Journal of Network Intelligence
Taiwan Ubiquitous Information

Volume **9**, Number **3**, August **2024**

Artistic Font Style Transfer Based on Deep Convolutional Generative Adversarial Networks

Juan Wang

School of Humanities and Social Science
Xi'an Jiaotong University, Xi'an 710049, P. R. China
School of Art and Design
Xi'an University, Xi'an 710063, P. R. China
wangjuan6000@163.com

Dong-Hun Liu

Faculty of Information Communication Technology
Limkokwing University of Creative Technology, 63000 Cyberjaya, Selangor, Malaysia
bm0898@163.com

*Corresponding author: Juan Wang
Received December 5, 2023, revised March 17, 2024, accepted May 30, 2024.

Abstract. *Art font style transfer is a very interesting but challenging task, its purpose is to transfer the art style of the font from the target font to the source font by some mapping method. In order to solve the problems of incomplete font structure, low image resolution and loss of texture details, this paper designs an art font style transfer method based on Deep Convolutional Generative Adversarial network (DCGAN). The method first optimizes DCGAN, and completes the art font denoising task better by changing the network structure Settings and corresponding parameters, and obtains the feature information closer to the original art font denoising image. Then, based on the optimized DCGAN, the convolutional neural network is replaced with a denoising encoder, and main characteristics are fused among the content characteristics and style characteristics of the art font, and a pooling stratum on the ground of wavelet multistage transmutation is added to the decoder of the generator to retain the main characteristics of the art font, so that the generated art font effect is closer to the original font style. Simulation outcome indicates that the Structural Similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR) and Frechlet Inception Distance (FID) of the proposed design method are 0.794, 19.823 and 91.267, respectively, which are superior to the comparison model and have good migration performance.*
Keywords: Art font; style transfer; deep convolution; generative adversarial network; wavelet pooling

---

1. **Introduction.** The artistic style of font is the additional style of text, for example color, font outline, strokes, lines and textures, etc. These artistic styles convey rich visual information, not only reflect the designer's philosophical thinking and emotional intention, but also affect the audience's cognitive feelings [1, 2, 3]. Nowadays, with the rapid development of multimedia technology. In a great deal of multi-visual inventive projects containing pictorial invent, for example advertising and movie placards, typefaces with particular artistic modes are progressively fashionable. For instance, in the popular film and television works in the past two years, carefully designed artistic fonts have been used. These unique style fonts have earned enough attention from the audience, and also

brought a beautiful visual experience. However, traditional art font design often requires professional design software, which has a high test and requirement on the skills and experience of designers, and it is also a very tedious and repetitive work [4, 5, 6]. In addition, because of the many complex combinations of lines, colors, and textures, art font design is often a time-consuming and labor-intensive process, especially when the character sets to be dealt with are very large.

1.1. **Related Work.** In recent years, with the strong rise of deep learning methods, all walks of life are trending toward big data and intelligence. Under this background, more and more researchers have begun to explore different methods to automatically generate art fonts [6]. How to effectively reduce the labor cost of art font design and automatically generate fonts consistent with the target font style has gradually become the focus of research. In some early methods, Zhang et al. [7] tried a series of methods based on deep learning to carry out font style conversion. Lian et al. [8] performed stroke decomposition of fonts, select fonts that cover the maximum number of font sidesteps to form a training set, and use encoders to extract the features of source fonts and target fonts as input and output of the feature transfer network, respectively. Since then, relevant studies on font style transfer have gradually attracted people's attention. For the problem of English font style transfer, Khosravi and Kabir [9] tried to parameterize font structure information, but manual intervention was needed. Later, Hu and Hersch [10] improved on the above method using the component-based idea, which explicitly describes characters as a combination of parameterizable shape components. On the other hand, Miyazaki et al. [11] explored the topology-based weighted mixing technology in the field of automatic generation of English characters. Therefore, in the study of Chinese character style transfer, Xu et al. [12] proposed a constrained likelihood-based reasoning process.

Subsequently, Gao et al. [13] explored a hierarchical representation of Chinese characters, but the font generation process was very complicated. In addition, Porziani et al. [14] also proposed a new method of automatic deformation for different styles of Chinese characters. Subsequently, the performance of Convolutional Neural Networks (CNNS) has been greatly improved, and more and more people have begun to open up new research directions in the field of font migration. Among them, Hayashi et al. [15] proposed a method for style transfer of English characters by using deep learning methods. Zeng and Pan [16] proposed the first Stacked CGAN model based on generative adversarial networks in the field of English character migration, but the training process was relatively complicated and the model generalization ability was limited. Reddy et al. [17] realized font style transfer by modifying neural network style transfer. Chung et al. [18] explored a Chinese character generation model of superimposed bidirectional recurrent neural network (RNN). Hassan et al. [19] proposed a model for automatic generation of handwritten English fonts, which could extract the texture and stroke features of font images. Cheng et al. [20] regards the design process of Chinese fonts as an image translation task and proposes the Rewrite model. Xu et al. [21] regarded the problem of font generation as a mapping problem of learning from a given standard font to an individualistic color style, and thus proposed DenseNet and CycleGAN to generate Chinese handwritten characters. Miao et al. [22] proposed a co-conditional automatic coding adversarial network for Chinese font feature learning. However, the training process of these methods is very time-consuming, and can only be transferred between pure Chinese fonts of a specific style, and can not deal with the problem of font style transfer with complex background. Zhou et al. [23] proposed a method to generate high-quality Chinese fonts by incorporating the component information of Chinese characters into the model. Wang et al. [24] put forward an end-to-end Chinese font generation model, which makes the degree of style change between the generated font and the source font softer, but there is a problem of font ambiguity.

