Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-11777-0

An unsupervised font style transfer model based on generative adversarial networks

Sihan Zeng[1] · Zhongliang Pan[1]

Received: 27 March 2021 / Revised: 6 September 2021 / Accepted: 25 November 2021
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Chinese characters, because of their complex structure and a large number, lead to an extremely high cost of time for designers to design a complete set of characters. As a result, the dramatic growth of characters used in various fields such as culture and business has formed a strong contradiction between supply and demand with Chinese font design. Although most of the existing Chinese characters transformation models greatly alleviate the demand for character usage, the semantics of the generated characters cannot be guaranteed and the generation efficiency is low. At the same time, the models require large amounts of paired data for training, which requires a large amount of sample processing time. To address the problems of existing methods, this paper proposes an unsupervised Chinese characters generation method based on generative adversarial networks, which fuses Style-Attentional Net to a skip-connected U-Net as a GAN generator network architecture. It effectively and flexibly integrates local style patterns based on the semantic spatial distribution of content images while retaining feature information of different sizes. Our model generates fonts that maintain the source domain content features and the target domain style features at the end of training. The addition of the style specification module and the classification discriminator allows the model to generate multiple style typefaces. The generation results show that the model proposed in this paper can perform the task of Chinese character style transfer well. The model generates high-quality images of Chinese characters and generates Chinese characters with complete structures and natural strokes. In the quantitative comparison experiments and qualitative comparison experiments, our model has more superior visual effects and image performance indexes compared with the existing models. In sample size experiments, clearly structured fonts are still generated and the model demonstrates significant robustness. At the same time, the training conditions of our model are easy to meet and facilitate generalization to real applications.

Keywords Chinese characters · Style transfer · Generative adversarial networks · Unsupervised learning · Style-attentional networks

✉ Zhongliang Pan
panzhongliang@m.scnu.edu.cn

Sihan Zeng
wles1996@126.com

[1] Physics and Telecommunication Engineering, South China Normal University, Guangzhou, China

Published online: 15 December 2021

1 Introduction

As a cultural symbol of Chinese culture, Chinese characters are an important medium for people to convey information. At the same time, as an ideogram, it has complex character structure and semantics. Therefore, the design principles of the printed should first meet the accuracy and readability of the text, and then pursue artistry. However, a set of Chinese characters designed using traditional methods relies heavily on individual drawing by the designer, which costs a great deal of time and energy. Due to the development of the times, against the backdrop of the rapid increase in the total amount of information in all areas of social life, there is an explosion of demand for all aspects of text applications, while the existing development of Chinese characters design technology is relatively lagging, resulting in a strong conflict of supply and demand between the both. Therefore, with the development of computer graphics and the improvement of image processing ability, font generation technology will certainly improve the efficiency of font design or automated production, and its importance is self-evident.

Font generation techniques have been proposed for many years, but the majority of work has focused on language types containing a small number of characters, such as English [2, 10], Latin [1, 19] or Persian [23]. As for Chinese characters, even the most commonly used Chinese character set (i.e. GB2312) is made up of 6763 characters. In addition to this, Chinese characters have complex structure of radicals. Before the prevalence of deep learning, traditional Chinese characters generation techniques relied heavily on the segmentation and synthesis of character structures and the rendering of character skeletons. But with the complexity and diversity of Chinese characters, these methods do not guarantee that the character structure is extracted correctly, and their style features are easily lost. It is these shortcomings that make font generation techniques difficult to apply in practice.

Over the past few years, the application of deep learning in the field of computer vision has get remarkable success. Because of the good feature extraction performance of deep neural networks, the use of deep neural networks for generative tasks has become a popular solution. Inspired by the successful application of deep learning to generative tasks, researchers have proposed generative adversarial networks to be used for the task of Chinese characters generation. For example, based on the pix2pix framework, Tian designed an encoder-decoder and embedded Gaussian noise as a one-to-many modeling category to form a zi2zi [46] model. This model has good transfer performance for some specific typefaces. However, if the font structure is too complex, the visual effect of the generated image is not as good as it could be. Thereafter, to carry out the generative task well, [7, 17, 35] improved the quality of the font by adding Prior-knowledge of Chinese characters to the generative model. These models rely heavily on radical and stroke extraction. In practice, however, perfect automatic stroke extraction is almost impossible. in addition, they pay attention to the generation of radicals/strokes, while ignoring the inner connection between them. In addition, the methods of [5, 38, 49] were proposed. These methods carry out multi-task training for which the generation efficiency has been further improved. However, these methods all use supervised learning that requires a large number of paired training samples, which is a significant limitation in practical application scenarios.

In order to solve the problem mentioned above, We propose a novel unsupervised learning method for end-to-end Chinese characters style transfer. The unsupervised learning-based approach greatly reduces the number of datasets and pre-processing requirements of the model. Our model uses a novel generative network to ensure content consistency and style consistency in the generated fonts, furthermore the network is flexible enough to combine content and style so that strokes and styles can be matched naturally without the need for prior knowledge to improve the model generation results. In addition, we have introduced a classification discriminator and a style specification module to the model, which allows for style learning of multiple fonts to generate different styles of target font images. The main contributions of our work are as follows:

- We propose a novel model for the Chinese characters style transfer task that fuses Style-Attentional Net to U-Net [29] as a GAN generator network architecture. This coding network aims to extract the content features of the source font and the style features of the target font separately, and flexibly match the style features and content features of multiple sizes with the Style-Attentional Net. Finally, the decoder performs the reconstruction of the image to generate the stylized target Chinese characters.
- In our model, we replace the discriminator with a relative discriminator to improve the stability of the model during training and to generate clear and complete images of Chinese characters. The loss function of the model is also updated to impose reasonable constraints on the generative network and to control the generated results more precisely. The model is insensitive to the weights of each loss function.
- To achieve multi-font style transfer and improve generation efficiency, we combine the labels of the target fonts with the source fonts and add a classification discriminator to the model.
- In this paper, several experiments are conducted on a database containing over 30 styles of Chinese characters. Furthermore, the generated images are evaluated and compared using various image evaluation metrics. The results show that our model can generate high-quality images.

Following the introduction, the rest of this paper is structured as follows. Section 2 describes the related work. The proposed model and the design of the loss functions are explained in Section 3. The experimental results and comparative data are given in Section 4 and the conclusion is presented in Section 5.

2 Related work

Chinese characters generation is a topic of long-term research. Researchers have been working on font generation as early as the 1980s. Many font generation methods have been proposed so far. The existing models can be divided into two types: computer graphics-based approaches and deep learning-based approaches.

