that need to be solved by using different loss functions in the past, which is equivalent to providing a universal architecture. In pix2pix, there are two datasets, A and B, whose contents are consistent. Two datasets, one for input and one for target. Dataset A is a set of images of one style and dataset B is a set of images of another style. For example, the data set of shoes a is the real image set. Pix2pix learns the mapping of two datasets and generates images. The error between the generated image and the target is calculated by the loss function, and the image generated by adjusting the parameters is closer to the target. Based on this judgment, Coupled Generative Adversarial Networks (CoGAN) [12] proposes a weight sharing strategy. In the generator and discriminator, to achieve cross domain image conversion, we can share the weight corresponding to the high-level semantic information, so that we can learn the joint distribution in different domains. However, CoGAN sometimes has a crash problem, especially when it needs to generate high-resolution images, the probability will be large, because the vector it inputs is random.

CycleGAN [13] uses two generator networks and two discriminator networks. Its goal is to learn the mapping and inverse mapping relationship between the source domain and the target domain. It can reduce the mapping, improve the quality of the generated image, and reduce the collapse problem of GAN model. DiscoGAN [14] and DualGAN [15] put forward ideas similar to it. In this paper, the evaluation of the image quality by CycleGAN is mainly discussed in terms of the loss of information in the input image to reduce the amount of information in the input image.

2 Preliminary Knowledge

2.1 Convolution Neural Network

Convolution Neural Network (CNN) [16–17] has convolution layer, activation layer, pooling layer and full connection layer.

The convolutional layer is the core structure, in which local perception and parameter sharing can reduce the data from high-dimensional to low-dimensional, and extract outstanding features from the data.

The pooling layer can sample the input data in the space from various dimensions, thereby reducing the data range and effectively avoiding model overfitting. This is because there is no local linear transformation, which improves the generalization ability of network processing.

The function of the activation layer is to introduce nonlinear learning and processing capabilities into the convolutional neural network. The linear output of the upper layer is processed by a nonlinear activation function.

The fully connected layer is the same as the traditional multilayer perceptron model. After repeated processing of the convolutional layer and the pooling layer, the input characteristics of the fully connected layer are repeatedly refined, so the effect is better than the effect of directly using the original data as the input.

2.2 Generative Adversarial Networks

2.2.1 Basic Principle of Generative Adversarial Networks

Generally speaking, the generation countermeasure network is a game between two networks [18], as shown in Fig. 1. The main function is completed by the generation network G and the discrimination network D. the network G generating the image receives the random noise Z and generates the image G(z) with noise. Judge whether the picture is true or not. Its input parameter is x, X represents an image, and the output D(x) represents the probability that x is a real image. The closer the result is to 1, the more like the real image, and relatively close to 0, it means that it is impossible to be a real image. In this process, the direction of generating network G should be to generate real pictures as much as possible, and deal with network D in a deceptive way. The function of D is to identify the difference between the image generated by G and the real image. In this case, the network G and D together constitute a dynamic "game process" that can evolve continuously. In the best case, the generated network G can output vivid and interesting pictures G(z) that may be false. From the perspective of network D, it is difficult to judge whether the image generated by G is true, so D(G(z)) = 0.5. The two networks are constantly improving and optimizing themselves.

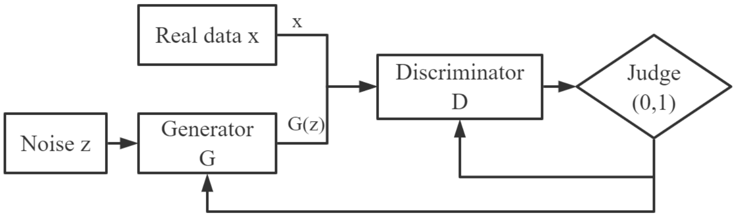

Figure 1: Topology of generate adversarial network

2.2.2 Problems in Generating Adversarial Network

In theory, adversarial training can learn and generate the same distributed output as the target domain [19]. However, model training cannot be performed only by using this loss. This is because if the sample size is large enough, any image in the target domain can be mapped to the same set of inputs, and anyone who learns this mapping can obtain any output distribution consistent with the target distribution. In short, during the mapping process, all inputs can be mapped to the same image in the target domain, which will invalidate the loss. Therefore, if only a single countermeasure loss is used, there is no guarantee that the learning function can map a single input to a desired output.

