Pattern Analysis and Applications (2021) 24:1533–1547
https://doi.org/10.1007/s10044-021-01003-w

THEORETICAL ADVANCES

Chinese font migration combining local and global features learning

Yalin Miao[1] · Huanhuan Jia[1] · Kaixu Tang[1]

Received: 15 January 2021 / Accepted: 8 June 2021 / Published online: 25 June 2021

Abstract
At present, deep learning has made great progress in the field of glyph modeling. However, existing methods of font generation have some problems, such as missing stroke, structural deformation, artifact and blur. To solve these problems, this paper proposes Chinese font style migration combining local and global feature learning (FTFNet). The model uses skipping connection and dense connection mechanism to enhance the information transfer between the network layers. At the same time, feature attention layer is introduced to capture the dependency relationship between local and global features. So as to achieve the purpose of strengthening local feature learning and global feature fusion. Experiments show that the method in this paper has better performance in the details of font generation, which simplifies the font generation process and improves the quality of generated fonts.

Keywords Chinese font · Style migration · Residual Dense network · Feature attention · Conditional generative adversarial network

1 Introduction

With the rapid rise of Internet media, people will be exposed to a variety of fonts in social intercourse and office work, and the demand for multi-style personalized Chinese font applications has increased. However, the number of Chinese characters is huge and varied. The traditional font production method relies on the splitting and reorganization of the font strokes or the deformation and matching of the font skeleton. They are greatly affected by the priori knowledge, which requires manual supervision and intervention, and the algorithm is complex and inefficient. Therefore, it is of great significance to explore more efficient font design methods and develop various styles of fonts through artificial intelligence.

The early research on glyph migration mainly focused on separating the skeleton content and style of the fonts, deforming and matching the skeletons of the two fonts, extracting the features of the style and mapping them to the deformed skeletons to generate the style fonts [1]. The skeleton deformation algorithm has the disadvantages of complicated algorithm flow and low efficiency in generating fonts. Later, it was proposed to decompose the strokes of Chinese characters, map and match the strokes of the two fonts, and recombine the style fonts with the matched strokes [2, 3]. However, the process of decomposition and reorganization of strokes is cumbersome and greatly affected by prior knowledge.

With the development of deep learning, researchers began to study glyph modeling in images, and train the network to learn the mapping from source font to target font [4, 5]. Atarsaikhan et al. [6] applied natural style transfer to font generation, and adjusted the network model through style loss and content loss to achieve the style transfer of English letters. Paul et al. [7] proposed an improved variational autoencoder, which captures pixel covariance by structuring similar targets, and generate a group of letter images with similar style from a single letter. Baluja et al. [8] proposed an English font generation method based on deep neural networks. The network generates the remaining characters of the same style by learning the characteristics of the four letters of a certain font, but the edges of the generated characters are blurred. Kumarbhunia et al. [9]. used deep recurrent convolutional neural networks to effectively process character images of arbitrary width, which train word images end to end, and maintain the consistency of the final image.

Compared with simple English characters and Arabic characters, the complex and diverse Chinese font style transfer is more challenging. Zhang et al. proposed a framework

✉ Huanhuan Jia
bessie_jh@163.com

[1] Xi'an University of Technology, Xi'an, China

using recurrent neural network (RNN) as the discriminative model for Chinese character recognition and the generative model for Chinese character generation. However, it is not ideal for generating complex glyphs [10]. The Rewrite method [11] designed convolutional network structure that can generate relatively standard fonts, but users need to write thousands of Chinese characters. The effect is not good for writing scribbles and having a large difference from the reference font style. The Zi2Zi method [12] will consider both Chinese character and style embedding to generate target character, but it still has fuzzy and false edges. The quality of the generated glyphs is not high enough. Lyu et al. [13] proposed a network model (AEGG) based on encoder and decoder, which can synthesize calligraphy images of specific styles from standard Chinese font images. However, the stroke style of calligraphy fonts is distorted greatly. Chang et al. [14] proposed a handwritten font generation method (HCCG) using CycleGAN [15]. As the main framework, non-paired data sets of source domain and target domain are used for training to generate handwritten Chinese character fonts. But the generated handwriting still has the phenomena of missing strokes and redundant strokes. Jiang et al. [16] proposed an automatic method for handwritten Chinese character (DCFont). It consists of font feature reconstruction network and font style conversion network which connect content and style representation with category embedding to generate handwritten fonts.

Aiming at the problems of the existing methods, this paper proposes Chinese font migration combining local and global feature learning (FTFNet). Each Chinese character is treated as a picture. Through local residual learning and global feature fusion, we can better learn the stroke detail features of the style font. The main contributions of this paper are follows:

(1) Use the residual dense block as the core conversion module, and combine skipping connection and dense connection mechanism to enhance the information transfer between the network layers.
(2) Introduce feature attention mechanism in the upsampling layer, as supplement to the convolutional network, to capture the relationship between long distance pixels in the image.
(3) Combine adversarial loss based on Wasserstein distance, pixel loss, perceptual loss and structural consistency loss to jointly stabilize network training.

2 Related work

2.1 Pix2Pix network

Conditional generative adversarial network (CGAN) [17] adds conditional expansion to the basis of the original generative adversarial network (GAN) [18]. The input of the model is the random variable z and the condition variable x. The added information x is used to add constraints to the model to guide the data generation process. The generator G needs to generate samples matching the real data y, and the discriminator D not only needs to discriminate whether the image is real, but also discriminates whether the image matches the condition x.

In order to realize the transformation from image to image, Pix2Pix [19] adopts the idea of CGAN. At this time, the additional information in the generator is no longer the label information, but the image that needs to be transformed. Pix2Pix uses paired training data. In the image conversion task, one training data can be expressed as a set of two pictures $\{x, y\}$, where x is the source image and y is the target image. The input of the generator G is no longer random noise vector z, but source image x that needs to be converted. The discriminator D no longer judges the true or false of single picture, but the true or false of pairs of data $\{x, y\}$ and $\{x, G(x)\}$.

2.2 Residual dense network

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. [20] first proposed the residual network (ResNet), which connected each layer with a short circuit of the previous layer. As shown in Fig. 1, the jump connection makes the data transmission between the networks smoother and improves the underfitting phenomenon caused by the disappearance of the gradient. DenseNet [21] is that each layer is spliced together with all the previous layers in the channel dimension, as shown in Fig. 2. Compared with Resnet, Densenet proposes a more aggressive 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. [22] 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 unit composed of convolution layer and activation layer, which is repeatedly connected in series, and the structure of residual dense block is as shown in Fig. 3 The residual dense block integrates the residual block and the dense block, reads the previous

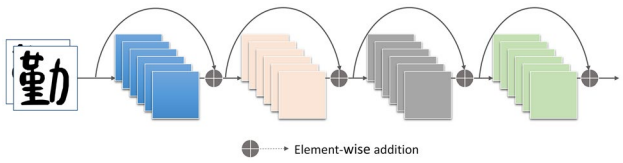

Fig. 1 Short-circuit Connection Mechanism of ResNet

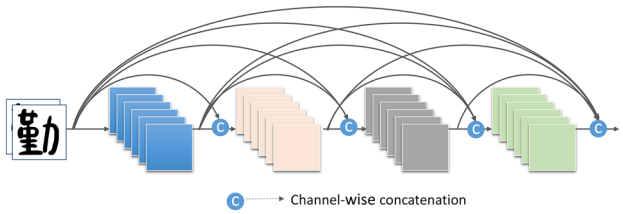

Fig. 2 Dense Connection Mechanism of Dense net

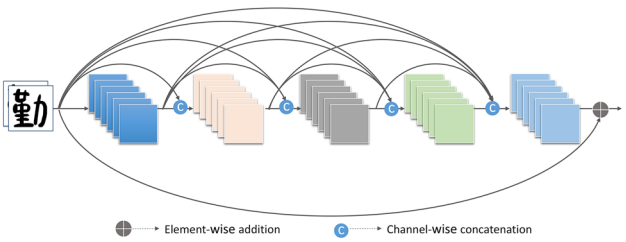

Fig. 3 Residual Dense Block

RDB state through continuous memory mechanism, and fully utilizes the features of each convolution layer through local dense connections to retain the accumulated features adaptively.