With the enhancement of computer graphics and image processing capabilities, computer graphics-based models for Chinese characters generation began to emerge. Researchers have proposed schemes to simulate the brush model [3, 20, 36, 37], For example, Lee [20] proposed to simulate the change of the brush using process by calculating the elasticity; Baxter and Govindaraju [3] proposed a data-driven 3D virtual brush; The brush model-based approach was introduced by WU [37], which creates calligraphy manually by simulating a realistic writing environment. However, these methods of simulating brushes are inefficient to generate. Moreover, they only simulate the basic style of calligraphic fonts, but cannot capture advanced styles features. The method of character splitting and synthesis [21, 33, 39, 51] is to cut and reorganize the existing Chinese characters, and finally generate the target font. For instance, Xu proposed a semi-automatic segmentation method in [39], where characters are segmented into basic structures and stored in the database, and then the target fonts are obtained by combining the learned structural layout rules with probability maximization as the training objective. In [51], Zhou applied predefined rules to select representative samples of the target font whose substructures have the ability to compose other characters. Moreover, the rules for the substructure composition are suitable for generalization to combinations of substructures in other fonts, so new characters for the target font can be obtained. The method of stroke extraction is proposed in [21], which extracts the same number of key points in the strokes of the reference and target fonts, minimizing the loss of the corresponding key points of both. These methods of character segmentation synthesis are more complex and time-consuming. Furthermore, this method is difficult to segment complex characters. Zhang used specific stroke texture patches to render skeletons in [47]. In [32], the authors proposed a skeleton rendering method that combines Chinese character's skeletons with calligraphic strokes to generate calligraphic characters. However, the character generation is inefficient and the imitation of the style is not natural.

Another solution that has attracted many researchers is the use of deep learning techniques, which have become very popular in the last few years. Methods using deep neural networks not only achieve state-of-the-art performance in many classical computer vision tasks, including Image coloring [15, 40], edge detection [8, 34], semantic segmentation [26, 53], etc. but also become increasingly competitive in solving generative problems. Therefore, the image-to-image translation is gradually applied to the task of Chinese characters style transfer.

The main network architectures that deal with the task of Chinese characters style transfer are CNNs, RNNs, and Generative Adversarial Networks (GAN). In [31], the author proposed “Rewrite” based on traditional CNN networks. The model adopts the traditional top-down CNN structure and takes different convolution sizes on different layers, allowing it to capture different details. However, the feature information cannot be fully extracted for a certain font style training. Meanwhile, noise and blurring often appear in the generated images. The experimental results are therefore not satisfactory. The experimental results are therefore not satisfactory. Lian proposed an RNN-based style transfer model for automatic extraction of strokes in [22], which extracts stroke styles and reassembles them to finally generate new characters. The model can accurately analyze the structural position of each Chinese character while reducing the need for experimental data and greatly improving the conversion of complex characters. However, the data preparation process of this method is complex, which restricts its promotion and application.

In 2014, Generative Adversarial Networks (GAN) was proposed by Ian J. Goodfellow [9]. The classical GAN consists of a generator and a discriminator, where the generator network generates images by given input noise and the discriminator network recognizes real and generated images. GAN gaining more distinguished effects in many computer vision tasks. With the development of GAN, various versions of GAN have arisen for different tasks, such as CGAN (conditional GAN) [24] adds constraints to the original GAN to make the network generate samples in a given direction. DCGAN [28] combines convolutional neural network with GAN, which ensures the quality and diversity of the generated images. Pix2pix [14] implements image transformation in two different domains, but it requires pairs of images with the same content in both domains as training data. CycleGAN [52] allows images in two domains to be transformed into each other and does not require pairs of images as training data. After learning the real image features in different style domains, StarGAN [6] generates images in different style domains with reference to the input images.

The excellent performance of GAN in image translation applications has prompted researchers to choose it as a method for font style transfer solutions. Based on the network architecture of Pix2pix, Zi2zi has established the conditional GAN-based font style transfer model by adding category embedding to the generator and discriminator. However, It does not perform significantly when trying to convert complex structured glyphs. Jiang [16] and Chang [4] use more elaborate encoders to capture character features and then generate target characters by decoders. These models solve the generation problem of complex fonts to a certain extent, but ghosting artifacts and blur often appear in the results. RNN-based methods such as [48], although solving common problems such as font structure ambiguity, cannot meet the requirements of practical applications in terms of dataset processing and training efficiency. EMD [49] and SAVAE [30] refine the functions of the generators, which model generators able to extract content and style separately. These methods generate fonts more completely and clearly, but ignore the intrinsic connection between content and style, which leads to unnatural generated fonts. [13, 17] utilizing additional prior knowledge to guide the Chinese characters generation model to produce clear and complete fonts. However, these methods rely on prior knowledge and can only apply to specific writing systems. The method of [5, 38] improves the model for the problem of generation efficiency. These methods can train multiple style samples simultaneously, which greatly improves the generation efficiency. However, the above methods are all supervised learning and require a large number of paired training samples.

To address the above-mentioned problems, we design an unsupervised learning Chinese characters style transfer model. Considering the internal relationship between content and style, the model uses SANet [27] to combine the extracted target image style features with the original image content features. Finally, the target font with clear structure and natural strokes is generated. It reduces the learning difficulty and sample demand of the generated model. Moreover, We employed Relative Discriminator, which aims to ensure the stability of the model operation and generate high-quality images. Inspired by StarGAN, we combine the source font content features and multiple target-style information, use the classification discriminator to learn multiple font feature mappings to generate multiple style fonts at once.

3 Method description

As mentioned above, our goal is to achieve font style transfer by unsupervised learning. To this end, we design new network models and loss functions. Our model consists of a generator and two discriminators. In this section, we focus on describing the model architecture as well as the loss function of the model.

The font style transfer task can be regarded as the process of mapping a given style to a target style. Our task is as follows: given source style font image $x \in X$ and multiple target-style fonts $(y_1, y_2, y_3 \ldots y_t) \in Y$. The generator outputs a fake image F_y of the target style, where the fake image contains all the target styles, which can be expressed as $F_y \rightarrow \{F_{y1}, F_{y2}, F_{y3} \ldots F_{yt}\}$. We define this process as

$$\mathcal{G}(X, Y) \rightarrow F_y \tag{1}$$

3.1 Base model

Like the traditional GAN model, our model consists of a generative module $\mathcal{G}$and a discriminative module $\mathcal{D}$. The generation module $\mathcal{G}$ is trained to generate the target image to complete the task of image translation. The main task of the discriminator module $\mathscr{D}$ is used to distinguish the generated image and the real image. In this way, $\mathcal{G}$and $\mathcal{D}$form a dynamic "game process".

As traditional GAN models are all single encoder-decoder architectures, they cannot accurately extract the required high-level features during the font generation, resulting in unsatisfactory results in the generated fonts. To solve this problem, we split the encoders into an encoder E_c with content extraction function and an encoder E_s with style extraction function. It is not only clarifying the function of each encoder but also enables the design of explicit loss functions to constrain the generated images.

In order to reduce the designer's workload, the model needs to reduce the size of the dataset as much as possible. However, Chinese characters have a complex font structure and semantics. If the dataset size is too small, the generated results will not be convincing. Therefore, our generator uses a U-Net network, which combines high-level features and low-level features to both reduce the requirement for dataset size and enhance the learning of feature information. Both ends of the U-Net consist of two encoders and one decoder, respectively. The structure of the generators is shown in Fig. 1.