1.2. **Contribution.** Aiming at the problems such as blurring and mislinking of strokes generated in the current art font transfer means, this article suggests a method of art font style transfer on the ground of Deep Convolutional Generative Adversarial Network (DCGAN). First, the convolutional neural network is replaced by a denoising autoencoder, and the initial picture and the denoised image are input into the discriminator together to obtain the feature information closer to the denoised image of the original art font, so as to achieve the purpose of optimizing DCGAN. Then, the encoder extracts the features of the artistic font for channel characteristic shift and characteristic refinement in terms of the style domain attribute refinement channel, conducts wavelet multistage decomposition of the feature map of the artistic font, separates the lofty and low oftenness message of the feature map, and carries out convolution and normalization operations to make the generated effect of the artistic font closer to the original font style. The experimental results show that compared with other models, the method designed in this paper has certain advantages in both image style transfer and font transformation, and has higher transfer performance.

2. Basic theoretical analysis.

2.1. **Generative adversarial network.** Generative adversarial network has two important components, the generator model and the discriminator model [25, 26]. As shown in Figure 1, the generative model generates simulated samples, and the discriminant network inputs two kinds of samples, one is the real sample and the other is the simulated sample. The discriminant network needs to learn constantly to judge the simulated sample as much as possible, so that the generative network can generate more realistic samples as much as possible. The generative adversarial network is used to supervise the self-learning process of the network.

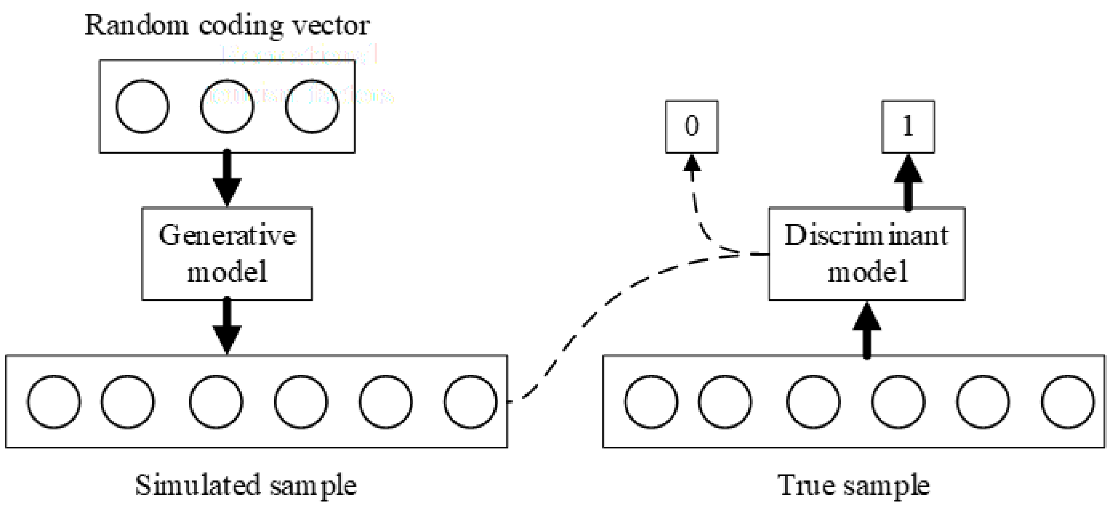

Figure 1. Generative Adversarial Network

The generation model maps from an input data to the generation data $y = F(x)$. Generally, the input x will satisfy the simple random distribution. In order to make the generated data distribution as close as possible to the real distribution, the generating function F is designed as a neural network, through which various distribution types are fitted, and the discriminant network is used to supervise and feedback the generated network.

$$K^{(C)}(y^{(C)}, y^{(F)}) = -\frac{1}{2}E_{y\sim P_d}\log C(y) - \frac{1}{2}E_{y\sim P_y}\log(1 - C(F(x))) \tag{1}$$

where the discriminant network function is $D(y)$, the generating network function is $F(x)$, the cost function is $K^{(C)}$ for the input data, and $\nu \sim P_d$ for the data achieved by the generated network.

The cost function of the generator is: $K^{(F)} = -K^{(C)}$.

From the above formula, it is necessary to find a problem that minimizes the respective cost functions of the generating network G and the discriminating network D, and define it as a problem of finding minimax values, the formula is as follows:

$$\min_F \max_C E_{y\sim P_d}[\log(C(y))] + E_{x\sim P_x}[\log(1 - C(F(x)))] \tag{2}$$

The generated simulated sample distribution is combined with the real sample distribution to obtain the joint distribution set. Then the simulated sample and the real sample are sampled, the distance between the two is calculated, and the mean of the distance is obtained. In this way, the network can be trained to optimize the way of taking a lower bound on the mean of this distance, and the discriminant network is no longer judging the truth or false, but calculating the distance between the two distributions. The objective function is as follows:

$$\min_F \max_{C\in\mathcal{C}} E_{y\sim P_s}[C(y)] - E_{y\sim P_y}[C(\overline{y})] \tag{3}$$

where P_s is the real data sample distribution, P_y is the simulated sample distribution, and C is the set satisfying Lipschitz condition.

In the generated model, the labeled data set is taken as the training input of the model, the content is taken as the output of the generated model, and the training results can be output according to the corresponding labels. This is the Conditional Generating Network (CGAN), and the objective function equation is as follows:

$$\min_F \max_C E_{y\sim P_d}[\log(C(y \mid x))] + E_{x\sim P_x}[\log(1 - C(F(x \mid y)))] \tag{4}$$

2.2. **Transfer learning strategies and algorithms.** Assuming that sample set $\{S = \{(x_j, y_j)\}_{j=1}^M\}$ is linearly separable, the transfer learning model is defined as follows:

$$g(y) = \text{sign}(h^S \cdot y) \tag{5}$$

where $x \in \mathbb{R}^m$, $h \in \mathbb{R}^m$, and sign are symbolic functions.

