DCFont: An End-To-End Deep Chinese Font Generation System

Yue Jiang
Institute of Computer Science and Technology
Peking University
PR China

Zhouhui Lian*
Institute of Computer Science and Technology
Peking University
PR China

Yingmin Tang
Institute of Computer Science and Technology
Peking University
PR China

Jianguo Xiao
Institute of Computer Science and Technology
Peking University
PR China

ABSTRACT

Building a complete personalized Chinese font library for an ordinary person is a tough task due to the existence of huge amounts of characters with complicated structures. Yet, existing automatic font generation methods still have many drawbacks. To address the problem, this paper proposes an end-to-end learning system, DCFont, to automatically generate the whole GB2312 font library that consists of 6763 Chinese characters from a small number (e.g., 775) of characters written by the user. Our system has two major advantages. On the one hand, the system works in an end-to-end manner, which means that human interventions during offline training and online generating periods are not required. On the other hand, a novel deep neural network architecture is designed to solve the font feature reconstruction and handwriting synthesis problems through adversarial training, which requires fewer input data but obtains more realistic and high-quality synthesis results compared to other deep learning based approaches. Experimental results verify the superiority of our method against the state of the art.

CCS CONCEPTS

• **Computing methodologies** → *Shape modeling*;

KEYWORDS

handwriting, generative models, font style transfer

ACM Reference format:
Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2017. DCFont: An End-To-End Deep Chinese Font Generation System. In *Proceedings of SIGGRAPH Asia 2017 Technical Briefs, Bangkok, Thailand, November 27–30, 2017 (SA '17 Technical Briefs),* 4 pages.
https://doi.org/10.1145/3145749.3149440

1 INTRODUCTION

Making a complete Chinese font library is a time-consuming task. Unlike the English font library that contains only 26 alphabets, the frequently used character set GB2312 is composed of 6763 Chinese characters. Furthermore, the complicated structure and diverse shape of Chinese characters markedly increase the difficulty. Majority of current commercial font generation procedures heavily rely on human design and adjustment, leading to low efficiency and high costs. Similarly, building a personalized Chinese handwriting font library is also a difficult mission for ordinary people since it is hard to write out such huge amounts of complicated characters correctly in a consistent handwriting style.

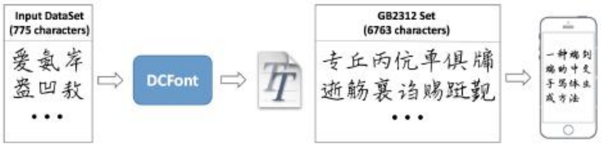


Figure 1: Overview of our system. With a small number of characters written by a user, our system (DCFont) can automatically generate the complete GB2312 font library with 6763 Chinese characters in the user's handwriting style.

Up to now, many attempts have been made to reduce manual work and increase the level of automation. One intuitive solution is to reuse the strokes or radicals contained in the characters designed/written by the user and then assemble them properly to generate other characters. However, human interventions are always required for this type of methods [Zhou et al. 2011; Zong and Zhu 2014] due to the fact that perfect automatic radical/stroke extraction is almost impossible in real applications. More recently, [Lian et al. 2016] solved this problem by proposing a style learning based synthesizing scheme in which human intervention is no longer required.

In last few years, a large amount of research works [Gatys et al. 2015; Johnson et al. 2016] on image style transfer via deep neural networks have been reported, which aim to combine the content of one image with the style of another. With the advent of Generative Adversarial Networks (GANs) [Goodfellow et al. 2014], handwriting synthesis became plausible without exploring any domain knowledge of characters. Recently, [Isola et al. 2016] proposed a new general image-to-image translation framework, "pix2pix", based on the U-net architecture and conditional GAN, which is well suited to solve the image mapping problems.