3.1.1 Encoder network

As can be seen from Fig. 1, there are two inputs to the encoder side of the U-Net [26], the source font x and the target font y. The source font is input to the content encoder to extract high-dimensional content features. The target font is input to the style encoder to be transformed into high-level features of the target style. Both encoders consist of a series of Convolution-InstanceNorm-LeakyReLu blocks. The encoder was downsampled four times. After the last downsampling, we used the DenseNet [12] module to extract high-level features from the $8 \times 8 \times 128$ feature map, which preserves as much spatial information as possible, while making the training of the generative network more effective and computationally efficient. The reasonableness of doing so has been verified by [4]. By feeding the two data x and y into the content encoder E_C and the style encoder E_s respectively, their respective feature maps can be obtained:

$$F_c = E_c(x) \tag{2}$$

$$F_s = E_s(y) \tag{3}$$

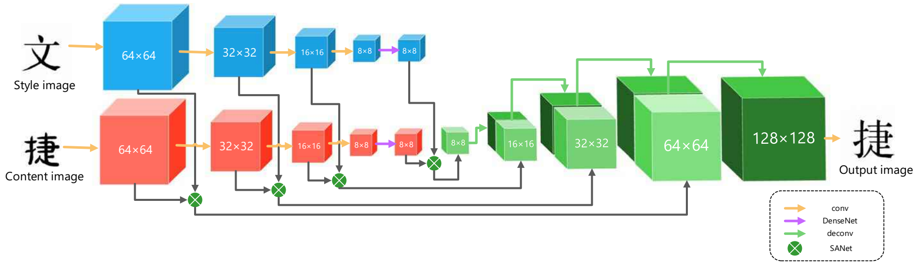

Fig. 1 Overview of our generative network. The style/content encoder extracts style/content feature maps of different sizes from the style/content images. Each SANet has the same architecture, and the content feature maps of different sizes are fused with the style feature maps. Finally the result is combined with advanced features to generate characters with the target style

$F_c(F_s)$ contains the feature maps generated by 4 downsamplings and a DenseNet module, which can be expressed by the order of generation as

$$F_c = \left\{F_c^0, F_c^1, F_c^2, F_c^3\right\}\left(F_{\mathrm{s}} = \left\{F_{\mathrm{s}}^0, F_{\mathrm{s}}^1, F_{\mathrm{s}}^2, F_{\mathrm{s}}^3\right\}\right) \tag{4}$$

3.1.2 Decoder network

The decoder, located at the right end of the U-Net, is composed of a series of upsampling layers and a Convolution-InstanceNorm-Tanh block. The Tanh function transforms the resulting output into [−1, 1]. The function of the decoder is to concatenate and reconstruct features of different sizes from the encoder to produce a target image with $128 \times 128 \times 1$. However, since the encoding layer extracts content and style features separately, we need the content features to merge with the style features and match the semantically closest style features to the content features. Here we use the Style-Attentional networks (SANet) module in Decoder D to learn the mapping between content features and style features. The structure of SANet is shown in the Fig. 2. This module is related to the self-attentive Generative Adversarial Network mentioned in [50]. The module makes use of soft attention mechanisms for stylistic decoration, thus balancing global and local stylistic styles and preserving the content structure of the original image. We input content features of different sizes and style features into the SANet module to produce feature maps:

$$F_{cs}^i = SANet\left(F_c^i, F_s^i\right) i \in (0,\ \ 3) \tag{5}$$

Correspondingly, the decoder performs 4 upsamples. The feature maps generated by each upsampling have a jump connection added to the corresponding F_{cs}^i. Low-level features are combined with high-level features to be able to make full use of the feature information. Ultimately, a 1×1 convolutional layer is used to obtain the generated image $F_y = D(F_{cs})$, where $F_{cs} = \left\{F_{cs}^0, F_{cs}^1, F_{cs}^2, F_{cs}^3\right\}$.

3.1.3 Discriminator

After completing the image reconstruction, a discriminator needs to be introduced to distinguish between the real samples and the pseudo-samples generated by the generator. Finally, the whole system parameters are updated employing gradient descent. In this paper, we use PatchGAN as the discriminator, which consists of a series of Convolution-InstanceNorm-LeakyReLu layers and ResNet modules. PatchGAN works by focusing attention on the

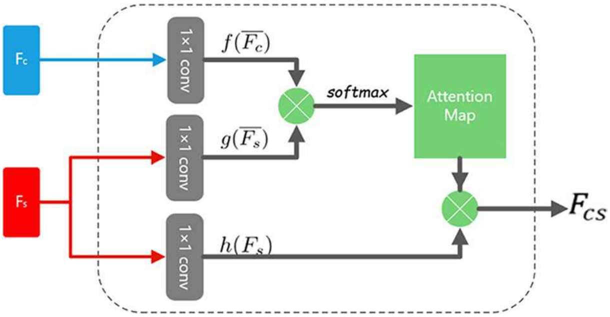

Fig. 2 A detailed illustration of the SANet module

local image rather than the global and is more accurate for image judgments. So far, the basic structure of our model has been established so that we can complete the task of one-to-one font style conversion.

3.2 Improved model

However, this model only performs a one-to-one style transfer task. If there are multiple styles of Chinese characters to be transferred, the model has to be trained many times. This fact leads to its inefficiency. Hence the model needs to be improved. Inspired by the Star-GAN model, we add a style specification mechanism to our model.

First, we need to attach the corresponding style label to the target data with the expression $(y, s) \in Y$, where y represents the real image in the Y domain and s is the style label of the font. In the generation module $\mathcal{G}$, we input both the source image x and the style label s of the target image into the content encoder, combining the input image and its style information into a new matrix: $F_c = E_c(x, s)$. And then it is combined with the style feature map F_s to generate the target image F_y. Ultimately, we introduce a classification discriminator $\mathcal{D}_{cls}$ in the discriminant module $\mathcal{D}$. This classifier constrains the generator by the classification loss function to better learn multiple styles of transfer. The classifier can replace multiple discriminators [38] for multiple styles of fonts, reducing the complexity of the network. The generators are constrained by the classification loss function to better learn multiple style transitions. Figure 3 shows our improved model with the basic workflow.

3.3 Objective function

our goal is to minimize the losses. For the design of the loss function, the paper is divided into the following main parts.