For any point (x_j, y_j) in the sample set, $h^S x_j > 0$, $y_j = +1$ are denoted if the point is correctly classified; Conversely, if the point is misclassified, write $h^S x_j < 0$, $y_j = -1$. So it can be defined: if $y_j h^S x_j > 0$, it means that the point is correctly classified; $y_j h^S x_j < 0$ indicates that the point has been misclassified. The loss function of the transfer learning model can be defined as:

$$L(h) = \sum_{j=1}^{M} I\{y_j h^S x_j < 0\} \tag{6}$$

Since the indicator function is noncontinuous and nondifferentiable, we can define A : $\{\text{misclassified samples}\}$ and change the loss function.

$$L(h) = \sum_{x_j \in A} -y_j h^S x_j \tag{7}$$

The resulting gradient is:

$$\nabla_h L = -y_j x_j$$

For the sample set S, set the initial weight h_0 and offset d_0, and note the learning rate as $\mu\,(0 < \mu \leq 1)$. In the sample set, a point (x_j, y_j), if $y_j(h \cdot x_j + d) \leq 0$, then

$$h^{(s+1)} \leftarrow h^{(s)} - \mu \nabla_h L$$

is substituted by the gradient:

$$h^{(s+1)} \leftarrow h^{(s)} + \mu y_j x_j \tag{8}$$

$$d \leftarrow d + \mu y_j \tag{9}$$

3. **Improved deep convolutional generation adversarial network.** In this paper, the enhanced deep convolutional generation adversarial network model optimizes the network architecture and training parameters of the generator and discriminator in DCGAN. The generator in DCGAN is not only replaced by convolutional neural network with denoising autoencoder, but also a new deep convolutional neural network is redesigned as the discriminator. By changing the network structure Settings and corresponding parameters, the art font denoising task is better completed, and the feature information is more similar to the original art font denoising image. The general flow of the algorithm is as follows: First, the noise image is input into the generator, and the de-noised image is output after processing by the generator. Then, the original image and the de-noised image are input into the discriminator, the discriminator outputs the similarity value between the two art font images, and then the similarity value is passed to the generator to guide the generator to optimize the performance, and finally the de-noised image is obtained. The overall structure of the improved DCGAN is indicated in Figure 2.

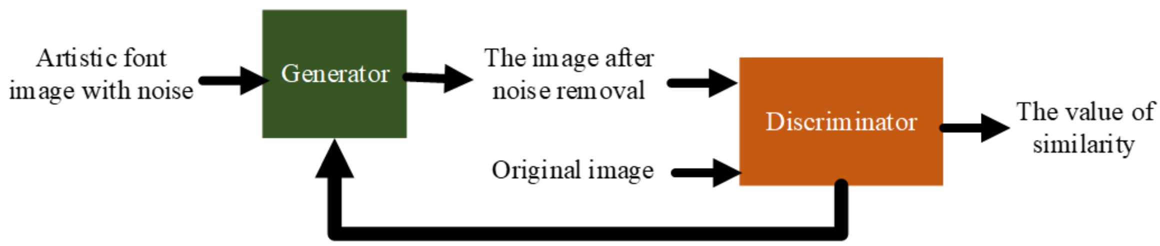

Figure 2. The structure of optimized DCGAN

The random vector y is assumed to have a symmetrical noise dispersion P_y^r, and the generator pattern $F(x)$ matches it to the message space of the actual picture. The stimulus x of the differentiator is the original image with label c, whose distribution is $P_d(x, y)$, and the output of the completely linked level of the discriminator is the $l+1$-dimensional transmitter of $k = \{k_1, k_2, \ldots, k_{l+1}\}$, that is commuted by the softmax perform into the class probability $P = \{p_1, p_2, \ldots, p_{l+1}\}$ of $l + 1$, where the actual picture will be umpired as the initial l class and the blurred picture will be umpired as the $l+1$ class. The softmax operation is shown as follows:

$$P_i = \frac{e^{k_i}}{\sum_{j=1}^{l+1} e^{k_j}}, \quad i \in \{1, 2, \ldots, l+1\} \tag{10}$$

where k_j represents the stimulus vector of the completely linked level, k_i represents the course vector of the harvest of the completely linked level, P_i represents the course potentiality of the harvest, and e is the found number of the natural logarithm.

In this paper, the traverse entropy loss operation is used as the loss operation of differentiator $C(x|y)$, mainly to distinguish whether the real image output value and the expected output value are similar. The smaller the loss value drops during training, it indicates that the model fits better. In order to better optimize the network model, the need is to minimize the value of the loss function. This paper defines $C(x|y)$ as Equation (11).

$$C(x|y) = -\sum_{i=1}^{l} x'^{i} \log(P_i) \tag{11}$$

where i represents the category and x' is the desired category. x and x' using a one-hot encoding (one-hot encoding), that is, if the discriminator output is class i, its corresponding position code is 1, the rest of the bit code is 0.

When training the deep convolutional generation adversarial network, the discriminator update speed is far behind the generator, so it is necessary to make the new speed of the two as much as possible balance during training. In this paper, the double time scale update rule is applied to network training. The generator and differentiator adopt two various studying rates and update them in $\varepsilon : \tau$ ratio.

