Chinese Calligraphy Generation Based on Residual Dense Network

Yalin Miao
Xi'an University of Technology
Xi'an ,China
myl@xaut.edu.cn

Huanhuan Jia*
Xi'an University of Technology
Xi'an ,China
1607421431@qq.com

Kaixu Tang
Xi'an University of Technology
Xi'an ,China
771581211@qq.com

Wenfang Cheng
Xi'an University of Technology
Xi'an ,China
1666547760@qq.com

ABSTRACT

Chinese calligraphy, as a writing art of Chinese characters, plays an important role in the inheritance of traditional culture. Compared with simple English fonts, the generation of Chinese calligraphy fonts is more challenging. This paper proposes Chinese calligraphy generation based on residual dense network. The generator network combines residual network structure and dense network structure, and uses local residual learning and global feature fusion to enhance feature reuse and information continuous transmission. The discriminator network adopts an autoencoder structure, and restores the similarity between the distributions by an approximate convergence strategy to achieve fast and stable training. At the same time, the structural similarity loss is introduced to quantify the similarity between the generated font image and the real font image. Experiments show that the proposed method achieves good calligraphy font generation effect, which effectively improves the quality of the generated font images and the rate of training.

CCS Concepts

• Computing methodologies→Computer graphics→Image manipulation→Image processing • Computing methodologies →Artificial intelligence→Computer vision→Computer vision representations

Keywords

Calligraphy generation; generative adversarial network; residual dense network

1. INTRODUCTION

Calligraphy is the artistic treasure of Chinese traditional culture. Because of the characteristics of the long history and difficult preservation of calligraphy works, the calligraphys of many famous calligraphers have been damaged or even lost, and the authentic works of calligraphers are not available [1,2]. With the rapid development of artificial intelligence technology, it is possible for computers to help people reproduce famous calligraphys. Therefore, the use of deep learning technology to generate Chinese calligraphy fonts is of great significance for promoting the digitalization research of Chinese traditional culture.

With the development of deep learning, researchers began to study the font modeling in images, and trained the network to learn the mapping of the source fonts to the target fonts [3]. Baluja et al. [4] proposed an English font generation method based on deep convolutional neural network (DCNN). The network generates the remaining characters of the same style by learning the characteristics of four letters of a certain font. Bhunia et al. [5] combined long short term memory (LSTM) with generative adversarial network, which can process character images of arbitrary width effectively. Azadi et al. [6] proposed multi-content GAN model, which generates style fonts through the combined training of font shape and texture, and transfers the style of a group of letters from A to Z.

Compared with simple English fonts, the style transfer of Chinese Fonts is more challenging. Chang et al. [7] used full convolution network (FCN) and U-Net skip connection strategy to restore the details of characters effectively. The Rewrite method [8] uses convolutional neural network (CNN) architecture to generate fonts, but the generated images are usually very fuzzy and the effect of more stylistic font processing is not good. The Zi2Zi method [9] proposes "class embedding", which combines the untrained gaussian noise as style embedding and Chinese character embedding in series. Sun et al. [10] embedded the multi-scale refined information pyramid into U-Net, which enables the generator to save more detailed information. Because of the various forms and more complex structure of calligraphy fonts, deep learning has less application in the generation of calligraphy fonts.

In recent years, generative adversarial network (GAN) [11] has received great attention. It generates high quality images through

*Corresponding author

ICIIP'2019, November 16–17, 2019, China, China

ACM ISBN 978-1-4503-6191-0/19/11…$15.00
DOI:https://doi.org/10.1145/3378065.3378159