2.3 CycleGAN

CycleGAN is a model that directly points to the pain points of the industry [20]. In the image style conversion, according to the traditional method, it is very troublesome to find two pictures with the same content but different styles, and it is not easy to find. Image matching limits the development of deep learning models. The above difficulties can be summarized as follows: since model training must rely on matching images, unless such images are intentionally generated, training is impossible, and data deviation is likely to occur.

The purpose of CycleGAN training is to avoid the above-mentioned difficulties. The model is highly adaptable and can adapt to a series of visual problems, such as super-resolution, style conversion and image enhancement.

3 CycleGAN Algorithm Model

3.1 Structure of CycleGAN Network

3.1.1 Structure of CycleGAN Model

CycleGAN is composed of two networks of generators and discriminators, as shown in Fig. 2. It is cyclic in structure. X represents the image in the X domain, and Y represents the image in the Y domain. The generator G generates an image in the X domain into an image in the Y domain, and then the generator F reconstructs the original image input in the X domain. The generator F generates a Y domain image from the X domain image, and then the generator G reconstructs the input original image Y domain. The discriminators D_X and D_Y play a discriminating role to ensure the transfer of the image pattern.

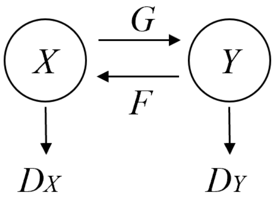

Figure 2: Structure of CycleGAN model

In the GAN model, all X can be mapped to the same picture in Y space, and CycleGAN can convert the image of Y into a picture in X space. This prevents the model from converting all X pictures to the same picture in Y space. In an ideal state, any input of real landscape photos can be converted into paintings with artistic style.

3.1.2 Network Structure of Generator

The generator network can be roughly divided into three steps, and the specific process is shown in Fig. 3. The three steps are as follows:

1. Set the input image scale (256,256,3). Firstly, the feature is extracted from the convolution layer. In order to extract more advanced convolution channels, the size of each convolution channel is reduced by half. The output of the final generator is (64,64,256).

2. There are nine residual blocks. Different channels of the output image combine different features of the image. According to these features, the feature vector of the image is transformed from the source domain to the target domain. The output scale is (64,64,256).

3. Deconvolution reconstructs the low-level features from the feature vector, and the output scale is (256,256,64). The output of the last convolution module is (256,256,3), which converts the low-level function into the image in the target domain.

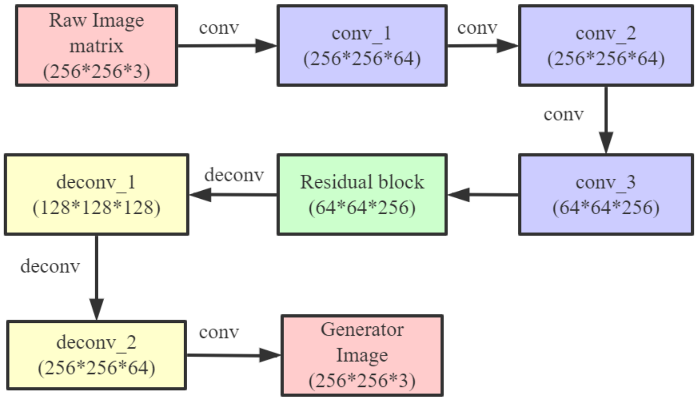

Figure 3: Network structure of generator

3.1.3 Network Structure of Discriminator

The discriminator is composed of multiple convolution layers. After extracting features from the image, it can judge whether these features belong to a specific category. The last layer of discriminator network is the convolution layer used to generate one-dimensional output. The first four convolution layers are used to extract features, and the last convolution layer is used to judge whether the image is true or false, as shown in Fig. 4.