3 Artfont generation model

This paper proposes Chinese font migration network combining local and global feature learning (FTFNet). In order to improve the authenticity of the generated image in the process of image domain conversion and obtain high quality font migration images, the overall network structure is shown in Fig. 4. The whole framework consists of two subnetworks: generator and discriminator, and the corresponding target loss functions.

(1) The generator network structure is composed of encoder, residual dense block and decoder. The encoder consists of convolutional neural network. Residual dense blocks are used as core conversion modules. The decoder consists of deconvolution and feature attention layer. While maintaining the font structure information, the Chinese character style information is changed.

(2) The discriminator is composed of patch-based network structure. The discriminator network performs authenticity discrimination on the generated font image data and real font image data, and uses the numbers in the range [0, 1] to measure the similarity between the generated image and the real image.

(3) To guide the generation of more realistic style fonts, we combine pixel loss, perception loss and structural consistency loss to form a generator loss function. And we use the target loss of WGAN-GP as the discriminant loss function. When the similarity between the real image and the generated image is low, a large loss function can be generated. The discriminator trains the generator to a better direction by monitoring the loss function. Finally, the model generation is closer to the distribution of real images and improves the quality of generated fonts.

3.1 Generative network

The input layer of the generator sends the target font as the label information and the source font to the encoder together. The encoder extracts the content and style characteristics of the font image. The encoder consists of 3 convolutional layers (Conv), BatchNorm (BN) and ReLU activation function. The conversion module consists of 6 residual dense blocks. Residual dense blocks enhance the transfer of features between layers and make more efficient use of features. The decoder includes two deconvolution layers (Deconv), BatchNorm, ReLU activation function and feature attention layer. The last convolution layer uses the Tanh activation function to output the generated style font image.

(1) Local residual learning

As shown in Fig. 5, the residual dense block in this paper is composed of 6 convolutional layers, ReLU activation function and 1×1 local feature fusion layer. The BN layer is removed on the basis of the original residual block, thus reducing the memory footprint and increasing the number of network parameters. The preceding convolutional layer has a direct connection to the output of each subsequent layer, while jump connections are added at the beginning and last layer. This not only retains the feed-forward property, but also fully extracts the local feature layer information and improves the capacity of the network. In an RDB, there are multiple convolutional layers, and the output for the i-th convolutional layer is:

$$F_{n,i} = \sigma(W_{n,i}[F_{n-1}, F_{n,1}, ..., F_{n,i-1}]) \quad (1)$$

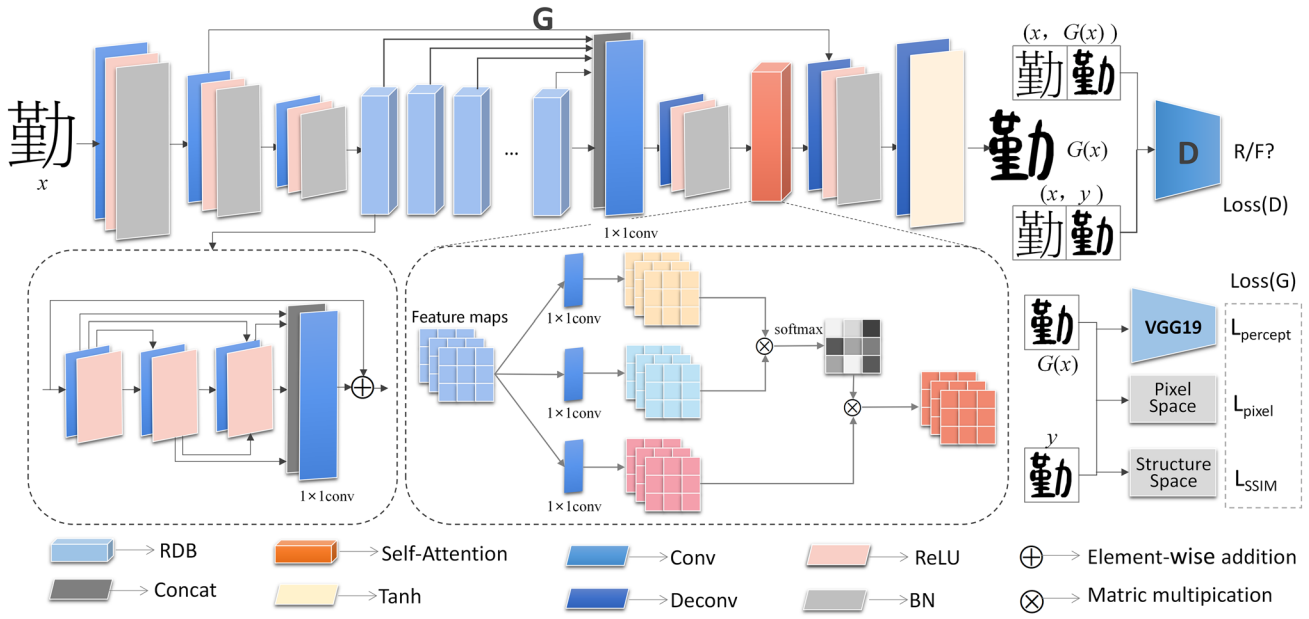

Fig. 4 Font Conversion Network Model Structure

where σ represents the ReLU activation function. $W_{n,i}$ represents the i-th convolution operation in the n-th RDB block. F_{n-1} represents the output of n-1th RDB, and $[F_{n,1}, ..., F_{n,i-1}]$ represents the output of the previous i-1 convolutions through dense connection.

$$F_{n,LF} = H^{d}_{LFF}([F_{n-1}, F_{n,1}, ..., F_{n,i}, ..., F_{n,I}]) \tag{2}$$

where $F_{n,LF}$ represents the convolution operation of 1×1 after concat, which is used to compress the dimensions of the output, and reduces the parameter growth caused by feature fusion in the residual block.

$$F_n = F_{n-1} + F_{n,LF} \tag{3}$$

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.

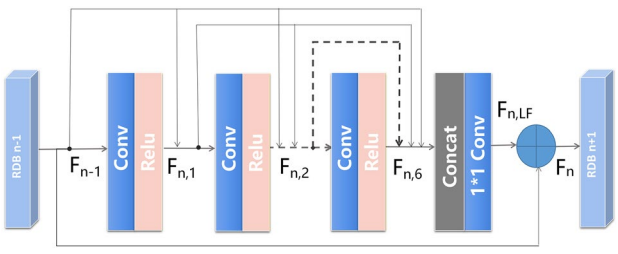

Fig. 5 Residual Dense Block

(2) Global residual learning

Global information integration of RDB can more effectively learn more effective features from previous and current local features, and stabilize the training of a wider network. Global Feature Fusion (GFF) splices the output of 8 RDB:

$$F_{GF} = H_{GFF}([F_1, ..., F_8]) \tag{4}$$

H_{GFF} is a composite function composed of 1×1 convolution and 3×3 convolution. 1×1 convolution layer is used to adaptively fuse features of different RDB layers, and then 3×3 convolutional layer is introduced to further extract features for global residual learning. While ensuring the deep structure, in order to ensure the maximum information flow between the layers in the network, F_{DF} was obtained by residual connection between the shallow feature F_0 and the global fusion feature F_{GF}. Finally, the convolutional layer restored the feature vector to the image, reducing unnecessary content structure and information loss of irrelevant image domain.

$$F_{DF} = F_{-1} + F_{GF} \tag{5}$$

(3) Feature attention mechanism

Biological scientists found that when many animals focus on visual activity, they will first observe the overall field of view to obtain a focus area that is worthy of attention, slowly focus and form an attention focus, and then the visual center will order the eyeball to devote more attention to the area to obtain richer and complete image details. The feature attention mechanism in this paper comes from the non-local neural networks proposed in the literature [23]. For the image generation task, Zhang et al. [24] proposed self-attention generative adversarial network (SAGAN), which combines non-local neural network and generative adversarial network, which can handle long-distance and multi-level image dependency. As shown in Fig. 6, the traditional convolutional neural network model generates image details at the local points of the feature map, while the self-attention mechanism generates details based on all feature points, so that the details of each position are well coordinated with the details at the far end.