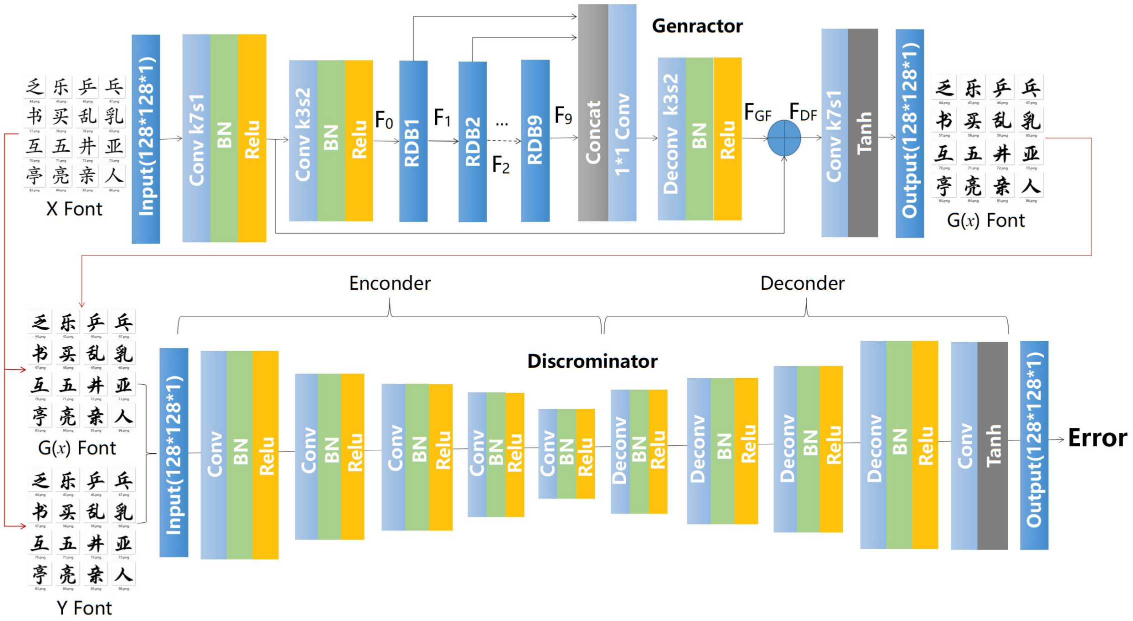


Figure 1. Calligraphy font generation network structure

adversarial learning of generator network and discriminator network. At the same time, many improved models of GAN have been derived. Conditional generative adversarial network (CGAN)[12] adds preconditions to input data to prevent training collapse. Boundary equilibrium generative adversarial network (BEGAN) [13] uses an autoencoder as discriminator, and adds additional equalization process to the training of balance generator and discriminator.

This paper introduces the method of confrontation training, and proposes Chinese calligraphy generation based on residual dense network. The residual dense network combines the residual network and the dense convolutional network. It can make full use of all hierarchical features extracted by different units in the model. It not only overcomes the problem of gradient disappearance through short connections, but also extracts the detailed information in the image, enhances feature reuse and information continuous transmission, and generates realistic target fonts. Through the adversarial training of generator network and discriminator network, the generation of Calligraphy fonts are realized end to end.

2. THE PROPOSED NETWORK MODEL

This paper proposes Chinese calligraphy generation method based on residual dense network. The generator introduces residual dense blocks, which make each layer tightly connected, and combines the input and output of each RDB to train the wider network and enhance the utilization of features. The discriminator adopts the autoencoder structure and matches the loss distribution of the autoencoder based on the loss of Wasserstein distance, and adds an additional equalization process in the adversarial training to balance the generator and the discriminator. The generator is guided through the loss function to continue to improve the generated effect until it is close to the real image. The overall network structure is shown in Figure 1.

2.1 Residual Dense Block

The study found that as the depth of the network increases, training is more difficult, and gradient explosion or gradient disappearance is prone to occur. He et al. [14] first proposed the residual network (ResNet), which connected each layer with a short circuit of the previous layer. The skip connection makes the data transmission between the networks smoother and improves the underfitting phenomenon caused by the disappearance of the gradient. DenseNet [15] is that each layer is spliced together with all the previous layers in the channel dimension. Compared with Resnet, Densenet proposes a more radical dense connection mechanism, that is, each layer accepts all the previous layers as additional inputs, and the dense connection effectively alleviates the gradient disappearance problem and enhances feature propagation.