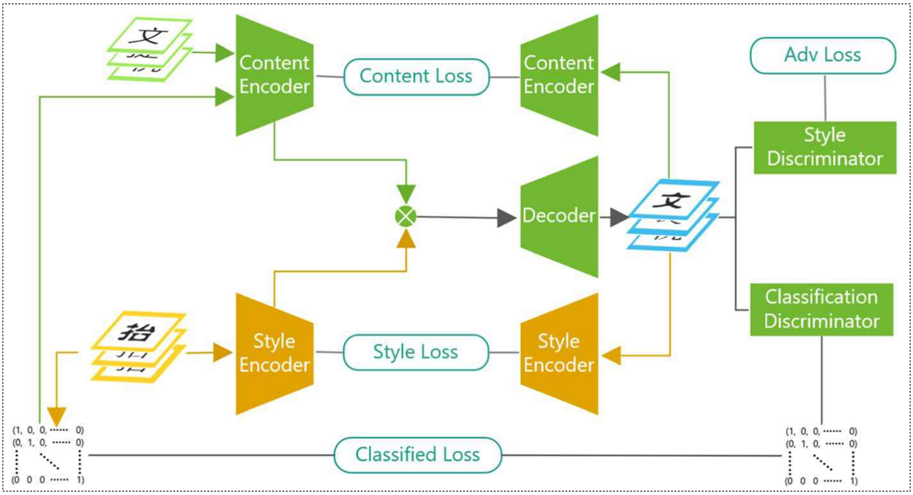

Fig. 3 The font in the green box is generated by our model. The real font is in the orange box

3.3.1 Adversarial loss

In traditional Generative Adversarial Networks, the role of the discriminator is to estimate the truthfulness of the input data, and the purpose of the generator is to increase the true probability of the fake image. Certainly, assuming that both discriminator $\mathcal{D}$and generator $\mathcal{G}$are trained to their optimal state during alternate training. At the end of discriminator $\mathcal{D}$ training, $\mathcal{D}\left(x_r\right) = 1$ and $\mathcal{D}\left(x_f\right) = 0$, and the results of generator $\mathcal{G}$ are $\mathcal{D}\left(x_r\right) = 1$ and $\mathcal{D}\left(x_f\right) = 1$, where x_r is the real data and x_f is the generated data. This causes an excessive focus on fake images and stops the learning of real images. The use of relative discriminator [32] solves this problem well and most importantly the GAN becomes more stable. We introduce a relative discriminator in this model while using the least-squares loss to replace the cross-entropy loss allowing for more stable convergence and fast training of the model, improving the quality of the generated images and model stability. The equation is shown below:

$$\begin{aligned}\mathcal{L}_{adv}^{\mathcal{G}} := & \underset{\substack{x\sim\mathbf{X}\\(y,s)\sim\mathbf{Y}}}{\mathbb{E}}\left(\|\mathcal{D}(\mathcal{G}(x|s)) - \mathcal{D}(y) - 1\|_2\right)\\ & + \underset{\substack{x\sim\mathbf{X}\\(y,s)\sim\mathbf{Y}}}{\mathbb{E}}\left(\|\mathcal{D}(y) - \mathcal{D}(\mathcal{G}(x|s)) + 1\|_2\right)\end{aligned} \tag{6}$$

$$\begin{aligned}\mathcal{L}_{adv}^{\mathcal{D}} := & \underset{\substack{x\sim\mathbf{X}\\(y,s)\sim\mathbf{Y}}}{\mathbb{E}}\left(\|\mathcal{D}(y) - \mathcal{D}(\mathcal{G}(x|s)) - 1\|_2\right)\\ & + \underset{\substack{x\sim\mathbf{X}\\(y,s)\sim\mathbf{Y}}}{\mathbb{E}}\left(\|\mathcal{D}(\mathcal{G}(x|s)) - \mathcal{D}(y) + 1\|_2\right)\end{aligned} \tag{7}$$

3.3.2 Content loss

In the style transfer task, although the original image differs significantly from the target image, the content of the generated fake image should be similar to the input image, especially for the content of the Chinese characters print, the source font and the generated font must maintain the same text topology, i.e. Character structure. However, simply using a pixel-level loss function to reduce the loss of content features is rough, as the input fonts are not simply similar to the generated fonts at the pixel level. The generated fonts have stylistic details embedded in them already. If the pixel-level loss is used, the stylistic details may be completely ignored, which hinders the stylization task. In contrast, the use of a content encoder allows stylistic features to be stripped out and only content features to be compared. The formula is shown below:

$$\mathcal{L}_{content} := \underset{\substack{x\sim\mathbf{X}\\(y,s)\sim\mathbf{Y}}}{\mathbb{E}}\left(\left\|E_c\left(D\left(E_c(x), E_s(y)\right)\right) - E_c(x)\right\|_2\right) \tag{8}$$

3.3.3 Style loss

It is well-known that different characters from the same stylistic font have the same stylistic features, which gives $E_s(y_1) = E_s(y_2)$. The optimization of the style encoder was an important part of this task. However, only the adversarial loss constrains the style encoder, thus a style loss function similar to the content loss function has to be designed in order to optimize the style encoder and force the style encoder to retain the target font style.

$$\mathcal{L}_{style} := \mathop{\mathbb{E}}_{\substack{x \sim \mathbf{X} \\ (y,s) \sim \mathbf{Y}}} \left(\| E_s(y) - E_s\left(D\left(E_c(x), E_s(y)\right)\right) \|_2 \right) \tag{9}$$

3.3.4 Classification loss

After adding the auxiliary classifier, we use the classification loss function to optimize the discriminator $\mathcal{D}$ and generator $\mathcal{G}$. We need to divide the classification loss function into two parts, with the classification loss of the real image used to constrain $\mathcal{D}$ and the classification loss function of the fake image used to constrain $\mathcal{G}$. The discriminant module $\mathcal{D}$ is shown as follows:

$$\mathcal{L}_{cls}^{\mathcal{D}} = \mathop{\mathbb{E}}_{\substack{x \sim \mathbf{X} \\ (y,s) \sim \mathbf{Y}}} \left[-\log \mathcal{D}_{cls}(y|s) \right] \tag{10}$$

By minimizing the above equation, the classification discriminator $\mathcal{D}_{cls}$ completes the correct classification of the real image. On the other hand, the loss function for fake image classification is defined as:

$$\mathcal{L}_{cls}^{\mathcal{G}} = \mathop{\mathbb{E}}_{\substack{x \sim \mathbf{X} \\ (y,s) \sim \mathbf{Y}}} \left[-\log \mathcal{D}_{cls}\left(F_y|s\right) \right] \tag{11}$$

The above equation is further optimized for the generation module to generate images that can be successfully classified as target label s.

3.3.5 Full objective

Eventually, we combine all the loss functions to get the final optimization function:

$$\min_{G} \max_{D} \mathcal{L} = \mathcal{L}_{\mathcal{D}} + \mathcal{L}_{\mathcal{G}} \tag{12}$$

And

$$\mathcal{L}_D = \lambda_{adv} \cdot \mathcal{L}_{adv}^{\mathcal{D}} + \lambda_{cls} \cdot \mathcal{L}_{cls}^{\mathcal{D}} \tag{13}$$

$$\begin{aligned} \mathcal{L}_G = \lambda_{adv} \cdot \mathcal{L}_{adv}^{\mathcal{G}} + \lambda_{cls} \cdot \mathcal{L}_{cls}^{\mathcal{G}} \\ + \lambda_s \mathcal{L}_{style} + \lambda_c \mathcal{L}_{content} \end{aligned} \tag{14}$$

where λ_{adv}, λ_{cls}, λ_s and λ_c are weights that apply to losses to allow for better trade-offs in terms of semantics, classification and adaptability.