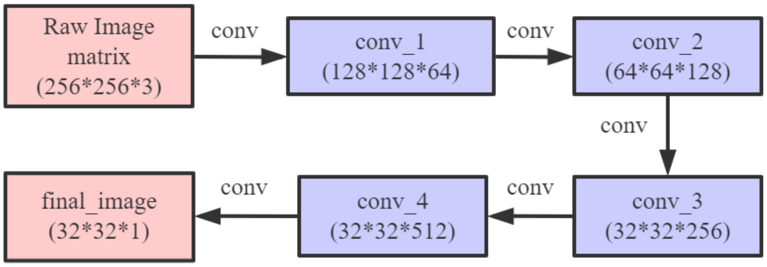

Figure 4: Network structure of discriminator

3.2 Objective Function

3.2.1 Resistance Loss Function

The loss function of generator and discriminator is the same as GAN. Discriminator D tries its best to detect the false image produced by generator G, and the generator tries to generate the image to deceive the discriminator. The function of this loss is to make the generated image more image, that is to say, the generated image is more realistic. But it is not guaranteed to produce the image we want.

The fight against loss consists of two parts, one of which is as follows:

$$\mathsf{L}_{\text{GAN}}\left(G, D_Y, X, Y\right) = \mathsf{E}_{y\sim p_{\text{data}}(y)}\left[\log D_Y(y)\right] + \mathsf{E}_{x\sim p_{\text{data}}(x)}\left[\log\left(1 - D_Y(G(x))\right]\right. \tag{1}$$

G tries to generate the image G(x) to make it closer to the target region Y in visual perception, while Dy aims to distinguish the difference between the generated image G(x) and the real image y. Then, a similar mapping function F: Y→X and its discriminator Dx are introduced.

$$\mathsf{L}_{\text{GAN}}\left(F, D_X, Y, X\right) \tag{2}$$

3.2.2 Cyclic Uniform Loss Function

In the part of loss, in addition to the classic basic GAN network against loss, a cycle loss is proposed. Because the network needs to ensure that the generated image has the characteristics of the original image, if a pseudo image is generated by generator Gx→y, another generator Gy→x should be used to restore the original image. This process must satisfy circular consistency. Cycle- loss function:

$$L_{\text{cycle}}(G, F) = \mathsf{E}_{x\sim p_{\text{data}}(x)}\left[\| F(G(x) - x)\|_1\right] + \mathsf{E}_{y\sim p_{\text{data}}(y)}\left[\| G(F(y) - y)\|_1\right] \tag{3}$$

Theoretically speaking, confrontation training can learn to map g and F to generate the same distribution results as Y and X in the target domain, respectively. However, if it has enough capacity, any image in the target domain may be mapped to by the same group of inputs, and anyone who learns this mapping can cause any output distribution consistent with the target distribution. In order to reduce the space of possible mapping functions, we think that the learned mapping functions should be cyclic consistent: for each image x from the x domain, the image conversion period should be able to make x return to the original image, that is, x→G(x)→F(G(x)) ≈ x, which is called forward loop loss.

$$x \to G(x) \to F(G(x)) \approx x \tag{4}$$

Similarly, for each image y from the y domain, the image conversion period should be able to make y return to the original image, i.e., y→F(y)→G(F(y)) ≈ y, which we call the backward loop loss. The function is as follows:

$$y \to F(y) \to G(F(y)) \approx y \tag{5}$$

The total loss function is as follows, where λ controls the relative importance of the two objectives:

$$\mathsf{L}\left(G, F, D_X, D_Y\right) = \mathsf{L}_{\text{GAN}}\left(G, D_Y, X, Y\right) + \mathsf{L}_{\text{GAN}}\left(F, D_X, Y, X\right) + \lambda \mathsf{L}_{\text{cyc}}(G, F) \tag{6}$$

3.3 CycleGAN Training Process

This section mainly introduces the training process of CycleGAN. There are two generators Gx→y and Gy→x, and two discriminator models Dx and Dy are created. The network topology of CycleGAN is shown in Fig. 5. The steps are as follows:

1. Input image set X and image set Y, after reading the image, normalize the image and convert it into 256 * 256 size picture;

2. The x-domain and y-domain images are input, and the generated x-domain and y-domain images, as well as the reconstructed x-domain and y-domain images, are obtained by generator Gx→y and Gy→x;

3. The discriminator Dx judges the generated x-domain image and the real x-domain image, Dy judges the generated y-domain image and the real-world y-domain image, and returns the judgment result;

4. Calculate the loss of generator and discriminator, and the loss in cycle;

5. Two or three steps are cycled, and the sequence of images in x domain and x domain is disturbed once every cycle until the maximum number of cycles epoch is reached.

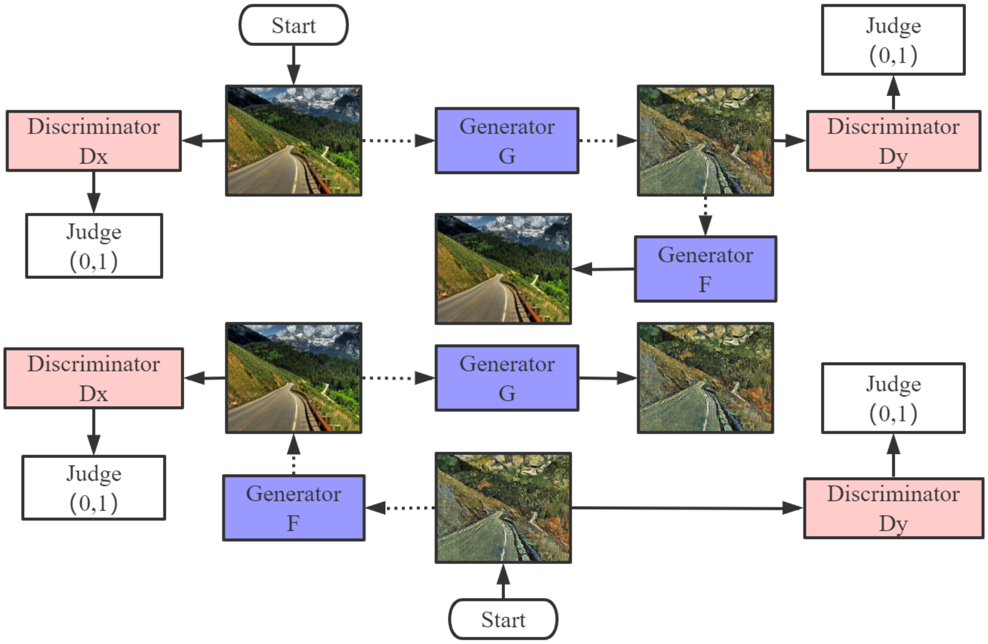

Figure 5: Topology of CycleGAN network

4 Comparison and Analysis of Experimental Results

4.1 Experimental Environment

The experimental platform configuration of this paper is shown in Table 1 as below.

Table 1: Experimental environment platform

Tool	version
Operating system	Ubuntu 16.04.5 LTS 4.15.0-48-generic GNU/Linux
CPU	Intel(R)Xeon(R)CPU E5-2660 v2
GPU	P102-100
Machine memory	62 GB
CUDA	10.2
TensorFlow	1.5.0
Python	3.6.2

4.2 Data Set Processing

The experimental data set is mainly from the network. In the training set, there are 303 Van Gogh art pictures and 300 real landscape pictures. There are 10 art pictures and 10 real landscape pictures in the test set. For example, please see Table 2.

Table 2: Data sets

TrainingA	TrainingB	testA	testB
303	300	10	10

In this experiment, OpenCV is used to process the data set, and the cv2.imread() function is used to read the images. First, the total length of the image list in the collected data set is obtained, and then the subscript of each image is obtained in turn. The path and name of each image are marked by the subscript. After the image is successfully obtained, the image is normalized, and the cv2.resize() function is used. If it is converted into 256 * 256 images, the images in X domain and Y domain should be processed respectively, and the processed images of x domain and y domain will be returned. In this way, we will make the pictures 256 * 256.