Zhang et al. [16] proposed residual dense network (RDN) based on dense networks. The residual dense block (RDB) is building module of the RDN. The RDB module is composed of feature extraction units consisting of convolution layers and activation layers, which are repeatedly connected in series. The residual dense block integrates the residual block and the dense block, reads the previous RDB state through continuous memory mechanism, and fully utilizes the features of each convolutional layer through local dense connections to retain the accumulated features adaptively. The residual dense block in this paper consists of 6 convolutional layers, Relu activation functions and a 1*1 local feature fusion layer. The number of convolution kernels in each layer is 64. Different from the traditional residual block, each layer has BatchNorm layer removed. Because the use of BatchNorm to normalize features will increase the network parameters and consume more memory, which is not conducive to convergence. The residual dense block structure is shown in Figure 2.

In an RDB, the output for the 6-th convolutional layer is:

$$F_{n,6} = \sigma(W_{n,6}[F_{n-1}, F_{n,1}, \ldots, F_{n,6}]) \qquad (1)$$

Where σ represents the Relu activation function. $W_{n,6}$ represents the 6-th convolutional operation in the n-th RDB block. $F_{n\text{-}1}$ represents the output of the n-1th RDB, and

$[F_{n,1},...,F_{n,6}]$ represents the output of the previous 6 convolutional layers through dense connection.

$$F_{n,LF} = H_{LFF}^{n}([F_{n-1}, F_{n,1},..., F_{n,6}]) \quad (2)$$

Where H_{LFF}^{n} represents the convolutional operation of 1×1 in the n-th RDB block, which is used to compress the dimensions of the output, and reduces the parameter growth caused by feature fusion in the residual block.

$$F_n = F_{n-1} + F_{n,LF} \quad (3)$$

In order to make full use of the feature information and maintain the state of the gradient, F_n performs a skip connection between the $F_{n,LF}$ and the output of the previous RDB while the feature map passes through the residual block. By integrating the previous RDB output information with the current RDB output feature, the hierarchical information is guaranteed not to be lost and local residual learning is formed.

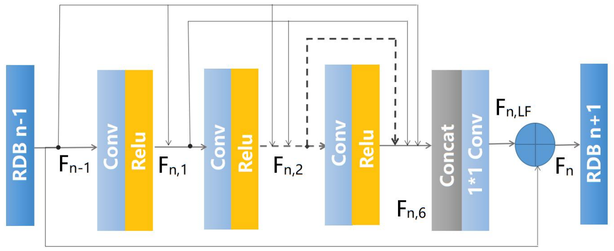


Figure 2. RDB network structure

2.2 Generator and Discriminator

The generator network draws on the idea of CGAN [12]. X Font is used as conditional input and does not need the input of noise. As shown in Figure 1, the shallow feature F_0 is first extracted using a combination of two convolutional layers (Conv), BatchNorm layers (BN), and Relu activation functions (Relu). The core migration module consists of 9 RDB structures and a 1×1 convolution layer. The number of convolution kernel in each layer is 64, which is different between each layer. Global information integration of RDB can more effectively learn more features from previous and current local features, and stabilize the training of a wider network. Global Feature Fusion (GFF) splices the output of 9 RDB:

$$F_{GF} = H_{GFF}([F_1,..., F_9]) \quad (4)$$

$$F_{DF} = F_0 + F_{GF} \quad (5)$$

While ensuring the deep structure, in order to ensure the maximum information flow between each layer in the network, a short path method of residual connection is adopted in the network. The shallow feature F_0 and the highest layer global fusion feature F_{GF} are connected to obtain the F_{DF}, and the final convolution layer restores the feature vector to an image and generates the target font.

The discriminator network draws on the idea of BEGAN [13]. It consists of an autoencoder. The encoder and decoder are composed of four convolution layers, BatchNorm and Relu activation functions respectively. The encoder generates an image close to the real sample as much as possible, estimates the distance between the errors of the distribution, and the closer the error distribution is, the more realistic the generated image is. Through the alternating training of the generator and the discriminator, more realistic high quality calligraphy fonts images are finally obtained.

2.3 Loss Function

In order to further optimize the accuracy of GAN, the generator loss function in this paper is composed of structural similarity loss, reconstruction loss and pixel loss. Discriminant loss refers to discriminant loss function of BEGAN to construct the details of high frequency part.