4 Experimental results

In this section, firstly, we first present the dataset used here and the implementation details of our model. Then, we analyze the generation effect of the model and compare model with other models in both quantitative and qualitative performance. Afterwards, we conduct ablation studies to show the effects of our model. Finally, We also carried out experiments on the robustness of the our model.

4.1 Model details

In order to evaluate the performance of the new model, we built a Chinese characters database containing a variety of Chinese font styles. The data of this database was obtained from the website http://www.Foundertype.com. In addition to traditional fonts, such as Song, Heiti, and Regular scripts, we also collected some fashionable fonts, such as HeiQian style, Spout style, Trendy style, and so on. With the ttf2png script processing, we generate a Chinese characters dataset from the collected fonts. Each typeface ends up with 6000 grey-scale images in 128 × 128 png format. We randomly select 2400 character images from each typeface, where the training set contains 2000 characters and the test set has 400 characters. Meanwhile, we assign a label to each font, for example, 1 for Heiti style, 2 for Regular scripts, 3 for Lishu style, etc. Finally, we convert each label to one-hot vector.

4.1.1 Dataset

In the training process, The only preprocessor we used was the ttf2png script to resize the source and target images to 128 × 128 pixels. The contents encoder and style encoder each have 4 stacked Convolution-InstanceNorm-LeakyReLu blocks and the DenseNet module with 4 blocks. The output channels of each convolutional layer are 64, 128, 256 and 512 respectively.

4.1.2 Model structure

The DenseNet module generates a final feature map of $32 \times 8 \times 8$, The decoder has 4 stacked Deconvolution-InstanceNorm-ReLu and a Convolution-InstanceNorm-LeakyReLu block. Output channels of each deconvolutional layer are 512, 256, 128 and 64 respectively. Finally, the convolutional layer completes the construction of the image. The learning rate of this model is initialized as 0.0003, and decayed to 0.0002 after 20 epochs, and then decayed to 0.0001 after 50 epochs; The Adam optimizer [18] with a batch size of 1 is used.

4.2 Tasks

There are two main tasks in this section: one-to-one font style transfer and one-to-many font style transfer.

We have employed Heiti as the reference font for the input content. Heiti, also known as Gothic, without serif decoration, with straight, horizontal and vertical strokes, all of which are the same thickness. It has the advantage of simple strokes and rigorous structure.

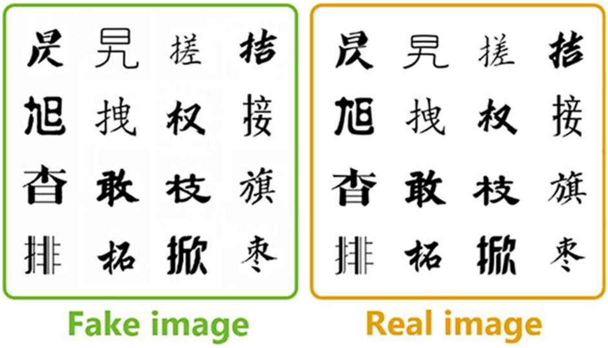

Fig. 4 Detail comparison of generated fonts and real fonts

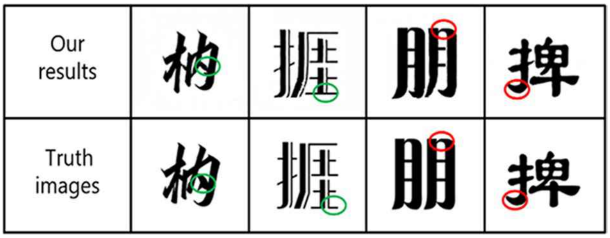

Fig. 5 Results generated from one-to-many experiments

4.2.1 One-to-one

In this experiment, We choose Heiti as the source font and randomly select font as targets to train our model. After an average of 60 repeated training epochs for each font, high-quality images with the target font style are generated. The generated Chinese characters images have a clear graphic structure and strokes. The results show that our model has strong learning and generation capabilities for different fonts. Figure 3 shows realistic and photorealistic images generated using Heiti Style.

Because we employ an unsupervised learning model, there are some differences between the generated images and the ground truth images. The differences are marked with green circles, as in the case of ‘Rui’ and ‘Ai’ in Fig. 4. However, in terms of detail, the generated image is similar to the real image in terms of the style features. For example, the words “Peng” and “Bai” are marked with red circles in Fig. 4. The results show that our model is very successful in learning, with significant feature extraction and generation capabilities.

4.2.2 One-to-many

In this experiment, we choose Heiti as the source font and Song, Jing Hei, Bold Song, Hua Li, Spout, Yiqi Li and Regular script as the target fonts to train the model, Both the source fonts and the target fonts are 1800. After 80 epochs in training, the model was able to produce high-quality font images with a clear structure and complete strokes. As shown in Fig. 5, the improved model has a strong learning ability and generating abilities for different fonts.

4.3 Comparison with other models

In this subsection, we compare our model with some existing methods for Chinese font generation in both quantitative and qualitative performance. Six existing methods are chosen as baselines to compare with our method, including Zi2zi, DCFont, SCFont, EMD, MTfontGAN, ChiroGAN.

Zi2zi is modified version of Pix2pix model, which implements font generation and uses Gaussian noise as category embedding to achieve multi-style transfer.

DCFont replaces the Gaussian noise embedding of the Zi2zi model with the style features extracted from the font feature reconstruction network, and then builds a mapping from the reference font to the target font with the residual blocks.

EMD employs style encoders, content encoders to separate content features and style features, and finally mixers and encoders to achieve new style character generation.

SCFont combines domain knowledge of Chinese characters with deep generative networks to ensure that characters with the correct structure can be generated.

MTfontGAN consists of an encoder with multiple subnets and multiple discriminators. The encoder layers can share feature information to achieve character generation.

ChiroGAN divides the task into three stages, extracting the skeleton using ENet, converting the structure of the source skeleton to the target style using TNet, and finally rendering the skeleton using RNet.

We compared several aspects of the above models and put the results of the comparison into Table 1.

For a fair comparison, we use the same dataset. The training is completed according to these model requirements. Among them, Zi2zi, EMD, DCFont, SCFont, ChiroGAN models cannot implement one-to-many Chinese characters generation experiments, so these models need to be trained several times. After the training, we compared all models in terms of quantitative and qualitative performance.

Qualitative evaluation Figure 6 shows the qualitative results of all models. Because of SANet's ability to learn the mapping between content and style and flexibly match style features and content features that are semantically similar, our model produces images with a clear correct font structure and full natural style features. In contrast, Zi2zi and DCFont generally achieve the overall style transfer, but the details show ghosting artifacts, blurring and even unreasonable strokes. EMD is ability to capture the stroke style characteristics precisely. However, in complex fonts, strokes show incompleteness and fusion between strokes. It can be seen that the model does not guarantee content consistency, and the mapping between learned content features and style features by the model's mixers is not satisfactory. SCFont is generally correct in generating font structure, but the tiny strokes are missed, especially for glyphs with complex shapes. MTfontGAN has been able to generate the correct structure and style, but there are some blurring or breakage in some strokes, which may occur because multiple typefaces trained by the same network cannot guarantee content consistency. ChiroGAN is effective in stroke rendering, but not effective in extracting font skeleton with complex strokes. The generated fonts often have wrong strokes spliced, which leads to font semantic errors.