In this paper, the gradient of discriminator model $C(x|y)$ is $w(c)$, and the gradient of generator model $F(x)$ is $w(f)$, assuming that the discriminator has n input image samples $y^{(s)}$ in each iteration training, and $1 \leq s \leq n$, the gradient $w(f)$ of the discriminator model is defined as Equation (12).

$$w(c) = \nabla_{\vartheta_t} \frac{1}{n} \sum_{j=1}^{n} [C(x|y^s, x < l+1) + C(x|F(y^s), x = l+1)] \tag{12}$$

where $x < l+1$ is the first l categories, $x = l+1$ is the fuzzy image category, y^s represents the s-th input image sample, $F(y^s)$ represents the s-th image sample generated by the generator model, and c is the value of the differentiator pattern.

Ascent $w(f)$ of the generator model is delimited as follows:

$$w(f) = \nabla_{\vartheta_t} \frac{1}{n} \sum_{j=1}^{n} C(x|F(y^s), x = l+1) \tag{13}$$

where f is the quantity of the generator pattern.

During training, it is necessary to balance the update rate of the discriminator and generator. In this chapter, the update rule with a double time scale is adopted as shown as follows:

$$w_{m+1} = \varepsilon b w_m(c) + \tau a w_m(f) \tag{14}$$

where ε and τ are the studying rates in the discriminator and generator alone, m is the number of repetitions, and $1 \leq m \leq n$. This paper iteratively updates the generator and differentiator patterns in terms of the rate of $\varepsilon : \tau$, so that the network can train more stably and extract the image features of art fonts better.

4. 4. Artistic font style transfer based on deep convolutional generative adversarial networks.

4.1. Generator network structure on the ground of feature transfer and wavelet pooling. Aiming at dealing with the issue of incomplete font structure, low image resolution and loss of texture details, this article designs a mode transmitted pattern of art font on the ground of deep convolution generation adversarial network. Firstly, based on the above optimized deep convolutional generation adversarial network, the generator in DCGAN is replaced by a convolutional neural network with a denoising encoder. Then, the encoder extracts the artistic font features to carry out channel characteristic transmit and characteristic refinement in terms of the style area dimension refinement channel, and adaptively matches the style features to the most suitable target font features. The pooling level on the ground of wavelet multistage convertation is added to the decoder of the generator, which can keep as many main characteristics of the art font as possible in the process of feature transfer. At last, aiming at enhancing the property of art font style transmit, the double-resolution discriminant network is used to discriminate the font, so that the effect of the generated art font is closer to the original font style. The overall structure is shown in Figure 3.

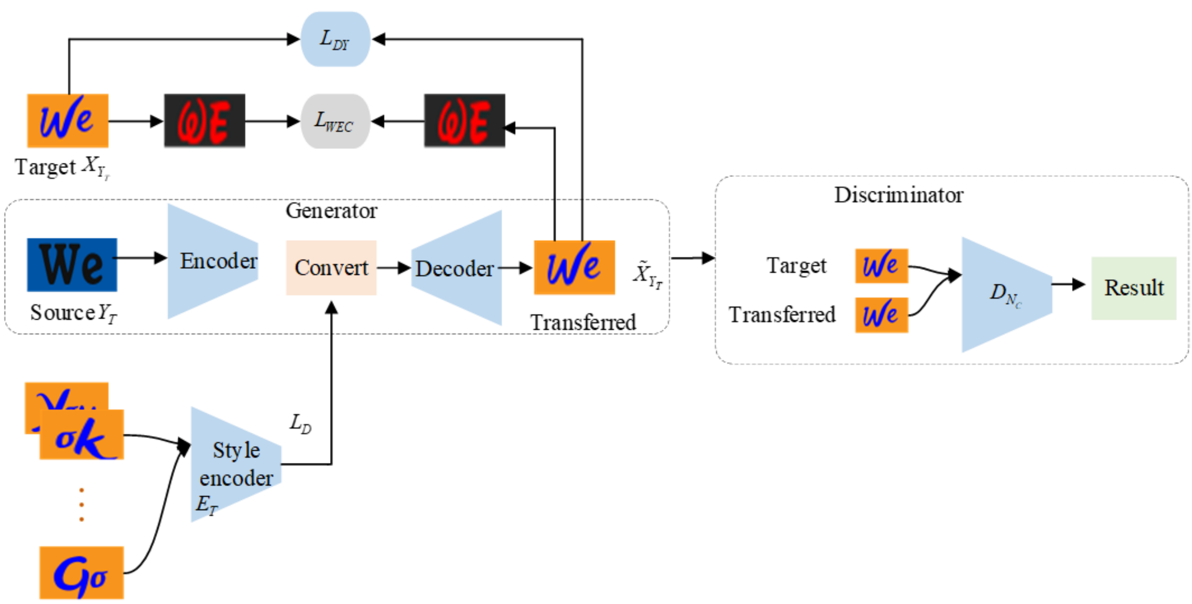

Figure 3. The overall structure

The generator network designed in the article contains the encoder, characteristic transmit pattern, and the decoder. The characteristic transmit pattern is composed of the fashion area dimension refinement cultivation to accomplish the project of characteristic alteration. The generator grips source font J_d and target font J_t as inputs and uses the content features and fashion characteristics of the source font to generate a new font J_{dt}. The generated network structure is shown in Figure 4.