The L1 loss function is introduced as the pixel loss to measure the difference of image pixel level, which makes the network pay attention to the information of image features and take into account the reconstruction of image pixel information.

$$L_1(G) = E_{x,y\sim P_{data}(x,y)}[\| y - G(x) \|_1] \quad (6)$$

In order to measure the differences between generated font images and real font images, structural similarity loss is introduced to quantify the similarity between them. x' represents generated fonts $G(x)$ and y represents real fonts. The structural similarity loss is:

$$SSIM(x', y) = \frac{(2\mu_{x'}\mu_y + c_1)(2\sigma_{x'y} + c_2)}{(\mu_{x'}^2 + \mu_y^2 + c_1)(\sigma_{x'}^2 + \sigma_y^2 + c_2)} \quad (7)$$

$$L_{SSIM}(G) = E_{x,y,l\sim P_{data}(x,y,l)}[1 - SSIM(y, G(x))] \quad (8)$$

Where μ_x and μ_y are the average of the images. σ_x^2 and σ_y^2 are the variance of the images. σ_{xy} is the covariance; c_1 and c_2 are used to maintain a stable constant.

The reconstruction loss is the generator loss of the original BEGAN, indicating the similarity between the generated image and the output image after discriminator D:

$$L(G(x)) = \|G(x) - D(x, G(x))\|_1 \quad (9)$$

The pixel loss, structural similarity loss and the reconstruction loss are weighted and superimposed to obtain loss function of the generated network. λ_{L_1} and λ_{SSIM} represent the weight parameters.

$$L(G) = L(G(x)) + \lambda_{L_1} \times L_1(G) + \lambda_{SSIM} \times L_{SSIM}(G) \quad (10)$$

Discriminant loss is a conditional constraint on BEGAN discriminant loss, and the similarity between distributions is restored by estimating the similarity between distribution errors. The loss function is derived from Wasserstein distance [17] and the loss of reconstructed real image and generated image, matching the loss distribution of the autoencoder. Wasserstein distance can be expressed as:

$$W(\mu_1, \mu_2) = \inf_{\gamma\sim\Gamma(\mu_1, mu_2)} E_{(x_1,x_2)\sim\gamma}\left[\|x_1 - x_2\|\right] \quad (11)$$

$$L(x) = \|y - D(x, y)\|_1 \quad (12)$$

$$L(D) = L(x) - k_t \cdot L(G(x)) \quad (13)$$

$L(\bullet)$ represents the pixel error output after the image and discriminator D. Because the structure of the discriminator is essentially an autoencoder, that is, both the input and the output are pictures. In order to reduce the error, calculate the distance

$\|y - D(x, y)\|_1$ between the real font Y Font and the image $D(x, y)$ output by the discriminator. When the error of the real image is the same as that of the generated image, the training is completed. The balance variable K_t in the proportional control theory is introduced to balance the generator and the discriminator, and the loss of the discriminator is optimized by modifying the K_t. An indicator M_{global} for judging convergence is designed, and a global convergence metric is derived using the equilibrium concept:

$$k_{t+1} = k_t + \lambda_k(\gamma L(x) - L(G(x))) \tag{14}$$

$$M_{global} = L(x) + |\gamma L(x) - L(G(x))| \tag{15}$$

Where $k_t \in [0,1]$, and λ_k is the learning rate of k. $\gamma \in$ [0,1], and γ is a proportional coefficient used to balance the quality and diversity of generated images. The smaller γ, the poorer the diversity, and the higher the quality of production.

3. EXPERIMENTS

3.1 Experimental Configuration and Datasets

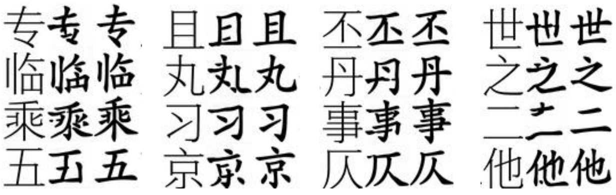
This paper is based on the deep learning framework Tensorflow. The font documents used in the study comes from the open founder calligraphy font library. The TrueType font decuments are decoded by Python script to construct the sample data sets. The single character of each font is processed as a gray image in 128×128 png format.

During the training period, the weight of the loss function is set to discriminate the loss function. Adma optimization algorithm (β_1= 0.5) is used to optimize the network parameters in the training process. The learning rate of the network model is set to 0.0002 and the number of iterations is 200. In the process of parameter adjustment, the generator G and discriminator D are alternately optimized in a ratio of 1:1.