4.3 Analysis of Experimental Results of CycleGAN

4.3.1 Definition of Image Style Conversion Assessment

Evaluate the content retention of the input image, where the content is defined as belonging to a specific scene category, such as forest, street, cloud, etc. If the content can be preserved, the stylized natural image should still have the same content. For example, if a mountain image is generated in the Van Gogh style, it should still be able to be classified as a mountain after stylization. If the generated image loses its connection with the content, it will not succeed.

In the image generated by the style conversion, the image of the mountain retains the shape and content of the original natural image, and adds a Van Gogh style texture to the mountain. Natural images generated from mountain images with Van Gogh style will also modify the Van Gogh style to natural image features accordingly. It shows that the image style conversion is completed by modifying the image characteristics.

4.3.2 Analysis of Training Results

In this experiment, the results of training visualization are divided into two lines and three columns. The first line is the original image of X, the image after X style, the image converted to original image after X style, and the image converted to original image after y style, respectively.

The result of training for 50,000 times is shown in Fig. 6. For the style transformation of real pictures, we can see that the effect is relatively good, and the conversion of mountains is still the best. It can be clearly seen that Van Gogh's style, where the sea water and sky are blue, will be replaced by yellow, and the details in the figure are still not enough. For example, the house becomes fuzzy, and the house can no longer be identified Compared with the original map, it will be mistaken for a part of the mountain.

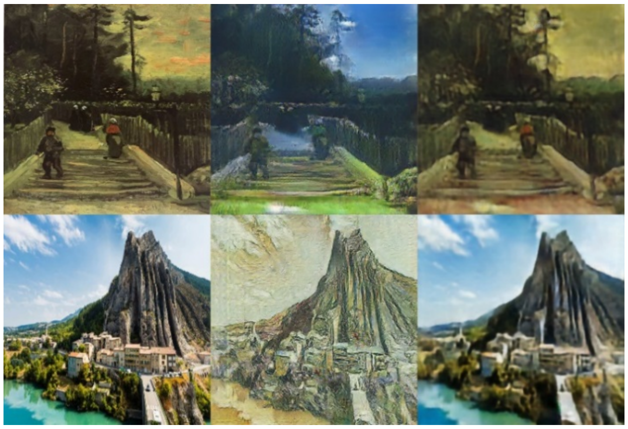

Figure 6: Training for 50,000 times

The result of 100,000 times is shown in Fig. 7. In the generated style pictures, the effect can be seen for trees, sky and white clouds, and the content of the picture has been largely retained, but there will be color problems. On the whole, it can show the style of Van Gogh.

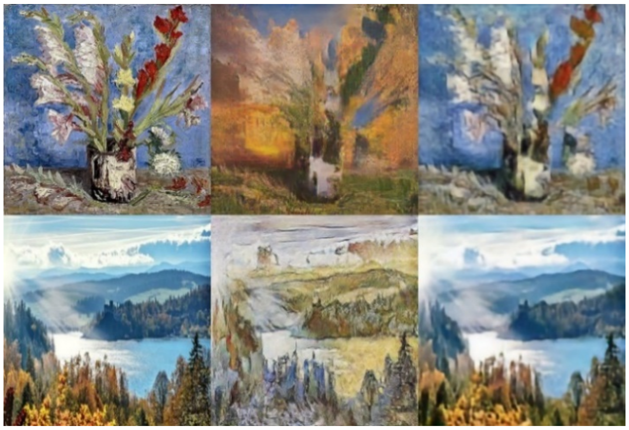

Figure 7: Training for 100,000 times

The test set map can clearly see the good effect of the transformation from real scenery to artistic style. As shown in Fig. 8.