Possibly inspired by the success of deep learning in generative tasks, several researchers have intended to synthesize Chinese handwritings by using deep neural networks. [Tian 2016] adopted a trickling down CNN structure "Rewrite" to generate Chinese characters. The method is able to produce standard printing fonts but performs badly for handwriting data. More recently, "zi2zi" [Tian 2017] (meaning characters to characters) was proposed based on the "pix2pix" framework [Isola et al. 2016] by adding the category embedding to the generator and discriminator, which results in good synthesizing performance in some specific font styles. [Lyu

*Corresponding author. Email: lianzhouhui@pku.edu.cn

SA '17 Technical Briefs, November 27–30, 2017, Bangkok, Thailand
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-5406-6/17/11…$15.00
https://doi.org/10.1145/3145749.3149440

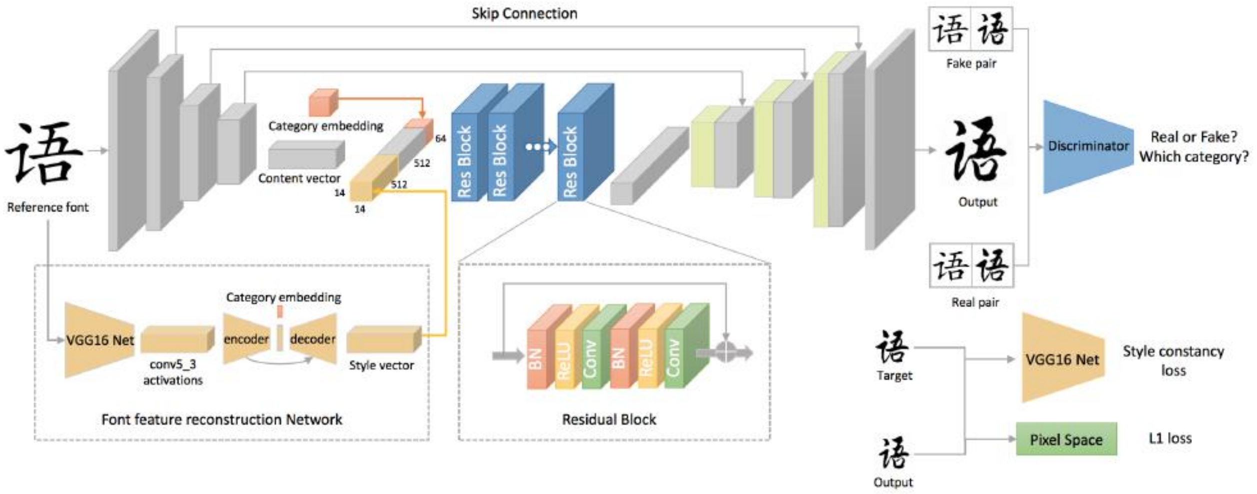


Figure 2: The architecture of the proposed DCFont system that converts characters in the reference font (KaiTi) style into those in the required handwriting style. Specifically, we use a font feature reconstruction network to estimate the deep features of characters, and then the content and style representations are concatenated with category embedding which are fed into the residual blocks and decoder to generate synthesized handwritings.

et al. 2017] regarded Chinese calligraphy synthesis as an image-to-image translation problem, which uses an auto-encoder network to supervise the generator. However, in addition to the poor-quality synthesis results for many characters, the training data size is 6000 in each style that is also too large to be applied in real applications.

In this paper, we propose DCFont, an end-to-end deep Chinese font generation system. As shown in Figure 1, it is capable of generating the complete GB2312 font library with 6763 Chinese characters in a specific handwriting style through learning on a small subset consisting of 775 or even less characters. Different against other existing approaches, our system aims to generate high-quality Chinese handwriting fonts automatically without any human intervention and structure information of characters during both online and offline periods. Experiments show that high-quality synthesis results can be obtained by our system and the proposed method clearly outperforms other state-of-the-art approaches.

2 METHOD DESCRIPTION

In this section, we describe the detailed architecture of the proposed DCFont system that contains two major components, font feature reconstruction network and font style transfer network, respectively. To be specific, given a small number of characters written by a user, the font feature reconstruction network tries to estimate the deep font features of all other characters, and the font style transfer network uses the reconstructed features to convert characters in the reference font (KaiTi) style to corresponding handwriting style. In this way, a complete font library with 6763 Chinese characters in the user's handwriting style can be obtained.