First, input J_d and J_t into the encoder, which can finish characteristic excerption. Besides, various characteristics excerpted from various encoder's levels (content feature $U_{d_j}(j)$ and style feature $U_{t_j}(j)$ of the source font, content feature $U_t(j)$ and style feature $U_{td}(j)$ of the target font (j represents different layers)). Feature passing as input. The fashion area property ϵ_j of every level is achieved by heightened swirl forward multiplication to achieve full characteristic contagion and accurate unification of cross-region

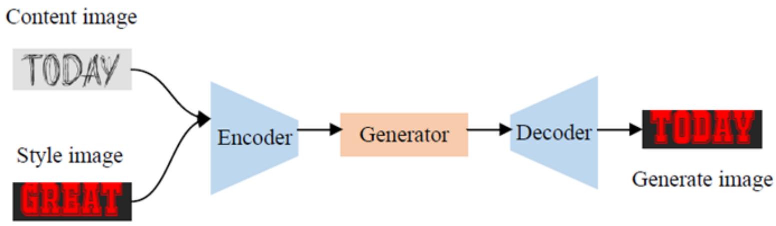

Figure 4. Network structure of generator

characteristics. The content characteristics $U_t(j)$ and fashion characteristics $U_{td}(j)$ of style images extracted from different encoder layers are fused into the dimension characteristic toter U'_{dt} through the style domain attribute thinning channel, which is passed to the decoder.

$$U_{dj_j} = f\left(\left[UD_j\left((Gra(U_j)) \odot U'_j\right)\right]\right) \tag{15}$$

$$\epsilon_j = \vartheta\left(U_{dj_j}(J)\right) \tag{16}$$

where $\odot$ represents the strait linked manipulator, UD represents the full connection level, and f represents the weight-shared full connection layer, $\epsilon(\cdot)$ represents the Sigmoid operation.

4.2. **Discriminant network structure on the ground of wavelet pooling layer.** The discriminant network uses a double resolution discriminator. The high resolution discriminator and the conventional resolution discriminator are combined to discriminate the generated art font. First, enter $b^{[k+1]}$ for linear activation to calculate $y^{[k+1]}$:

$$y^{[k+1]} = H^{[k+1]}\alpha^{[k]} + \beta^{[k+1]} \tag{17}$$

where $H^{[k+1]}$ is the weight matrix and $\beta^{[k+1]}$ is the bias factor. Then $y^{[k+1]}$ performs nonlinear activation through activation function ReLU to obtain the output $\alpha^{[k+1]}$ of the initial layer.

$$\alpha^{[k+1]} = g(y^{[k+1]}) \tag{18}$$

where g represents the activation function ReLU. Similarly, $\alpha^{[k+1]}$ first performs linear activation calculation to obtain $y^{[k+2]}$, and then obtains the final output result $\alpha^{[k+2]}$ through nonlinear activation of activation function ReLU.

Then, the feature map of the art fonts is decomposed by wavelet multistage, and the lofty and low oftenness message in the feature map is separated. The gray value shows a step change, corresponding to the lofty-oftenness sub-band message WW, KW, WK, that is spliced into the characteristic match for swirl and standardization operation, and triggered by RELU actuation operation to generate the sub-characteristic match. The specific process is as follows:

$$\{KK, KW, WK, WW\} = F_{DWT}(KK) \tag{19}$$

$$f = ReLu(DM(Conv(KW, WK, WW))) \tag{20}$$

4.3. **Loss function.** The loss operation adopted in this article contains the deprivation of GAN network, fashion area property metamorphosis network and characteristic metamorphosis pattern, which is explicated as:

$$L_{GAN} = \mathbb{E}_y[\ln C(J_y)] + \mathbb{E}_y[\ln(1 - C(V(J_d)))] + L_{CJ} \tag{21}$$

where $V(J_d)$ represents the generation of art font stylized image, $C(J_y)$ is the viability that the input font image is judged to be true, $C(V(J_d))$ is the viability that the art font stylized picture given by the differentiator is $V(J_d)$ is the real image. L_{CJ} represents the canal deprivation of style area attribute refinement, as indicated in Equation (22).

$$L_{CJ} = \mu_{cde} L_{cde} + \mu_{ck} L_{ck} \tag{22}$$

where L_{cde} represents area property deprivation, L_{ck} represents characteristic unification procedure deprivation, μ_{cde} and μ_{ck} are the weighting constituents of area property deprivation and feature fusion process loss, respectively.

For the purpose of making the fashion domain property polish the canal to learn the picture property, binary cross-randomness is adopted, and its aspect is indicated in Equation (23), in which j denotes various levels of the encoder.

$$L_{cde} = \frac{1}{|J|} \sum_{j \in J} \left(E_{F_{T_D}} \left[\log \left(F_j^{CJ}(F_{T_D}) \right) \right] + E_{F_{T_T}} \left[1 - \log \left(F_j^{CJ}(F_{T_T}) \right) \right] \right) \tag{23}$$

The channel is refined according to the attributes of μ training style domain, and key features are fused among content characteristics and fashion characteristics. The deprivation definition is indicated in Equation (24), and the range among characteristics is computed adopting K_1 distance.

$$L_{ck} = \frac{1}{|J|} \sum_{j \in J} (1-\eta) \cdot dist \left(F_j^{CJ}(F_{T_f}), F_j^{CJ}(J_{mix}) \right) + \eta \cdot dist \left(F_j^{CJ}(F_{T_f}), F_j^{CJ}(J_{mix}) \right) \tag{24}$$

where J_{mix} ($J_{mix} = mix(F_{T_f}, F_{T_d}, \eta)$) represents the fusion feature, mix is the Mixup method, and η represents the interpolation intensity.