Because the convolutional neural network in the generator network is limited by the size of the convolution kernel, it cannot capture the global dependence in finite network hierarchy. In order to increase the dependency information between local and global features, the feature attention model is introduced, as shown in Fig. 7. Combining global and local spatial feature information, the near and far distance correlation between pixels of each position of the image is established, and improve the coordination and quality of the generated image.

In Fig. 7, the feature attention module takes the output feature tensor $x \in R^{C\times N}$ of the previous hidden layer channel number C and the size $N =$ height $\times$ width as input, and uses two convolution networks with convolution kernel size of 1×1 and channel number of $C' = C/8$, and obtains two feature spaces f and g, as shown in formula (6):

$$\begin{aligned} f(x) &= W_{\mathrm{f}}x \\ g(x) &= W_{\mathrm{g}}x \end{aligned} \tag{6}$$

where $W_f \in R^{C'\times N}$, $W_g \in R^{C'\times N}$. The similarity S_{ij} of two feature Spaces f and g is calculated by multiplying the tensor, and softmax function is used to normalize the weight $\beta_{\mathrm{j,i}}$ of the value of position i when calculating the value of position j, so as to obtain the parameters of the feature attention layer.

$$\beta_{j,i} = \frac{\exp(S_{ij})}{\sum_{i=1}^{N}\exp(S_{ij})}, S_{ij} = f(x_i)^T g(x_j) \tag{7}$$

An attention weight matrix is formed by $\beta_{\mathrm{j,i}}$, which represents the influence of the i-th position feature on the j-th position feature. The more similar the two position features, the greater the correlation between them. The final output of the feature attention mechanism is show in formula (8):

$$o_j = \sum_{i=1}^{N}\beta_{j,i}h(x_i), h(x_i) = W_h x_i \tag{8}$$

In the formula, h is the product of the input information x and the weight matrix $W_{\mathrm{h}} \in R^{\mathrm{C\times N}}$. In order to increase the global dependence on the basis of local dependence, the attention module and the original convolutional feature map are added to the matrix. y is the output feature map, the output of the attention layer is multiplied by the scale parameters, and accumulate the original input feature map. Finally, the convolution output through the attention mechanism is:

$$y = x_{\mathrm{i}} + \gamma o_{\mathrm{i}} \tag{9}$$

The scale parameter γ increases from 0 during the training process. The network first depends more on the local area, and then gradually assigns weights to the remote area to update learning.

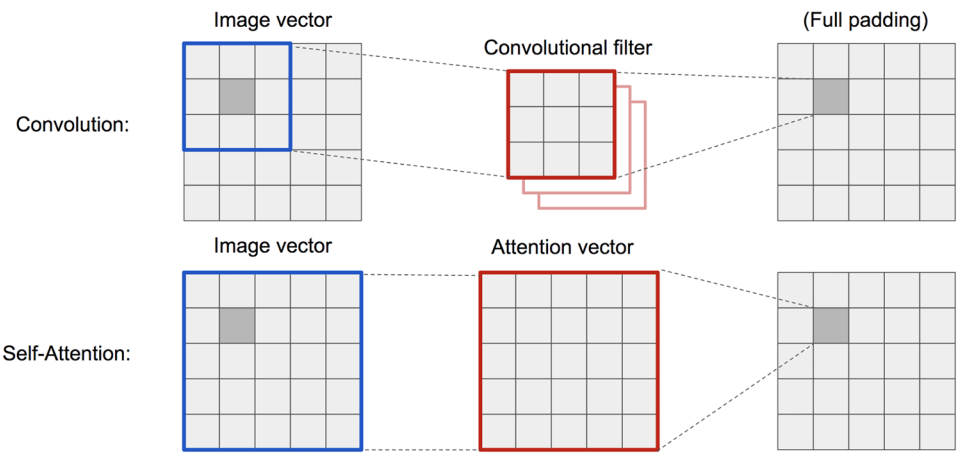

Fig. 6 Convolution Operation and Self-attention Mechanism

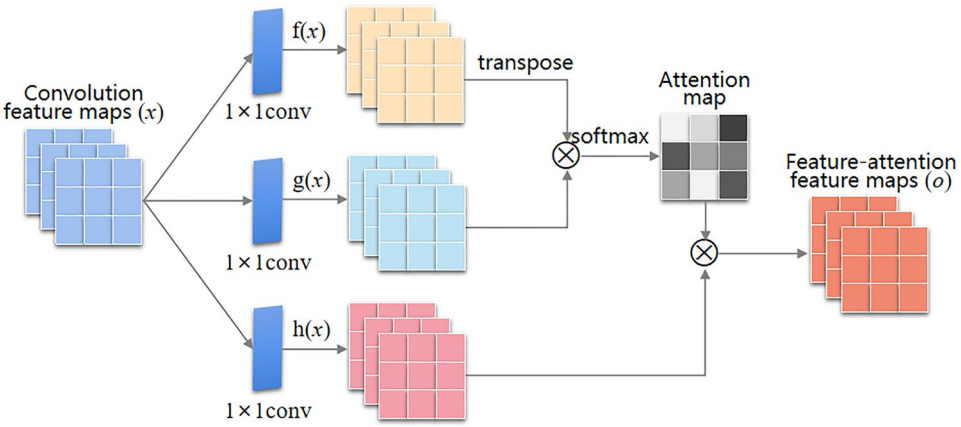

Fig.7 Feature Attention Mechanism

3.2 Discriminative network

In order to better judge the locality of the image, this paper uses patch-based network structure to authenticate the image. As shown in Fig. 8, the general two classifier outputs a True or False vector, which represents the evaluation of the entire image. The patch-based discriminator maps the input picture into N×N patches, and , respectively, judges whether each patch is true or false. Finally, the average value is taken as the final output. The advantage of patch-based network is that the input of D becomes smaller, the amount of calculation is small, and the training speed is fast, so that the model can pay more attention to the details. By identifying each Patch, local image features are extracted and characterized. This facilitates the fusion of local feature and global feature to achieve higher quality image generation.

As shown in Fig. 9, the network structure of discriminator is all composed of convolution. The input image is 256×256 font image. The first convolution module consists of the convolution layer and activation function Leaky Relu (LReLU). The second, third and fourth convolutional layers are followed by BatchNorm (BN) and LReLU, which can improve the speed of the network and avoid mode crash as much as possible. The convolution kernel size is 4×4, the image size is halved, and the number of channels is doubled. The slide step size of the first three layers is 2, and the last two layers is 1. Discarding the pooling layer can enhance the model's description of image details. The final convolutional layer maps the output to the probability score of [0, 1].