Quantitative Evaluation While qualitative evaluations provide an intuitive indication of the quality of the generated results for all models, quantitative evaluations can provide higher-level indications of model performance. In this experiment, we employ RMSE (root mean square), PSNR (signal to noise ratio) and SSIM (structural similarity) as pixel-level evaluation metrics between the generated image and the real image, and also use the perceptual-level metric FID [11] to evaluate the features similarity. We use these metrics to compare several fonts generated by all models. The quantitative results are shown

Table 1 Comparison of our model with existing methods

Model	Requirements for new style transfer	What the model learned?	Data format	Is it supervised learning
Zi2zi	Retrain on a lot of training images for a source style and a target style.	The translation from a certain source style to a specific target style.	paired	Yes
DCFont				
EMD	One or a small set of style/content reference images.	The feature representation of style/content.	paired	Yes
SCFont	Need to acquire the Prior-knowledge of new styles	Learn the conversion of the target font strokes and the style features of the target strokes	paired	Yes
MTfontGAN	Retrain on a lot of training images for a source style and many target style.	The translation from a certain source style to a specific target style.	paired	Yes
ChiroGAN	Need to pre-train the stroke structure and style of the target font	Transformation of source font to target font skeleton and stroke style of target font	unpaired	No
Our Model	One or a small set of content reference images.	Learn the mapping between source font content characteristics and target font style characteristics	unpaired	No

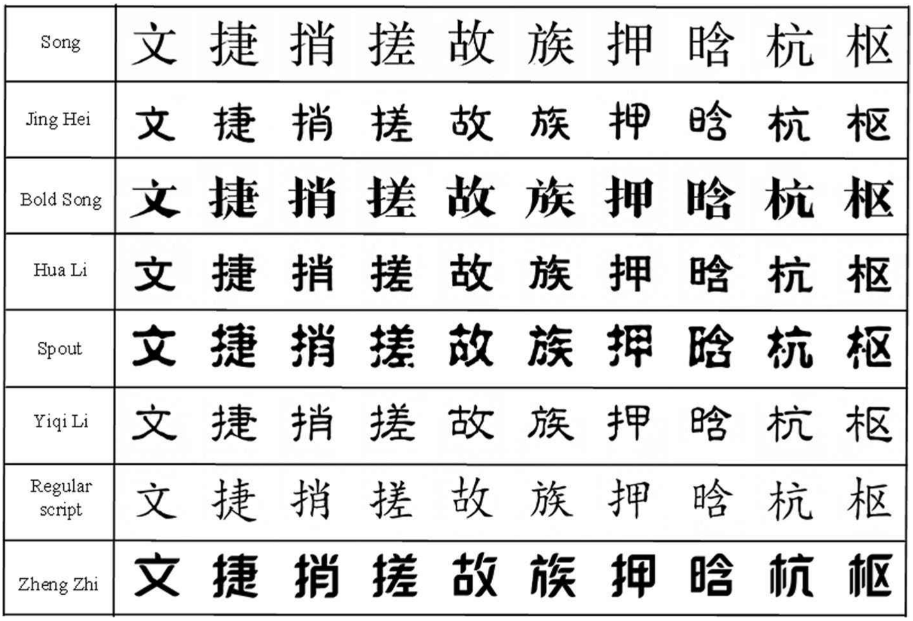

Fig. 6 Comparisons to the stat-of-art methods for font generation

Table 2 Evaluation of our method and existing models for generating LiShu images in RMSE, PSNR and SSIM

Model	RMSE	PSNR	SSIM	FID
Zi2zi	0.4845	7.9684	0.2215	124.15
DCFont	0.4362	8.1457	0.3554	121.43
EMD	0.4211	8.5437	0.3641	98.61
SCFont	0.3975	9.2421	0.4448	79.42
MTfontGAN	0.4211	8.4352	0.4942	75.82
GhiroGAN	0.3542	9.9485	0.5096	72.84
Our Model	0.4898	9.9384	0.5481	67.63

in Table 2, We can see that our approach performs well in several performance metrics, obtaining the highest PSNR and lower FID. in terms of SSIM, ours is comparable to ChiroGAN. These metrics show the superiority of our model.

To evaluate the quality of the generated characters, we also conducted content accuracy experiments on the generated fonts of all models. In this experiment, we evaluated the quality of the generated fonts using model of [25]. One of the prominent features of the proposed network is employing two excitation steps and two inhibition steps, augmenting the accuracy of recognizing characters. The model is trained with real font images, and then tested with the generated images after the training is completed. If the generated characters have the correct structure and complete strokes, our trained character recognition network is able to recognize them correctly. Table 3 shows the recognition accuracy of this model for several fonts generated by the above model.

Table 3 Chinese characters recognition model is used to test the accuracy of Bold song, Spout and Hua Li generated by the four models

Model	Bold Song	Spout	Hua Li
Zi2zi	90.45%	89.84%	90.16%
DCFont	92.71%	92.14%	91.22%
EMD	93.11%	94.53%	92.78%
SCFont	95.94%	96.53%	96.98%
MTfontGAN	98.12%	97.69%	98.15%
GhiroGAN	94.33%	95.40%	94.28%
Our model	**98.46%**	**98.14%**	**98.81%**

As can be seen from the data in Table 3, For zi2zi, DCFont and EMD, their content accuracy results may be related to the quality of the generated images. GhiroGAN generates characters with a clearer structure and correct style. But there is an error in the strokes splicing, which leads to a poor content accuracy. Our model has excellent content accuracy in generating fonts compared to other models. To some extent, this indicates that our model has high performance.

4.4 Ablation study

To study the role of each component of our model, this subsection performs ablation experiments on our model. Figure 7 shows the results generated with the complete model. We remove different sub-networks and loss functions, which are used to analyze the effect of each component.

Effect of skip-connection U-Net Figure 7(A) displays the generation results using a common encoder-decoder instead of U-Net as the generative network. We can easily see that the generated characters have defects in stroke and radical structure and blurring in details. This is due to the large loss of effective information in the convolutional layer transfer process. The use of U-Net allows the fusion of low-level and high-level feature information, so that the neural network can extract more effective feature information.