Content loss is defined as the L_2 distance between the low-frequency subband information components of generation code $\xi(J_{D_T})$ and content code $\xi(J_d)$ extracted from the convolutional layer, as shown in Equation (25). Style loss, defined as Equation (26), is the L_2 range between the mean and discrepancy of the lower frequency subband message components of $\xi(J_{D_T})$ and $\xi(J_t)$, where $j \in \{1, 2, 3, 4\}$.

$$L_d = ||\xi_{KK}(J_d) - \xi_{KK}(J_t)||_2 \tag{25}$$

$$L_t = \sum_{k=1}^{4} \| \alpha(\xi_{KK}(J_{DT})) - \alpha(\xi_{KK}(J_T)) \|_2 + \| \vartheta(\xi_{KK}(J_{DT})) - \vartheta(\xi_{KK}(J_T)) \|_2 \tag{26}$$

5. **Experiment and analysis.**

5.1. **Model comparison experiment and performance analysis.** This paper uses the art font image data set collected in literature [27]. The dataset separated the original paired font images, resulting in 60 font pictures with various artistic styles and 80 plain text font pictures with various fonts. There are 117,180 pictures in the whole dataset, of which 50,220 are art font images and 66,960 are pure font images. The size of each image is processed as 256*256. The experimental comparison models were trained in Python v3.7 environment. For the convenience of description, reference [19] is denoted as FCGAN, reference [28] as FSTDF, reference [29] as F2PNET, reference [30] as GTPIX, and the algorithm in this paper is denoted as OURS.

The results of the comparison experiment are shown in Figure 5. Fuzzy and artifacts appear in the results of FSTDF, and the generated font strokes are missing when transferring from art font to plain text font image. The style transfer effect of FCGAN is striking, and it can generate a good artistic style font image, but the font transformation is not realized, and the generated image font is still consistent with the font of the source image. F2PNET and GTPIX can realize style transfer, but they cannot realize font transformation. Moreover, in the result of GTPIX, superfluous strokes appear in the transfer from art font to plain text font, resulting in poor legibility of font structure. FSTDF, F2PNET and GTPIX do not support conversion between invisible effects. In the results of F2PNET, there are serious artifacts in the transfer between some font image styles, and the font image font structure is distorted and deformed. The results of 5 pairs of comparison experiments show that the method in this paper has certain advantages in the transfer of image style and font transformation, especially in the transfer from plain text font image to art font image.

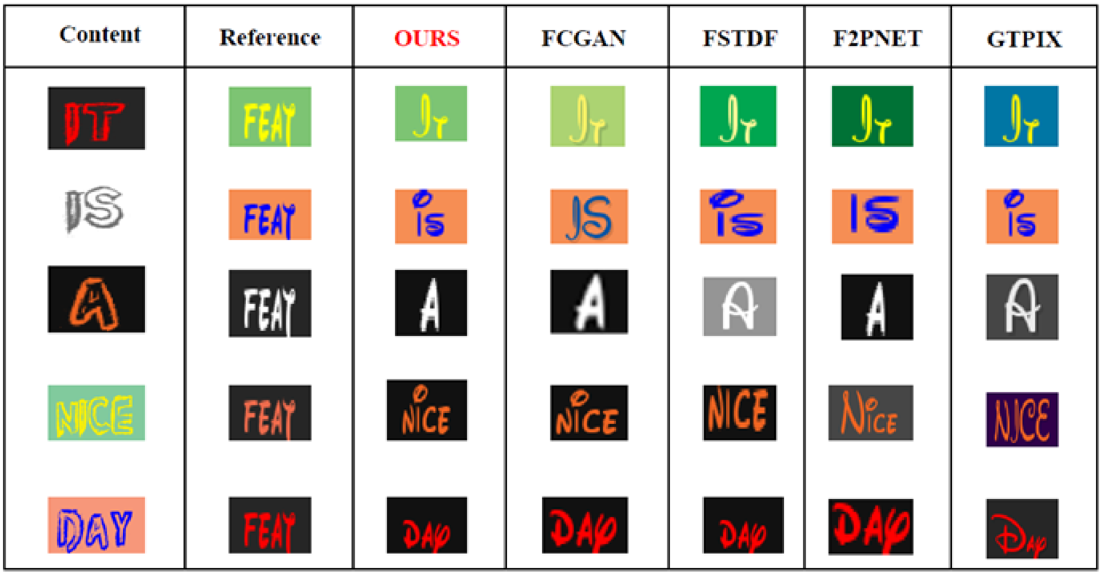

Figure 5. OURS compared with the results of the other four models

Aiming at proving the superiority of the proposed algorithm, the paper quantitatively calculates the scores of the generated results of the proposed model and the other four comparison methods on three commonly used evaluation indicators, namely Structural Similarity (SSIM), Peak Signal-to-Noise Ratio (PSNR) and Frechlet Inception Distance (FID). As can be seen from the results in Table 1, The results of the method in this article are better than other methods in the three indexes. Therefore, according to the comparison results in Figure 5 and the scores in Table 1, the model proposed in this paper achieves high clarity and less distortion and artifacts in the generated results.

Table 1. Evaluation index comparison

Model	SSIM↑	PSNR↑	FID↓
FCGAN	0.651	16.518	135.291
FSTDF	0.427	12.627	174.249
F2PNET	0.382	10.526	195.274
GTPIX	0.475	13.841	162.059
OURS	0.794	19.823	91.267

5.2. **Ablation experiment.** As shown in Figure 6, this paper studies the effects of content features, style features, attribute domain features and loss function on experimental results. In the training process, the uniform parameters of all models are set: the input value of the reference art font image is 4, and the total iteration cycle is 50 times. The results of different models are compared under the same input conditions.

In Figure 6, from top to bottom, the first line: the real target font; Second line: reference font; Compared with the second line, the content features of the fourth line are removed. Due to the instability of GAN network training, the font stroke structure in the generated plain text font image is missing. In the fifth line, the stylistic features were removed, and the color distribution was uneven and some font images were blurred. In the sixth line, the attribute field features are removed, the color is different, and the font image is deformed. In the seventh line, the generic wavelet pooling is removed, there are ambiguities and artifacts, and the font structure is seriously missing in some results.