3.3 Loss function

In order to generate font images with good visual effects and structural integrity, the generator combines pixel loss, perceptual loss, and structural consistency loss with appropriate weights to form a new thinning loss function. The discriminator uses the loss function of WGAN-GP, measures the distance and difference between the two distributions by wasserstein distance, and uses gradient penalty mechanism instead of weight clipping. Through the adversarial training between the generator and the discriminator, the style transfer between different Chinese fonts is realized.

(1) Generator loss function

This article uses the L_1 distance between the generated font and the real font as the pixel-level loss, so that the generated image and the target image are as similar as possible. Because its optimization process is more stable than cross-entropy loss, and it is sharper than the constrained result of mean square error and L_2 distance. The generated image is $G(x, l)$, and the corresponding groundtruth target image is y, then the L_1 loss is expressed as formula (10):

$$L_{\text{pixel}}(G) = E_{x,y\sim P_{\text{data}}(x,y),l\sim p_l(l)}[||y - G(x)||_1] \tag{10}$$

In order to better characterize the subjective quality of the image, the perceptual loss [25] commonly used in image style transfer is used. The deep convolutional network used in this paper is VGG-19 network pretrained on the ImageNet

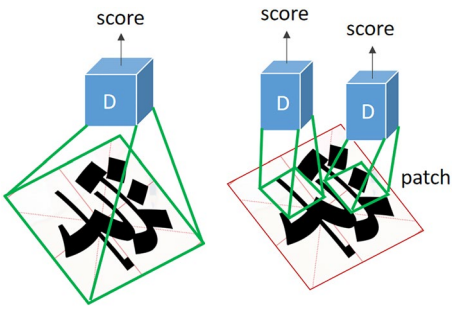

Fig. 8 Binary Discriminator and Patch-based Discriminator

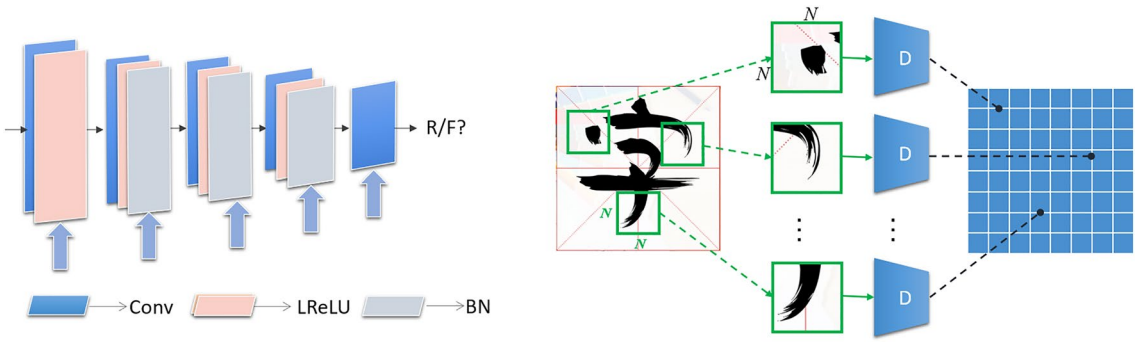

Fig. 9 Structure and Principle of Patch-based Discriminator

classification task. The generated image is $G(x)$, and the corresponding groundtruth image is y. We use three intermediate layers in the VGG-19 network to constrain the similarities of features, which are conv3_4, conv4_4, and conv5_4 layers. As shown in formula (11).

$$L_{percept}(G) = \frac{1}{C_i H_i W_i} \sum_{c=1}^{C_i} \sum_{w=1}^{W_i} \sum_{h=1}^{H_i} \left\| \varphi_i(y) - \varphi_i(G(x)) \right\|_2^2 \quad (11)$$

where φ_i is the i layer network of VGG19.

For Chinese characters under the same label, although different sample images in the training data have different font styles, the generated Chinese characters should try to maintain the correct stroke structure under the corresponding label. In the process of model training, it is necessary to limit the range of mapping functions learned by generator G and reduce the interference caused by irregular stroke data. Therefore, in order to measure the difference between the generated font image and the real font image, this paper proposes structure consistency loss. It can reduce the difference in font structure between the generated image and the real image, and limit the mapping scope of generator G. Structural consistency loss is expressed as:

$$\mathrm{SSIM}(x\prime,\ y) = \frac{(2\mu_{x'}\mu_y + c_1)(2\sigma_{x'y} + c_2)}{(\mu_{x'}^2 + \mu_y^2 + c_1)(\sigma_{x'}^2 + \sigma_y^2 + c_2)} \quad (12)$$

$$L_{\mathrm{SSIM}}(G) = E_{x,y,l\sim P_{\mathrm{data}}(x,y,l)}[1 - \mathrm{SSIM}(y, G(x))] \quad (13)$$

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

The pixel loss, perceptual loss and structural consistency loss are weighted and superimposed to obtain the loss function of the generated network, which represents the weight coefficient of the loss function.

$$L(G) = \lambda_{\mathrm{pixel}} \times L_{\mathrm{pixel}}(G) + \lambda_{\mathrm{percept}} \times L_{\mathrm{percept}}(G) + \lambda_{\mathrm{SSIM}} \times L_{\mathrm{SSIM}}(G) \quad (14)$$

(2) Discriminator loss function

In the original GAN [18], when there was no overlap between the two distributions, the Jensen-Shannon divergence could not provide a continuous and effective gradient for the generator, which caused the model to fail. The wasserstein distance reflects the minimum loss from one distribution to another. It can measure the distance between the two distributions even if there is no overlap, and provides continuous and effective gradient for the generator. Therefore, wasserstein generative adversarial network (WGAN) [26] use wasserstein distance to measure the difference between two distributions. The wasserstein distance is also called the Earth-Mover distance and is defined as follows:

$$W(p_{\mathrm{r}}, p_{\mathrm{g}}) = \inf_{\gamma\sim\Pi(p_{\mathrm{r}},p_{\mathrm{g}})} E_{(x,y)\sim\gamma}[||x - y||] \quad (15)$$

The discriminator of WGAN is defined as D, $W_{\mathrm{i}} \in [-c, c]$ and the objective function is:

$$L = E_{x\sim P_{\mathrm{r}}}[D(x)] - E_{x\sim p_{\mathrm{g}}}[D(x)] \quad (16)$$

However, WGAN's practice of cutting weights is too direct and rude, which may cause gradient disappearance or gradient explosion. In response to this problem, the WGAN-GP network [27] improved the WGAN network and proposed using a gradient penalty mechanism instead of weight clipping. The Lipschitz limit can be represented by setting additional loss $[(||\nabla_{\tilde{x}} D(\tilde{x})||_2 - K]^2$, setting K to 1 and combining WGAN's original loss. In this paper, the input data of the discriminator is integrated with the image label y, and the label information guides the sample generation of the generator. The loss function of the discriminator D is as follows:

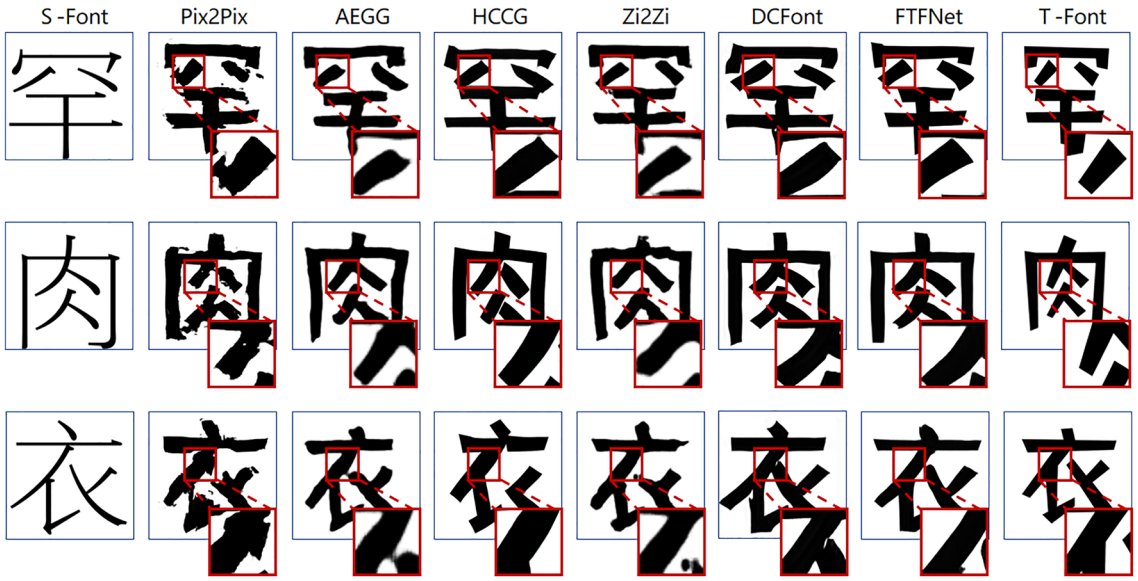

Fig. 10 The Result of Generating Songti to Jingyu

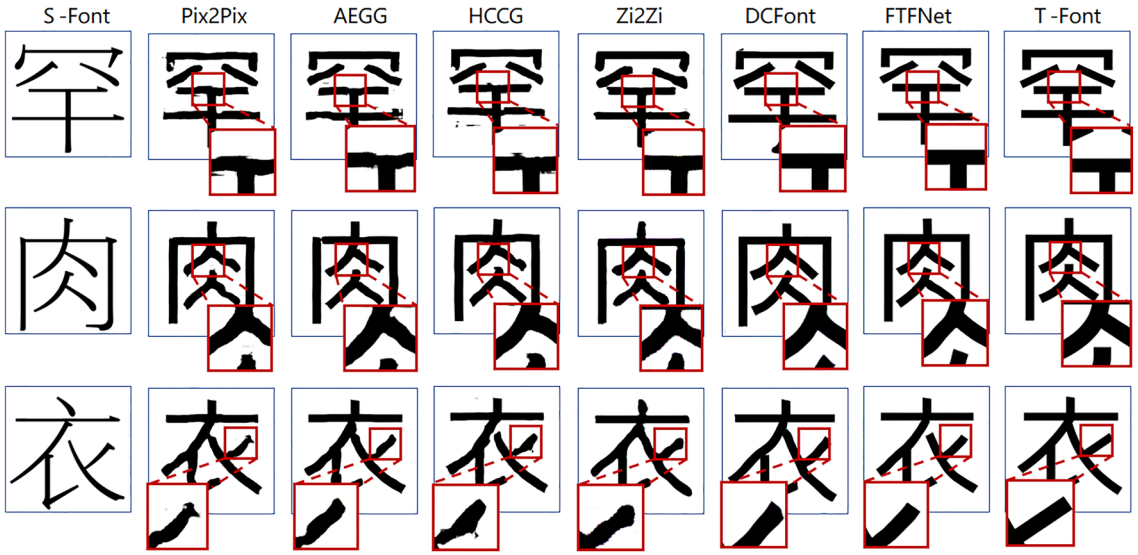

Fig. 11 The Result of Generating Songti to Heiti

$$L(D) = \underset{x\sim P_g}{E}[D(x, G(x)] - \underset{x\sim P_r}{E}[D(x, y)] + \lambda \underset{\tilde{x}\sim P_x}{E}[(||\nabla_{\tilde{x}} D(\tilde{x}, y)||_2 - 1)^2] \tag{17}$$

$$\tilde{x} = \varepsilon x_r + (1-\varepsilon)\, x_g \tag{18}$$

where $\tilde{x}$ is sampled by random difference on the line between x_r and x_g, $x_r \sim P_r$, $x_r \sim P_r$, $\varepsilon \sim$ Uniform [0, 1].

WGAN-GP further stabilizes the training process of GAN and ensures the quality of the generated results. The size of the objective function of the discriminator represents the network training process. The smaller the L(D) value is, the better the network training degree is. Through the change in the objective function, the quality of the network training and the convergence are visually displayed, which solves the problem of unstable network training.