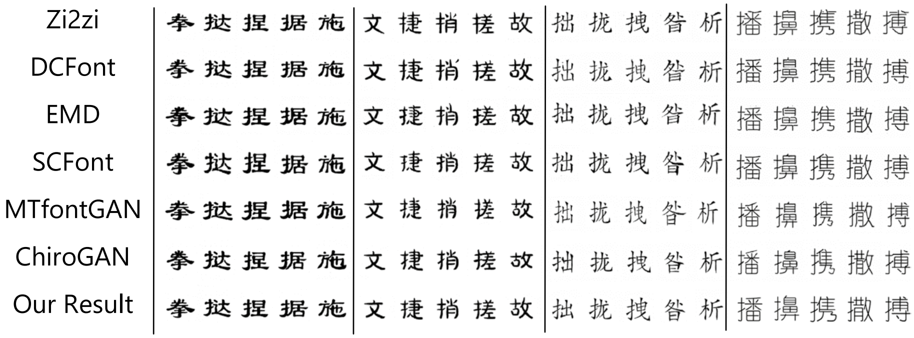

Fig. 7 Demonstrated the results of ablation experiments

Effect of SANet Instead of SANet, we use U-Net's skip connection to combine content features and style features directly with upsampling directly. The generated results are shown in Fig. 7(B), where the font local style and the font content cannot be combined correctly, leading to incorrect font structure and unnatural strokes. Therefore, it can be inferred that SANet can not only maintain the content structure efficiently, but also easily combine style features to enrich the global style and local style statistics.

Effect of relativistic discriminator In this sub-experiment, we used Standard GAN's discriminators instead of Relativistic GANs (RGANs). During the training process for some fonts, as shown in Fig. 7(C), our model always has significant gradient vanishing and finally fails to generate the target font. Therefore, it can be demonstrated that the use of RGANs helps the stability of training and generates higher-quality images.

4.5 Robustness experiments

To examine the robustness of the new model, we trained the model with different size samples. The original training set was 2000. Among these samples, different numbers of samples were randomly selected 2000, 1700, 1400, 1100, 800 and 500 respectively. Our model was chosen with bold as the input font and Bold Song as the target font.

As shown in Fig. 8, the generated fonts of our model do not significantly degrade the generated visuals with the reduction of training samples, whose structure and style still remain better. When the training samples are reduced to 1100, the strokes and details of the generated fonts show small defects. For example, for the character "Min", the strokes in the lower right corner of the generated image are gradually eroded with decreasing number of samples. When the number of samples is lower than 800, the radical of "Xia" gradually dissolve. It can be concluded that the visual effect of the generated fonts does not deteriorate significantly when the number of samples is more than 800. The structure and style of the Chinese characters can still be maintained relatively well. It is shown that our model has good robustness, which allows the generated images to maintain the source font structure and the target font style when the training samples are gradually reduced. However, it is obvious that the network does not learn the features accurately enough when the training samples are reduced to a certain level.

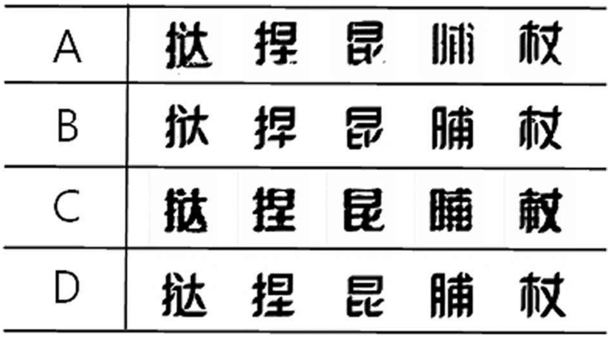

Fig. 8 Bold Song fonts generated by our model at different sizes of training samples

5 Conclusion

In this paper, we propose a novel unsupervised learning generative adversarial network model to accomplish the font style transfer task. The model adopts a modified U-Net as the generator network, which can accurately extract different size feature information. Moreover, we realize the effective combination of content structure and style features with the SANet network. Finally generate the required typeface image. We employ the loss function of the relative discriminator to ensure the stability of the model training. Numerous experiments have shown that our model has the best performance in terms of both quantitative results and visual effects compared to other existing methods. At the same time, Our model has little requirement for dataset and training conditions, which is more convenient for font designers to apply to practical production.

However, our model works mainly on printed Chinese fonts because the printed font design is more regular and therefore the style features can be considered as definite potential embeddings. As for the handwritten font, its design may be influenced by a variety of conditions, which requires the use of large-scale biologically meaningful networks in order to meet the learning of handwriting font styles. In the future, we focus our work on Spiking neural networks [43]. In addition, Yang proposed several efficient, low-power, real-time digital neuromorphic systems for large-scale biologically meaningful networks in [41, 42, 44, 45]. Its excellent performance and near-realistic simulated neurons have greater improvement on training efficiency and style extraction ability of handwritten fonts, which is helpful for complex handwriting font generation tasks.

6 Acknowledgments

This work was supported by Guangzhou Science and Technology Project (201904010107), Guangdong Provincial Natural Science Foundation of China (2019A1515010793), Guangdong Province Science and Technology Project (2016B090918071).

References

1. Atarsaikhan G, Iwana BK, Narusawa A, Yanai K, Uchida S (2017, November). Neural font style transfer. In 2017 14th IAPR international conference on document analysis and recognition (ICDAR) (Vol. 5, pp. 51-56). IEEE.
2. Azadi, S., Fisher, M., Kim, V. G., Wang, Z., Shechtman, E., & Darrell, T. (2018). Multi-content Gan for few-shot font style transfer. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7564-7573).
3. Baxter W, Govindaraju N (2010, February). Simple data-driven modeling of brushes. In proceedings of the 2010 ACM SIGGRAPH symposium on interactive 3D graphics and games (pp. 135-142).
4. Chang B, Zhang Q, Pan S, Meng L (2018, March) Generating handwritten chinese characters using cyclegan. In 2018 IEEE winter conference on applications of computer vision (WACV) (pp. 199-207). IEEE.
5. Chen J, Ji Y, Chen H, Xu X (2019) Learning one-to-many stylised Chinese character transformation and generation by generative adversarial networks. IET Image Process 13(14):2680–2686
6. Choi Y, Choi M, Kim M, Ha JW, Kim S, Choo J (2018) Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8789–8797).
7. Gao, Y., & Wu, J. (2020, April). GAN-based unpaired Chinese character image translation via skeleton transformation and stroke rendering. In proceedings of the AAAI conference on artificial intelligence (Vol. 34, no. 01, pp. 646-653).