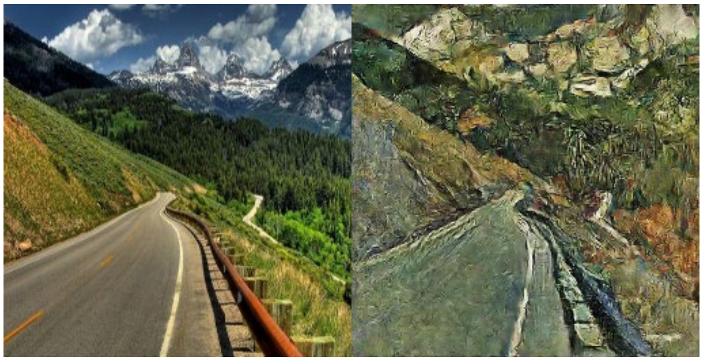

Figure 8: Test picture

The loss function curve is shown in Fig. 9. It was observed that the loss value of the discriminator showed a downward trend.

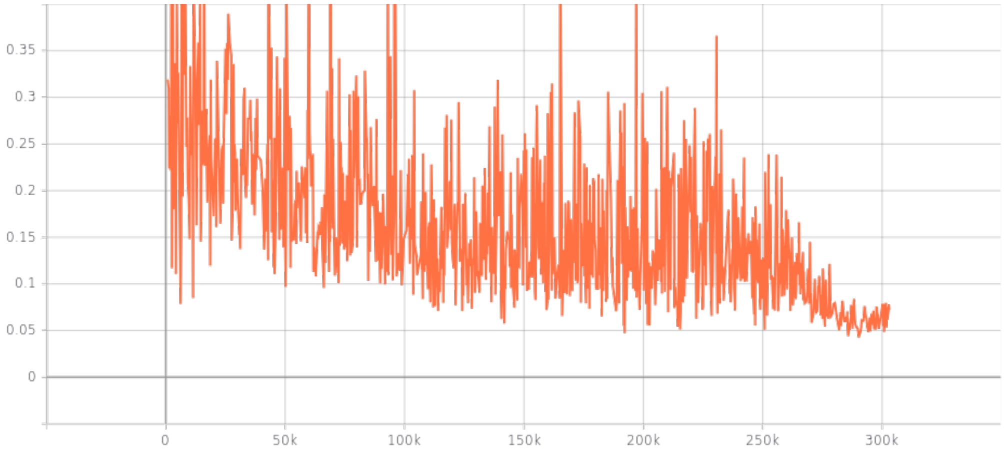

Figure 9: Total loss values of Dx and Dy

The loss curve of the generator is shown in Fig. 10, which first decreases, then stabilizes, and then rises. According to the change of the loss curve, the experiment is the best at 50K–100K.

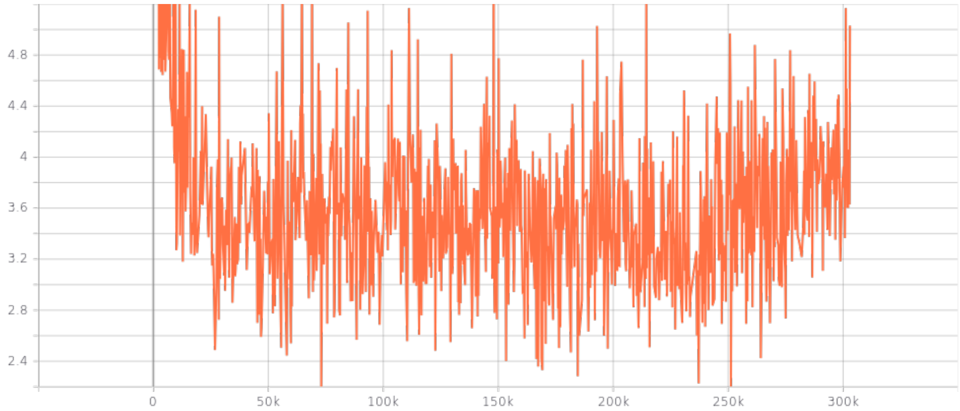

Figure 10: Loss value of generator

5 Conclusion

This paper first describes the research and development story of GAN, and then introduces its application in image style conversion. Then the network structure of CycleGAN algorithm model is introduced, and its objective function is explained. From the experimental results, CycleGAN can learn the artist's overall artistic style and successfully transform the real landscape pictures. Its advantage is that it can retain the content of the picture to a large extent, and only change the texture and line of the image to reach the level similar to the artist’s style. Experiments show that CycleGAN can complete the task of real landscape image style conversion.