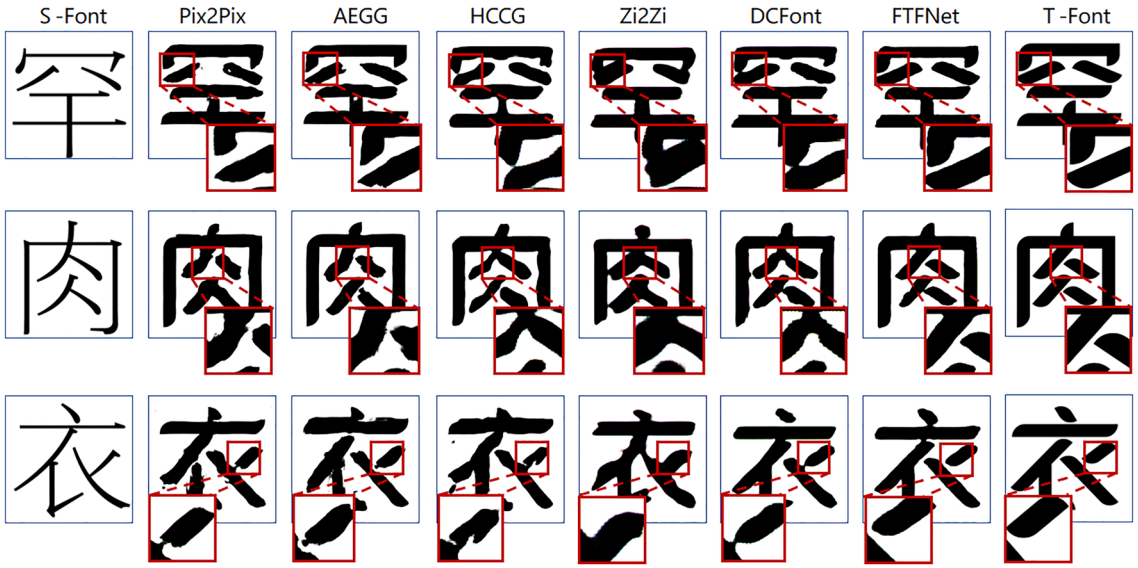

Fig. 12 The Result of Generating Songti to Benmo

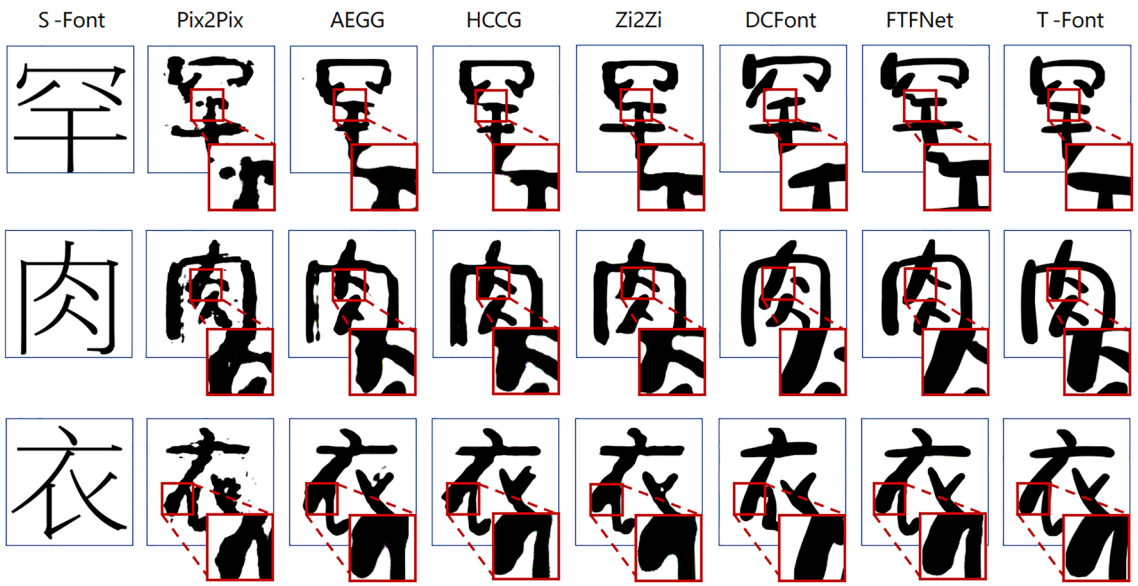

Fig. 13 The Result of Generating Songti to Mala

4 Experiments

4.1 Data sets

In order to verify the algorithm of this paper, a large amount of Chinese character sample data is needed. The font used in the study comes from the public founder font library. The TrueType fonts are decoded by python script to construct 775 font sample data. According to the ratio of 9:1, the font image is divided into training set and test set. 700 Chinese characters are used as training sets, and 75 Characters are used as test sets. The font image size is 256×256. The source domain font selected in this article is Song type, and the target domain font is six other styles, such as Heiti, Jingyu, Zengbai, Mala, Benmo and Xingkai.

This article uses four different objective indicators of mean square error (MSE), signal-to-noise ratio (PSNR), structural similarity (SSIM), and visual information fidelity

Table 1 Evaluation Index for the Different Fonts

Fonts	Method	MSE	PSNR	SSIM	VIF	UV (%)
Jingyu	Pix2Pix	0.4251	10.2536	0.5972	0.0622	10
	AEGG	0.3806	18.8657	0.7234	0.0896	30
	HCCG	0.3016	20.4524	0.7481	0.1024	63
	Zi2Zi	0.3297	17.9756	0.6821	0.0854	42
	DCFont	0.3084	24.7655	0.7513	0.1207	76
	FTFNet	0.2933	25.6397	0.7616	0.1235	79
Heiti	Pix2Pix	0.3467	20.5384	0.6824	0.0932	36
	AEGG	0.3221	21.1935	0.7296	0.0965	48
	HCCG	0.3195	23.8862	0.7331	0.1008	65
	Zi2Zi	0.3293	23.6431	0.7569	0.1044	75
	DCFont	0.2684	26.4133	0.7788	0.1507	82
	FTFNet	0.2708	26.7542	0.7832	0.1535	90
Benmo	Pix2Pix	0.3681	15.6274	0.6215	0.0975	46
	AEGG	0.3325	19.1243	0.7023	0.1067	52
	HCCG	0.3452	20.4861	0.6981	0.1041	63
	Zi2Zi	0.3466	18.5738	0.6872	0.0980	66
	DCFont	0.2784	25.9675	0.7622	0.1319	80
	FTFNet	0.2663	26.1549	0.7784	0.1566	88
Mala	Pix2Pix	0.4019	14.2891	0.6584	0.0781	18
	AEGG	0.3572	17.5854	0.6829	0.0816	26
	HCCG	0.3166	20.7941	0.7249	0.0942	52
	Zi2Zi	0.3089	20.9352	0.7312	0.0974	55
	DCFont	0.2827	24.7543	0.7588	0.1356	70
	FTFNet	0.2678	25.9572	0.7637	0.1422	75
Zeibai	Pix2Pix	0.5216	9.3257	0.4682	0.0421	8
	AEGG	0.4107	20.5119	0.6277	0.0918	22
	HCCG	0.3025	22.8436	0.6581	0.1007	46
	Zi2Zi	0.2903	23.7512	0.6866	0.1082	53
	DCFont	0.2638	24.4673	0.7486	0.1359	75
	FTFNet	0.2611	25.5934	0.7725	0.1468	78

(VIF), as well as subjective indicators of user version (UV). They are used to evaluate the quality of generated images from different levels of image low-frequency information, structural information and image perception.

4.2 Network model training

This paper optimizes the Chinese font transfer network with the idea of adversarial training. The size of the model input picture is 256×256. During training, the weight of the loss function is set to $\lambda_{\text{pixel}} = 10$, $\lambda_{\text{percept}} = 1$, $\lambda_{\text{SSIM}} = 1$. Adma algorithm ($\beta_1 = 0.5$, $\beta_2 = 0.9$) is used to optimize the gradient in the training process. The TTUR strategy proposed by Heusel et al. [28] is used to compensate for the slow learning of discriminators. The discriminator and generator use different learning rates to balance their training speed. The learning rate of discriminator and generator is 0.0002 and 0.0001, respectively. The number of iterations is 100.

4.3 Experimental analysis of different methods

The method based on image transformation is the main method of font generation and the foundation of this paper. This section compares FTFNet with five font generation methods based on image conversion. They are Pix2Pix [19], AEGG [13], HCCG [14], Zi2zi [12] and DCFont [16]. Song font is selected as the source font (S-Font) and the other five fonts as the target font (T-Font). In order to show the advantages of the method in font detail generation, this paper enlarges some stroke details by 4 times.

In Fig. 10, the Song font is converted into Jingyu font, and the three Chinese characters "han", "rou" and "yi" are tested and displayed. The fonts generated by Pix2Pix have serious false contours, missing details, and incomplete structures. This is because the Pix2Pix network mainly studies image conversion. Chinese fonts must not only achieve complete font structure, but also clearly reproduce the details of stroke outlines. Therefore, it is impossible to generate fonts with

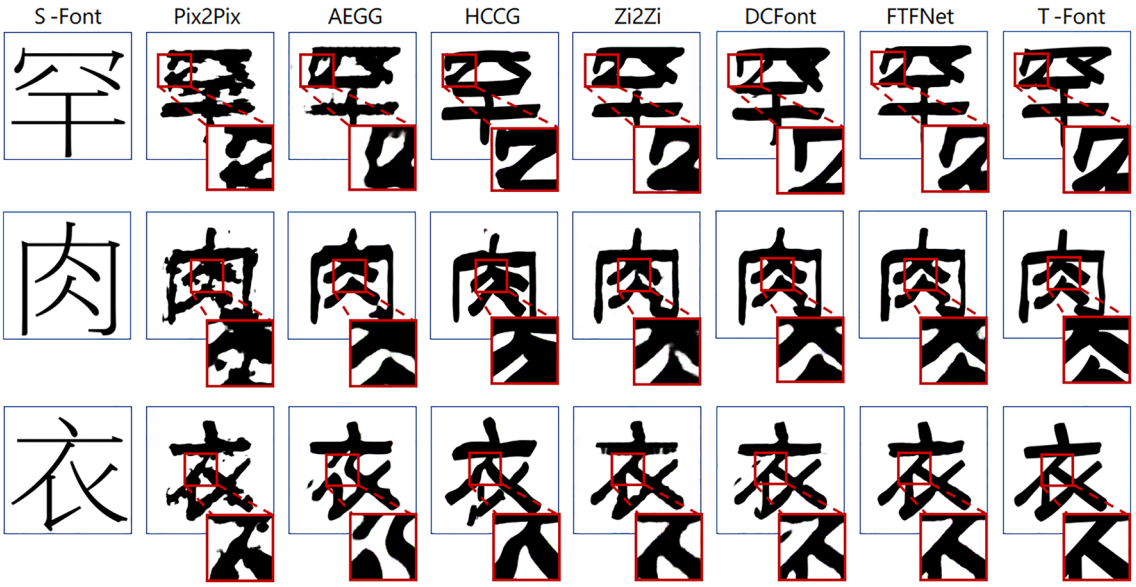

Fig. 14 The Result of Generating Songti to Zengbai

precise semantics and learn the style of fonts better. The edge of the "han" word of the AEGG method is jagged, and the outline is not clear. The outline of the Chinese characters generated by HCCG is smooth, but from the perspective of the overall structure of the font, such as "yi" there are stroke deformations. The Zi2Zi method generates complete font structure, but there are noise and pseudo contours. The DCFont method for generating fonts is more detailed and complete. Due to the complex structure of fonts and the large number of strokes, when the five comparison methods generate the target font, they cannot either completely maintain the Chinese character font or learn the detailed features of the font. Compared with other methods, from the perspective of visual evaluation, the fonts generated by FTFNet in this paper is smooth, the font shape is accurate and complete, and the topological details of fonts are maintained well.