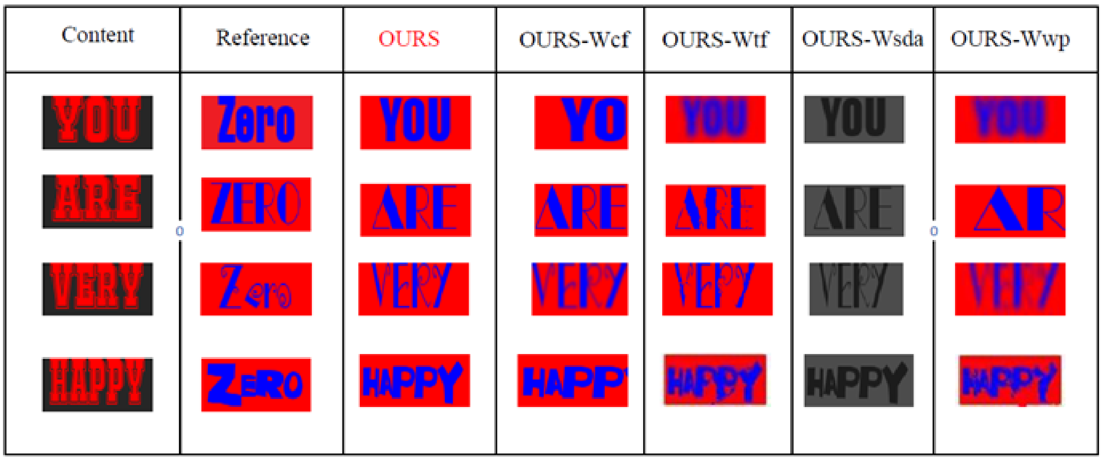

Figure 6. Comparison of ablation results

Table 2. Comparison of evaluation indexes under ablation experiment

Model	SSIM↑	PSNR↑	FID↓
Without content feature (OURS-Wcf)	0.691	17.024	129.581
Without style feature (OURS-Wtf)	0.683	16.981	140.172
Without style domain attribute (OURS-Wsda)	0.527	14.726	176.659
Without wavelet pooling (OURS-Wwp)	0.632	16.463	145.284
OURS	0.794	19.823	91.267

In addition, this paper also provides quantitative results for the evaluation of ablation experiment results (see Table 2). These indicators visually reflect the impact of each component on the entire model. It can be clearly seen from the data in Table 2 that the SSIM and PSNR of OURS-Wcf model after removing content features are 0.691 and 17.024. FID is 129.581; The SSIM of OURS-Wtf model without style features is 0.683, PSNR is 16.981 and FID is 140.172. The SSIM, PSNR and FID of OURS-Wsda model without attribute field features are 0.527, 14.726 and 176.659. The SSIM, PSNR and FID of OURS-Wwp model without wavelet pooling are 0.632, 16.463 and 145.284 respectively. The SSIM of OURS model is 0.794, the PSNR is 19.823, and the FID is 91.267. Compared with other models, OURS model shows better migration performance. This further illustrates how content features and style features can make the generated glyphs as similar as possible to the target glyphs. Secondly, the attribute domain features of art fonts also contribute greatly to the quality of local texture details of generated font images. Finally, the introduction of wavelet pooling helps to enhance the quality of the generated art font pictures.

6. **Conclusion.** There are many innovations in the research and application of the typeface style transfer method based on generative adversity-network, and its generation effect has exceeded the traditional method. However, there are still many problems to be solved in the practical application of these technologies, such as font blur, stroke mislink and so on. Therefore, the style transfer method of art font based on generative adversarial network needs to be improved in a large number of details. To solve the above problems, this paper proposes a method of art font style transfer on the ground of DCGAN. First, the generator outputs the denoised image, inputs the initial image and the de-denoised image into the discriminator, and the discriminator outputs the similarity value between the two art font images, and then passes the similarity value to the generator to guide the generator to perform performance tuning, so as to achieve the purpose of optimizing DCGAN. Then, the encoder extracts the features of the artistic font and carries out the channel feature transfer and feature refinement according to the style domain attribute refinement channel, adaptively matches the style features to the most suitable target font features, and uses the discriminant network of the wavelet pooling layer to discriminate the font, so that the generated artistic font effect is closer to the original font style. The experimental outcome indicates that the suggested method has SSIM, PSNR and FID, and shows good migration performance.

REFERENCES

[1] D. T. Vijayakumar, and M. R. Vinothkanna, "Capsule network on font style classification," Journal of Artificial Intelligence and Capsule Networks, vol. 2, no. 2, pp. 64-76, 2020.

[2] A. U. Hassan, I. Memon, and J. Choi, "Real-time high quality font generation with Conditional Font GAN," Expert Systems with Applications, vol. 213, pp. 118907, 2023.

[3] C. Li, Y. Taniguchi, M. Lu, S. i. Konomi, and H. Nagahara, "Cross-language Font Style Transfer," Applied Intelligence, pp. 1-15, 2023.

[4] Y. Ma, Y. Peng, and T.-Y. Wu, "Transfer learning model for false positive reduction in lymph node detection via sparse coding and deep learning," Journal of Intelligent & Fuzzy Systems, vol. 43, no. 2, pp. 2121-2133, 2022.