Acknowledgement: We are grateful to the peoples for the support and encouragement.

Funding Statement: This work was supported by Scientific Research Starting Project of SWPU [No. 0202002131604]; Major Science and Technology Project of Sichuan Province [Nos. 8ZDZX0143, 2019YFG0424]; Ministry of Education Collaborative Education Project of China [No. 952]; Fundamental Research Project [Nos. 549, 550].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] R. Zhang, P. Isola and A. A. Efros, “Colorful image colorization,” in *Proc. ECCV,* Amsterdam, NH, NLD, pp. 649–666, 2016.

[2] Z. Cheng, Q. Yang and B. Sheng, “Deep colorization,” in *Proc. ICCV*, Santiago, CHL, pp. 415–423, 2015.

[3] D. Pathak, P. Krahenbuhl and J. Donahue, “Context encoders: Feature learning by inpainting,” in *Proc. CVPR*, Las Vegas, NV, USA, pp. 2536–2544, 2016.

[4] C. Ledig, L. Theis and F. Huszr, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proc. CVPR*, Honolulu, HI, USA, pp. 105–114, 2017.

[5] C. Dong, L. C. Chen and K. He, “Image super-resolution using deep convolutional networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[6] J. Johnson, A. Alahi and F. F. Li, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. ECCV*, Amsterdam, NH, NLD, pp. 694–711, 2016.

[7] A. L. Gatys, S. A. Ecker and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proc. CVPR*, Las Vegas, NV, USA, pp. 2414–2423, 2016.

[8] X. Y. Li, Q. S Zhu, M. Z. Zhu, Y. M. Huang, H. Wu *et al.,* “Machine learning study of the relationship between the geometric and entropy discord,” *Europhysics Letters*, vol. 127, no. 2, pp. 542–557, 2019.

[9] X. Liu and X. Chen, “A survey of GAN-generated fake faces detection method based on deep learning,” *Journal of Information Hiding and Privacy Protection*, vol. 2, no. 2, pp. 29–36, 2020.

[10] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” *Computer science*, vol. 23, no. 4, pp. 2627–2680, 2014.

[11] P. Isola, Y. J. Zhu and T. Zhou, “Image-to-image translation with conditional adversarial networks,” in *Proc. CVPR*, Honolulu, HI, USA, pp. 5967–5976, 2017.

[12] Y. M. Liu and O. Tuzel, “Coupled generative adversarial networks,” in *Proc. NIPS*, Barcelona, Catalan, ESP, pp. 469–477, 2016.

[13] Y. J. Zhu, T. Park and P. Isola, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proc. ICCV*, Venice, Veneto, ITA, pp. 2242–2251, 2017.

[14] T. Kim, M. Cha, H. Kim, K. J. Lee and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” in *Proc. ICML*, Sydney, NSW, AUS, pp. 1857–1865, 2017.

[15] Z. Yi, H. Zhang and P. Tan, “DualGAN: Unsupervised dual learning for image-to-image translation,” in *Proc. ICCV*, Venice, Veneto, ITA, pp. 2868–2876, 2017.

[16] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[17] L. Y. Cun, B. Boser and S. J. Denker, “Handwritten digit recognition with a back-propagation network,” *Advances in Neural Information Processing Systems*, vol. 2, no. 2, pp. 396–404, 1990.

[18] Y. Z. He, *Gan Learning Guide: Introduction to Demo Generation.* Beijing, China: Zhihu, 2017. [Online]. Available: https://zhuanlan.zhihu.com/p/24767059.

[19] W. Z. Liang, *Take You to Understand CycleGan and Implement It Easily with TensorFlow.* Beijing, China: Zhihu, 2017. [Online]. Available: https://zhuanlan.zhihu.com/p/27145954.

[20] F. X. Xiao, *Brief Introduction of Principle about CycleGan Model.* Beijing, China: CSDN, 2018. [Online]. Available: https://blog.csdn.net/xiaoxifei/article/details/83830842.