Each font style is different and the font details are different. For fonts with similar font structure and source font, as shown in Figs. 11 and 12, which are Heiti and Benmo. The stroke thickness of the font is even, and the features of network learning are also reduced, so the generated effect is more realistic than other fonts. As shown in Figs. 13 and 14, it is the generation effect of the Mala font and the Zengbai font. Like the Jingyu font, their structure changes greatly from the original font (Song font) style, so the generated result has a small deviation in the glyph structure. However, this article has good generation effect on font details, and the overall generation effect is compared with mainstream deep learning algorithms. The FTFNet network strengthens local residual learning and global feature fusion. The font generation effect has been greatly improved, and the details are realistic.

As shown in Table 1, FTFNet generates five fonts that surpass other comparison methods in 5 indicators. The average decrease in the MSE indicator is 0.0985. The PSNR, SSIM and VIF indicators increased by an average of 1.2855, 0.3205 and 0.0365. At the same time, the quantitative index of user vision evaluation is also over 75%. They proves that the model in this paper can fully learn the local detail features and global structure features of fonts, and can fit the data distribution well. While retaining the topological structure information of Chinese characters, the detailed

Table 2 Evaluation Indicators for Different Generators

Method	MSE	PSNR	SSIM	VIF	UV (%)
CNN	0.6349	6.6251	0.2617	0.0873	3
U-Net	0.3674	15.3249	0.4849	0.0916	53
ResNet	0.3369	16.2938	0.5568	0.1098	66
DenseNet	0.3155	19.8643	0.7215	0.1126	71
No self-attention	0.2917	20.7852	0.7326	0.1095	78
FTFNet	0.2842	20.2594	0.7853	0.1269	80

Table 3 Evaluation Indicators of Generator Loss Function Ablation

Loss	MSE	PSNR	SSIM	VIF	UV (%)
No-L_{piexle}	0.4467	8.1254	0.0617	0.0851	8
No-L_{percept}	0.4359	8.5461	0.1954	0.0924	38
No-L_{SSIM}	0.3255	9.2938	0.4421	0.1015	66
FTFNet	0.3124	10.5845	0.6827	0.1061	79

Fig. 15 Font Generation Results of Different Generators

characteristics of style fonts can be generated well, the font effect generated is of high quality, the edge of the generated effect is smooth, the font is clear, and the font style migration effect is the best.

4.4 Comparative analysis of generated network

The generator introduces residual dense network. The residual dense block integrates the residual block and the dense block. Compared with ResNet and DenseNet, the residual dense network strengthens the transfer of features between each layer. CNN network uses DCGAN's [29] generator composition principle, U-Net uses encoder-decoder and skipping connection [30]. The experiments of CNN, U-Net, ResNet and DenseNet are compared with the methods in this paper. We transfer Kaiti to Xingkai. The experimental effect diagram is shown in Fig. 15. The experimental results generated by CNN are very poor, and the style characteristics of fonts can hardly be seen. The U-net network and ResNet network have the phenomenon of missing some strokes for the adherent font details. The font results generated by DenseNet and ResDenNet are similar, and the details of the continuous strokes of the font are not complete enough. From a visual point of view, the method of this article is slightly improved in the details of the font, and the font structure is complete. Combined with the evaluation indicators in Table 2, the average decrease in MSE is 0.1066, the average increase in PSNR is 2.0807, the average increase inf SSIM is 0.1297,

Table 4 Evaluation Indicators of Different Discriminant Loss Functions

Loss	MSE	PSNR	SSIM	VIF	UV (%)
$L_{\text{LSGAN}}(D)$	0.4467	8.6251	0.2915	0.0873	46
$L_{\text{WGAN}}(D)$	0.4216	8.7693	0.4538	0.0963	72
$L_{\text{WGAN-GP}}(D)$	0.3124	10.5845	0.6827	0.1061	79

Fig. 16 Comparison of Generator Loss Function Ablation

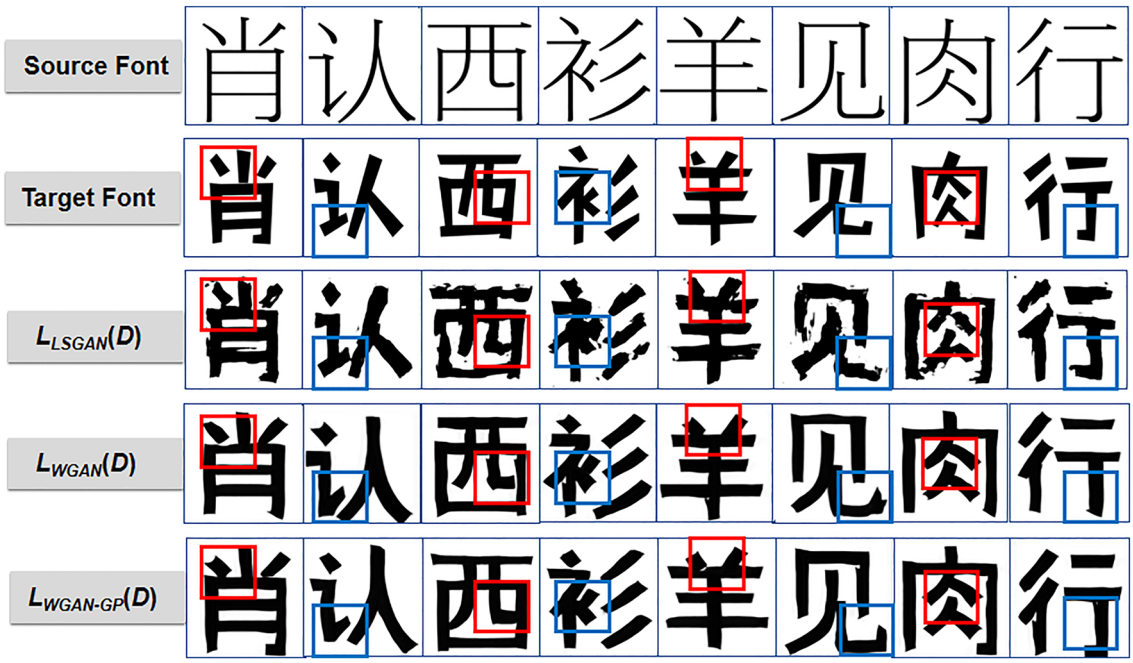

Fig. 17 Effect of Different Discriminant Loss Functions

the average increase in VIF is 0.0247, and the user evaluation index has also reached 80%. The method of this paper has been greatly improved in the details of font generation. The font and style are kept the best.

4.5 Experimental analysis of loss function ablation

In order to investigate the function of the FTFNet network loss function, this section conducts the ablation experiment of the model loss function. In the case of removing different loss functions, compare the effects of generating the same target font.