[5] S. Zhou, X. Huang, M. S. Obaidat, B. A. Alzahrani, X. Han, S. Kumari, and C.-M. Chen, "Transferability of Adversarial Attacks on Tiny Deep Learning Models for IoT Unmanned Aerial Vehicles," IEEE Internet of Things Journal, 2023. [Online]. Available: https://doi.org/10.1109/JIOT.2023.3329954

[6] S. P. Xu, K. Wang, M. R. Hassan, M. M. Hassan, and C.-M. Chen, "An Interpretive Perspective: Adversarial Trojaning Attack on Neural-Architecture-Search Enabled Edge AI Systems," IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 503-510, 2023.

[7] X.-Y. Zhang, F. Yin, Y.-M. Zhang, C.-L. Liu, and Y. Bengio, "Drawing and recognizing chinese characters with recurrent neural network," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 40, no. 4, pp. 849-862, 2017.

[8] Z. Lian, B. Zhao, X. Chen, and J. Xiao, "EasyFont: a style learning-based system to easily build your large-scale handwriting fonts," ACM Transactions on Graphics, vol. 38, no. 1, pp. 1-18, 2018.

[9] H. Khosravi, and E. Kabir, "Farsi font recognition based on Sobel–Roberts features," Pattern Recognition Letters, vol. 31, no. 1, pp. 75-82, 2010.

[10] C. Hu, and R. D. Hersch, "Parameterizable fonts based on shape components," IEEE Computer Graphics and Applications, vol. 21, no. 3, pp. 70-85, 2001.

[11] T. Miyazaki, T. Tsuchiya, Y. Sugaya, S. Omachi, M. Iwamura, S. Uchida, and K. Kise, "Automatic generation of typographic font from small font subset," IEEE Computer Graphics and Applications, vol. 40, no. 1, pp. 99-111, 2019.

[12] S. Xu, F. C. Lau, W. K. Cheung, and Y. Pan, "Automatic generation of artistic Chinese calligraphy," IEEE Intelligent Systems, vol. 20, no. 3, pp. 32-39, 2005.

[13] Y. Gao, Y. Guo, Z. Lian, Y. Tang, and J. Xiao, "Artistic glyph image synthesis via one-stage few-shot learning," ACM Transactions on Graphics, vol. 38, no. 6, pp. 1-12, 2019.

[14] S. Porziani, C. Groth, W. Waldman, and M. E. Biancolini, "Automatic shape optimisation of structural parts driven by BGM and RBF mesh morphing," International Journal of Mechanical Sciences, vol. 189, pp. 105976, 2021.

[15] H. Hayashi, K. Abe, and S. Uchida, "GlyphGAN: Style-consistent font generation based on generative adversarial networks," Knowledge-Based Systems, vol. 186, pp. 104927, 2019.

[16] S. Zeng, and Z. Pan, "An unsupervised font style transfer model based on generative adversarial networks," Multimedia Tools and Applications, vol. 81, no. 4, pp. 5305-5324, 2022.

[17] P. Reddy, Z. Zhang, Z. Wang, M. Fisher, H. Jin, and N. Mitra, "A multi-implicit neural representation for fonts," Advances in Neural Information Processing Systems, vol. 34, pp. 12637-12647, 2021.

[18] T. Chung, B. Xu, Y. Liu, C. Ouyang, S. Li, and L. Luo, "Empirical study on character level neural network classifier for chinese text," Engineering Applications of Artificial Intelligence, vol. 80, pp. 1-7, 2019.

[19] A. U. Hassan, H. Ahmed, and J. Choi, "Unpaired font family synthesis using conditional generative adversarial networks," Knowledge-Based Systems, vol. 229, pp. 107304, 2021.

[20] Z. Cheng, H. Gao, Z. Liu, H. Wu, Y. Zi, and G. Pei, "Image-based CAPTCHAs based on neural style transfer," IET Information Security, vol. 13, no. 6, pp. 519-529, 2019.

[21] L. Xu, Y. Wang, X. Li, and M. Pan, "Recognition of handwritten Chinese characters based on concept learning," IEEE Access, vol. 7, pp. 102039-102053, 2019.

[22] Y. Miao, H. Jia, and K. Tang, "Chinese font migration combining local and global features learning," Pattern Analysis and Applications, vol. 24, pp. 1533-1547, 2021.

[23] P. Zhou, Z. Zhao, K. Zhang, C. Li, and C. Wang, "An end-to-end model for chinese calligraphy generation," Multimedia Tools and Applications, vol. 80, pp. 6737-6754, 2021.

[24] R. Wang, S. Huang, Y. Zhou, and Z. G. Cai, "Chinese character handwriting: A large-scale behavioral study and a database," Behavior Research Methods, vol. 52, pp. 82-96, 2020.

[25] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," IEEE Signal Processing Magazine, vol. 35, no. 1, pp. 53-65, 2018.

[26] A. Aggarwal, M. Mittal, and G. Battineni, "Generative adversarial network: An overview of theory and applications," International Journal of Information Management Data Insights, vol. 1, no. 1, pp. 100004, 2021.

[27] S. Jiang, Z. Wang, A. Hertzmann, H. Jin, and Y. Fu, "Visual font pairing," IEEE Transactions on Multimedia, vol. 22, no. 8, pp. 2086-2097, 2019.

[28] A. Zhu, X. Lu, X. Bai, S. Uchida, B. K. Iwana, and S. Xiong, "Few-shot text style transfer via deep feature similarity," IEEE Transactions on Image Processing, vol. 29, pp. 6932-6946, 2020.

[29] G. Li, J. Zhang, and D. Chen, "F2PNet: font-to-painting translation by adversarial learning," IET Image Processing, vol. 14, no. 13, pp. 3243-3253, 2020.

[30] M. Xue, Y. Ito, and K. Nakano, "An Art Font Generation Technique using Pix2Pix-based Networks," Bulletin of Networking, Computing, Systems, and Software, vol. 12, no. 1, pp. 6-12, 2023.