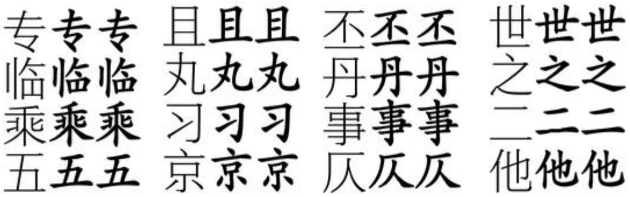
3.2 Analysis of Experimental Results

In the experimental part of this paper, we test the font generation of the test sets. In this paper, two objective evaluation indicators, peak signal to noise ratio (PSNR) and structural similarity index (SSIM), are used as comprehensive evaluation criterias to measure the font generation performance of the algorithm.

The SSIM loss function improves the distribution similarity between the generated image and the target image. In this paper, the song typeface is used as the source font and the regular script is the target font. The model with SSIM loss function and without SSIM loss function are trained respectively, and the experimental results are compared. The experimental results are shown in Figure 3. The left images of each group of fonts are the original fonts, the middle images are the generated fonts, and the right images are the target fonts. Combined with the objective evaluation index of Table 1, the font structures generated without SSIM loss function are incomplete. The generated fonts in this paper are complete in content and the effects are realistic.



(a) The result of without L_1 loss function



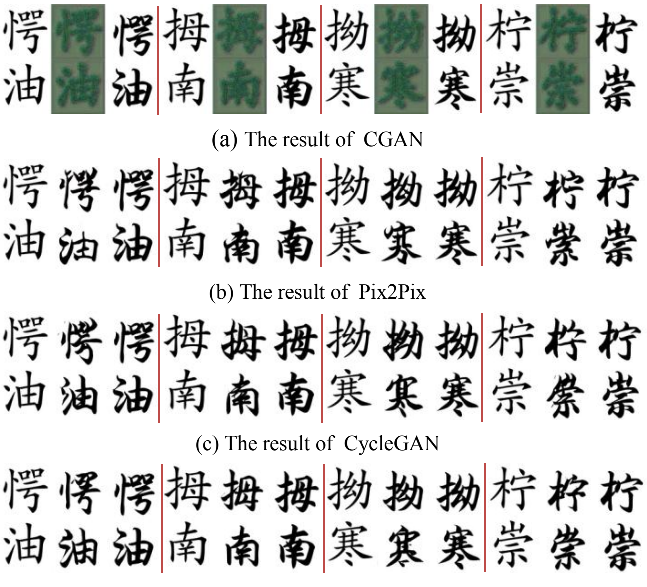
(b) The result of with L_1 loss function

Figure 3. Experimental effects with or without L_1 loss function

Table 1. Objective evaluation indicators with or without L_1 loss function

Loss	PSNR	SSIM
Without SSIM loss	11.772	0.732
With SSIM loss	25.455	0.957

Experimental comparisons are made between CGAN, pix2pix [18], CycleGAN [19], ResNet-6, DenseNet-5 and the method proposed in this paper. Pix2pix and CycleGAN are used for image style transfer. ResNet-6 and DenseNet-5 respectively adopt 6 residual blocks and 5 dense blocks to generate calligraphy fonts on the generator network of this paper. Compared with regular script, running script has the characteristics of continuous writing, and the generation of running script is more challenging. The experimental effect is shown in Figure 4. CGAN cannot reconstruct Chinese characters. Pix2Pix method and CycleGAN method can learn the content and style characteristic of Chinese characters, but due to the complex structure of concatenation writing, the effect of generating fonts is poor. ResNet-6 generate the effect with missing stroke and deformation, Such as "e" and "han". DenseNet-5 method has a relatively complete effect structure, but it needs to be improved for the generation of font details. Combined with the objective evaluation indexes in Table 2, the generated calligraphy fonts in this paper have the complete structures and realistic details. The PSNR value of the same font generated is above 23dB, and the SSIM value is above 0.93. The generation effect of calligraphy font is the best.