(1) Generator loss function

Investigate the image generation effects and index evaluation results after removing pixel loss, perceptual loss and structural similarity loss, respectively. The L_1 loss function, as a pixel loss, is used to measure the difference of image pixel level, which makes the network pay attention to the information of image features while taking into account the reconstruction of image pixel information. Perceptual loss is a feature-level loss. The purpose is to make the feature expression of the image before and after the migration closer in the convolutional neural network, so that certain factors of the image are retained. The SSIM loss function improves the distribution similarity between the generated image and the target image. In this paper, the Song font is used as the source font and Jingyu font is used as the target font to compare the ablation experiment effects of the loss function.

As shown in Fig. 16, after removing the pixel loss, the characteristics of the style font cannot be learned. The generated font is only slightly deformed on the source Song font, and result in more noise points. The generation effect of removing the perception loss has serious pseudo contours, and the style details of the font are distorted. The stroke details of removing the structural similarity loss are deformed and missing. The generated font in this article has some improvements compared to other methods, and is close to the style of the target font. The objective evaluation indicators are shown in Table 3. After reasonable weighting of each loss function, the constraints of the shallow and deep features of the image can be strengthened at the same time, and the overall quality of the generated image can be improved.

(2) Discriminator loss function

In order to solve the problem of disappearance of training gradient, LSGAN [31], WGAN [26], WGAN-GP [27] are proposed. The traditional GAN uses simple cross entropy loss for updating. LSGAN updates with mean square error loss, but LSGAN's excessive penalties for outliers may lead to a decrease in the diversity of sample generation. WGAN uses wasserstein distance instead of Jensen-Shannon distance to measure the distance between real samples and generated samples. WGAN-GP uses gradient penalty to satisfy lipschitz continuity conditions. In this paper, the objective functions of LSGAN, WGAN, and WGAN-GP are used as the discriminative loss function for experiments. The experimental results are shown in Fig. 17. The font image generated by LSGAN has serious false contours and distorted strokes. WGAN and WGAN-GP have good transition in contour. WGAN-GP is more stable than WGAN training, and closer to the target image in detail. Combined with the objective evaluation in Table 4, the loss function in this paper has a better generating effect than the other two loss functions.

5 Summary

This article proposes Chinese font migration method based on local and global feature learning. We treats each Chinese font as a picture and does not rely on the pre-processing in the early stage and the reorganization of the strokes in the later stage. The network model introduces residual dense network, strengthens local residual learning and global feature fusion, and enhances information transfer between network layers. At the same time, the feature attention mechanism is introduced in the down-sampling layer, which can capture the dependency relationship between local features and global features. The algorithm in this paper effectively improves the quality of Chinese font generation and realizes the style migration from different fonts. However, the network model in this paper explicitly learns the transformation from a particular source style to a given target style, so the learned model cannot be generalized to the new style. Therefore, how to learn more style features of fonts and extract more complex feature patterns without retraining is also the direction of research in the field of image style conversion.

Acknowledgements This work was supported by the two funds. They are Research on the Inheritance Technology of Ancient Inscription Calligraphy Culture Based on Artificial Intelligence, 62076200, Chinese National Natural Science Foundation, and Research on Font Generation Technology Application Based on Artificial Intelligence, 2020JM-468, Natural Science Foundation of Shaanxi Provincial Department of Education.

References

1. Yu Kai (2010) Research on some key technologies of computer calligraphy, Zhejiang University
2. Xu S, Jin T, Jiang H et al (2009) Automatic generation of personal Chinese handwriting by capturing the characteristics of personal

handwriting[C]. In: Proc of the 21st innovative applications of artificial intelligence conference. [S.I.]: IAAI-09, pp 191–196
3. Zhou B, Wang W, Chen Z (2011) Easy generation of personal Chinese handwritten fonts
4. Chang J, Gu Y (2017) Chinese typography transfer. arXiv: Computer Vision and Pattern Recognition
5. Zheng Z, Zhang F (2018) Coconditional autoencoding adversarial networks for Chinese font feature learning. arXiv: Computer Vision and Pattern Recognition
6. Atarsaikhan G, Iwana B K, Narusawa A et al (2017) Neural font style transfer. In: Iapr international conference on document analysis & recognition
7. Upchurch P, Snavely N, Bala K et al (2016) From A to Z: supervised transfer of style and content using deep neural network generators. arXiv: Computer Vision and Pattern Recognition
8. Baluja S 2017 Learning typographic style: from discrimination to synthesis. machine Vis Appl 28(5): 551–568
9. Kumarbhunia A, Kumarbhunia A, Banerjee P et al (2018) Word level font-to-font image translation using convolutional recurrent generative adversarial networks. In: international conference on pattern recognition, pp 3645–3650
10. Zhang X Y, Yin F, Zhang Y M et al (2018) Drawing and recognizing Chinese characters with recurrent neural network. IEEE TransPattern Anal Machine Intell 40(99):849–862
11. Tian. ReWrite. Retrieved from https://github.com/ kaonashi -tyc /Rewrite/. (2016)
12. Tian. ReWrite. Retrieved from https://github.com/ kaonashi-tyc/ zi2zi/. (2017)
13. Lyu P, Bai X, Yao C et al (2017) Auto-encoder guided GAN for Chinese calligraphy synthesis. In: international conference on document analysis and recognition, pp 1095–1100
14. Chang B, Zhang Q, Pan S et al (2018) Generating Handwritten Chinese Characters Using CycleGAN. Workshop on applications of computer vision, pp 199–207
15. Zhu J Y, Park T, Isola P et al (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232
16. Jiang Y, Lian Z, Tang Y et al (2017) DCFont: an end-to-end deep chinese font generation system. In: international conference on computer graphics and interactive techniques
17. Mirza M, Osindero S (2014) Conditional generative adversarial nets. arXiv:1411.1784
18. Goodfellow I, Pouget-Abadie J, Mirza M et al (2014) Generative adversarial nets. Advances in neural information processing systems, pp 2672–2680
19. Isola P, Zhu JY ,Zhou T et al (2017) Image-to-image translation with conditional adversarial networks. Computer vision and pattern recognition (CVPR), pp 5967–5976
20. He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. IEEE, pp 770–778
21. Huang G, Liu Z, Weinberger KQ, van der Maaten L (2017) Densely connected convolutional networks. CVPR
22. Zhang Y, Tian Y, Kong Y et al (2018) Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2472–2481
23. Wang X, Girshick R, Gupta A et al (2017) Non-local neural networks[C]. Computer vision and pattern recognition, pp 7794–7803
24. Zhang H, Goodfellow I, Metaxas DN et al (2018) Self-attention generative adversarial networks. arXiv: Machine Learning
25. Johnson J, Alahi A, Li FF Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision
26. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein GAN[J]. arXiv:1701.07875
27. Gulrajani I, Ahmed F, Arjovsky M et al (2017) Improved training of Wasserstein GAN. arXiv:1704. 00028
28. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: Proceedings of the 31th conference on neural information processing systems, Long Beach, California, US: MIT Press, pp 6626−6637
29. Zhu W, Miao J, Qing L et al (2015) Unsupervised representation learning with deep convolutional generative adversarial networks computer science. arXiv:1511.06434
30. Zhang Lyumin, Ji Yi, Lin Xin et al (2017) Style transfer for anime sketches with enhanced residual U-net and auxiliary classifier GAN [C]. In: Proc of the 4th Asian conference on pattern recognition. Piscataway, NJ: IEEE Press, pp 506–511
31. Mao X, Li Q, Xie H et al (2017) Least squares generative adversarial networks. In: International conference on computer vision, pp 2813–2821