8. Gholizadeh-Ansari M, Alirezaie J, Babyn P (2020) Deep learning for low-dose CT denoising using perceptual loss and edge detection layer. J Digit Imaging 33(2):504–515
9. Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y (2014) Generative adversarial networks. arXiv preprint arXiv:1406.2661.
10. Hayashi H, Abe K, Uchida S (2019) GlyphGAN: style-consistent font generation based on generative adversarial networks. Knowl-Based Syst 186:104927
11. 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 Advances in neural information processing systems:30
12. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).
13. Huang, Y., He, M., Jin, L., & Wang, Y. (2020, August). RD-GAN: few/zero-shot Chinese characters style transfer via radical decomposition and rendering. In European conference on computer vision (pp. 156-172). Springer, Cham.49
14. Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).
15. Ji G, Wang Z, Zhou L, Xia Y, Zhong S, Gong S (2020) SAR image colorization using multidomain cycle-consistency generative adversarial network. IEEE Geosci Remote Sens Lett 18(2):296–300
16. Jiang, Y., Lian, Z., Tang, Y., & Xiao, J. (2017). Dcfont: an end-to-end deep chinese font generation system. In SIGGRAPH Asia 2017 technical briefs (pp. 1-4).
17. Jiang, Y., Lian, Z., Tang, Y., & Xiao, J. (2019, July). Scfont: structure-guided chinese font generation via deep stacked networks. In proceedings of the AAAI conference on artificial intelligence (Vol. 33, no. 01, pp. 4015-4022).
18. Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
19. Lake BM, Salakhutdinov R, Tenenbaum JB (2015) Human-level concept learning through probabilistic program induction. Science 350(6266):1332–1338
20. Lee J (1999) Simulating oriental black-ink painting. IEEE Comput Graph Appl 19(3):74–81
21. Lian, Z., Zhao, B., & Xiao, J. (2016). Automatic generation of large-scale handwriting fonts via style learning. In SIGGRAPH Asia 2016 technical briefs (pp. 1-4).
22. Lian Z, Zhao B, Chen X, Xiao J (2018) EasyFont: a style learning-based system to easily build your large-scale handwriting fonts. ACM Transactions on Graphics (TOG) 38(1):1–18
23. Minoofam SAH, Dehshibi MM, Bastanfard A, Eftekhari P (2012) Ad-hoc Ma'qeli script generation using block cellular automata. J Cell Autom 7(4):321–334
24. Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
25. Modhej N, Bastanfard A, Teshnehlab M, Raiesdana S (2020) Pattern separation network based on the hippocampus activity for handwritten recognition. IEEE Access 8:212803–212817
26. Musto, L., & Zinelli, A. (2020). Semantically adaptive image-to-image translation for domain adaptation of semantic segmentation. arXiv preprint arXiv:2009.01166.
27. Park, D. Y., & Lee, K. H. (2019). Arbitrary style transfer with style-attentional networks. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 5880-5888).
28. Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
29. Ronneberger O, Fischer P & Brox, T (2015, October) U-net: convolutional networks for biomedical image segmentation. In international conference on medical image computing and computer-assisted intervention (pp. 234-241). Springer. Cham.
30. Sun, D., Ren, T., Li, C., Su, H., & Zhu, J. (2017). Learning to write stylized chinese characters by reading a handful of examples. arXiv preprint arXiv:1712.06424.
31. Tian Y (2020) :ReWrite.https://github.com/kaonashityc/Rewrite/, accessed August 2020
32. Velek, O., Liu, C. L., & Nakagawa, M. (2001, September). Generating realistic kanji character images from on-line patterns. In proceedings of sixth international conference on document analysis and recognition (pp. 556-560). IEEE.
33. Wang, Y., Wang, H., Pan, C., & Fang, L. (2008, March). Style preserving Chinese character synthesis based on hierarchical representation of character. In 2008 IEEE international conference on acoustics, speech and signal processing (pp. 1097-1100). IEEE.
34. Wang D, Li C, Song H, Xiong H, Liu C, He D (2020) Deep learning approach for apple edge detection to remotely monitor apple growth in orchards. IEEE Access 8:26911–26925

35. Wen, C., Pan, Y., Chang, J., Zhang, Y., Chen, S., Wang, Y., ... & Tian Q (2021) Handwritten Chinese font generation with collaborative stroke refinement. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3882–3891).
36. Wong HT, Ip HH (2000) Virtual brush: a model-based synthesis of Chinese calligraphy. Comput Graph 24(1):99–113
37. Wu, Y., Zhuang, Y., Pan, Y., & Wu, J. (2006, July). Web based chinese calligraphy learning with 3-d visualization method. In 2006 IEEE international conference on multimedia and expo (pp. 2073-2076). IEEE.
38. Wu, L., Chen, X., Meng, L., & Meng, X. (2020, July). Multitask adversarial learning for Chinese font style transfer. In 2020 international joint conference on neural networks (IJCNN) (pp. 1-8). IEEE.
39. Xu S, Lau FC, Cheung WK, Pan Y (2005) Automatic generation of artistic Chinese calligraphy. IEEE Intell Syst 20(3):32–39
40. Yamini K, Swetha KS, Prasanna PL, Swathi MRV, Maddumala VR (2020) Image colorization with deep convolutional open cv. Journal of Engineering Science 11(4):533–543
41. Yang S, Wang J, Deng B, Liu C, Li H, Fietkiewicz C, Loparo KA (2018) Real-time neuromorphic system for large-scale conductance-based spiking neural networks. IEEE transactions on cybernetics 49(7):2490–2503
42. Yang S, Deng B, Wang J, Li H, Lu M, Che & Loparo, K. A. (2019) Scalable digital neuromorphic architecture for large-scale biophysically meaningful neural network with multi-compartment neurons. IEEE transactions on neural networks and learning systems 31(1):148–162
43. Yang S, Gao T, Wang J, Deng B, Lansdell B, Linares-Barranco B (2021) Efficient spike-driven learning with dendritic event-based processing. Front Neurosci 15:97
44. Yang S, Wang J, Hao X, Li H, Wei X, Deng B, Loparo KA (2021) BiCoSS: toward large-scale cognition brain with multigranular neuromorphic architecture. IEEE Transactions on Neural Networks and Learning Systems.
45. Yang, S., Wang, J., Zhang, N., Deng, B., Pang, Y., & Azghadi, M. R. (2021). CerebelluMorphic: large-scale neuromorphic model and architecture for supervised motor learning. IEEE Transactions on Neural Networks and Learning Systems.
46. Yuchen T (2017) “Zi2zi: master chinese calligraphy with conditional adversarial networks” https://github.com/kaonashi-tyc/zi2zi. Accessed 27 Nov 2020
47. Zhang, Z., Wu, J., & Yu, K. (2010, June). Chinese calligraphy specific style rendering system. In proceedings of the 10th annual joint conference on digital libraries (pp. 99-108).
48. Zhang XY, Yin F, Zhang YM, Liu CL, Bengio Y (2017) Drawing and recognizing chinese characters with recurrent neural network. IEEE Trans Pattern Anal Mach Intell 40(4):849–862
49. Zhang, Y., Zhang, Y., & Cai, W. (2018). Separating style and content for generalized style transfer. In proceedings of the IEEE conference on computer vision and pattern recognition (pp. 8447-8455).
50. Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019, May). Self-attention generative adversarial networks. In international conference on machine learning (pp. 7354-7363). PMLR.
51. Zhou, B., Wang, W., & Chen, Z. (2011, July). Easy generation of personal Chinese handwritten fonts. In 2011 IEEE international conference on multimedia and expo (pp. 1-6). IEEE.
52. Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In proceedings of the IEEE international conference on computer vision (pp. 2223-2232).
53. Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable convnets v2: more deformable, better results. In proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 9308-9316).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.