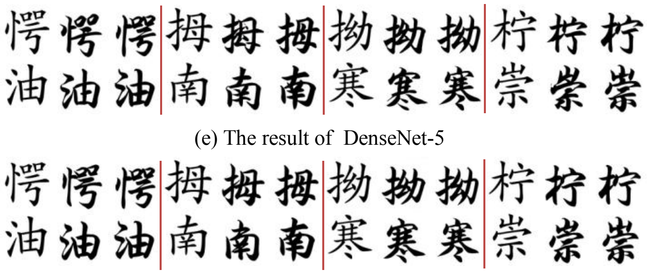


(a) The result of CGAN

(b) The result of Pix2Pix

(c) The result of CycleGAN

(d) The result of ResNet-6



(e) The result of DenseNet-5

(f) The result of this method

Figure 4. The comparison of different methods

Table 2. Objective evaluation indicators of different methods

Methods	PSNR	SSIM
CGAN	4.297	0.304
Rewrite	7.663	0.417
pix2pix	8.027	0.566
ResNet-6	11.429	0.728
DenseNet-5	19.378	0.912
This method	23.336	0.939

4. SUMMARY

This paper presents a method of Chinese calligraphy generation based on residual dense network. Residual Dense blocks are introduced into the generator network to make dense connections between the networks of each layer. Through local residual learning and global feature fusion, the features of all layers are fully utilized and retained, which is conducive to the recovery of more high frequency information. The discriminator network refers to the thought of BEGAN and restores the similarity between distributions through approximate convergence strategy, which realizes the rapid and stable training and achieved high visual quality, and then effectively generated high quality calligraphy fonts. However, there are still minor differences between the generated font and the target font. In the future work, we should seek a better network optimization strategy, in order to further improve the generation effect of calligraphy fonts.

5. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of Shaanxi, China [Grant numbers Nos. 2017050NC /NY011(1)]. We thank senior engineer Yilong Xiao for excellent technical support and value able discussion.

6. REFERENCES

[1] Yu Kai. Research on Some Key Technologies of Computer Calligraphy [D]. Zhejiang University, 2010.

[2] Du Xueying. Research and application of Chinese calligraphy AI [D]. Zhejiang University, 2018.

[3] Atarsaikhan G, Iwana B K, Narusawa A, et al. Neural Font Style Transfer[C]. *Iapr International Conference on Document Analysis & Recognition*. 2017.

[4] Baluja S. Learning typographic style[J]. arXiv preprint arXiv:1603.04000, 2016.

[5] Bhunia A K, Banerjee P, et al. Word level font-to-font image translation using convolutional recurrent generative adversarial networks[C]. *2018 24th International Conference on Pattern Recognition (ICPR). IEEE*, 2018: 3645-3650.

[6] Azadi S, Fisher M, Kim V, et al. Multi-Content GAN for Few-Shot Font Style Transfer[J]. 2017.

[7] Chang J, Gu Y. Chinese typography transfer[J]. *arXiv preprint arXiv:1707.04904*, 2017.

[8] Tian. ReWrite. Retrieved from https://github.com/kaonashi-tyc/Rewrite/.2016.

[9] Tian. ReWrite. Retrieved from https://github.com/kaonashi-tyc/zi2zi/.2017.

[10] Sun D, Zhang Q, Yang J. Pyramid Embedded Generative Adversarial Network for Automated Font Generation [C]. *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018: 976-981.

[11] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]. A*dvances in neural information processing systems*. 2014: 2672-2680.

[12] Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411. 1784, 2014.

[13] Berthelot D, Schumm T, Metz L. Began: Boundary equilibrium generative adversarial networks[J]. arXiv preprint arXiv:1703.10717, 2017.

[14] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016: 770-778.

[15] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten.Densely connected convolutional networks. CVPR, 2017.

[16] Zhang Y, Tian Y, Kong Y, et al. Residual dense network for image super-resolution[C]. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018: 2472-2481.

[17] Arjovsky M, Chintala S, Bottou L. Wasserstein GAN[J]. 2017.

[18] Isola P, Zhu J Y, Zhou T, et al. Image-to-Image Translation with Conditional Adversarial Networks[J]. 2016:5967-5976.

[19] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[C]. *Proceedings of the IEEE international conference on computer vision.* 2017: 2223-2232.