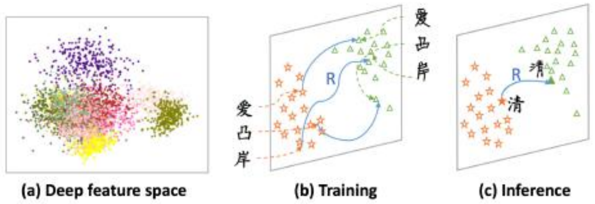
2.1 Font Feature Reconstruction Network

Recently, convolutional neural networks have been widely used to extract features of images. Here, we describe the handwriting style based on a 16-layer VGG [Simonyan and Zisserman 2014] network ϕ pretrained on 100 different fonts.

In order to extract the underlying features in high levels and conserve as much spatial information as possible, we select the output of $conv5\_3$ layer (after relu activations) to represent each character's style. As shown in Figure 3(a), the characters of the same font tend to be clustered together in deep feature space. Thus, we assume that there are similar transformation relationships of different characters from one style to another. We attempt to reconstruct the relationship through the font feature reconstruction network R. In the training process, we learn the transformation relationship from the reference font style to the corresponding handwriting style through the limited handwritten data (see Figure 3(b)). While for other characters that are not written, we estimate their features in deep space via the font feature reconstruction network R (see Figure 3(c)). To be specific, as shown in Figure 2 , the reference character x is sent to the VGG16 net to obtain the font feature of the reference character $\phi_{relu_{5\_3}}(x)$. The whole architecture is similar to the encoder-decoder network. We add the skip connections in the corresponding layers of the encoder and decoder to reduce the information loss during the downsampling process. Besides, we combine the category embedding h_f with the encoded result, which is a 64-dimensional random vector, to make the network distinguish different fonts better. The output is the estimated style representation of a target character in deep feature space $h_s = R(\phi_{relu_{5\_3}}(x))$.

2.2 Font Style Transfer Network

As mentioned above, we regard a handwritten character as the combination of a given content and a required handwriting style. Given a specific character, the content is invariant but its style may vary from person to person. The font style transfer network converts one character in the reference font (KaiTi) style $x \in S$ to a specific handwriting style $y \in T$ with the corresponding content.



(a) Deep feature space (b) Training (c) Inference

Figure 3: Illustration of our font feature reconstruction procedure. (a) the dimension reduction result of *conv*5_3 features through t-SNE. (b) the transformation relationship between reference font style and the handwriting style reconstructed by using the limited number of human-written characters. (c) deep font features inferred by the network.

2.2.1 Network Architecture. We use two separate convolutional neural networks to encode the content and style, respectively. For the style, we obtain the style vector h_s through font feature reconstruction network. For the content, the input reference image x is passed through a series of downsampling layers to encode the image to a $14 \times 14 \times 512$ content vector h_c. In addition, we expand the 64-dimensional category embedding h_f to $14 \times 14 \times 64$ to have the same dimension with content and style vectors. Then, we concatenate these into a vector $h = [h_s, h_c, h_f]$ to represent a specific character. After that, the vector h is fed into five residual blocks [He et al. 2016], which contain two convolutional layers with 3×3 filters. Finally, we utilize a series of upsampling layers to achieve the output image $\tilde{y} = G(h)$. Considering that the reference and generated characters should have similar structure indicating the same content, we connect the low layers with rich details in the content encoder to the corresponding decoder layers directly.

The residual blocks and upsampling layers are similar to the generator that generates samples from vector h. We also use a discriminator to classify generated images as real or fake and identify the font category (see Figure 2). Compared with the classical GAN model, our network has its own potential advantages. On the one hand, we can generate a required character with a specific handwriting according to h. On the other hand, there exists rich spatial information in the content and style encoded vectors, which are both $14 \times 14 \times 512$ instead of one-dimensional.

2.2.2 Loss Functions. In our model, we combine the adversarial loss with the pixel-wise loss and the style constancy loss. As for the adversarial loss, the discriminator should not only be able to identify the fake images, but also correctly classify font styles. Thus, we calculate the log-likelihood of style L_{GANs} and category L_{GANc}, respectively. D maximizes the probability of predicting the correct labels and font categories, which updates parameters by maximizing $L_{GANs} + L_{GANc}$. While G tries to minimize the likelihood of making correct labels on style and maximize the correct category predictions by minimizing $L_{GANs} - L_{GANc}$, where

$$\begin{aligned} L_{GANs} =& E_{x,y \sim p_{data(x,y)}}[logDs(x,y)]+ \\ & E_{x \sim p_{data(x)}, h \sim p_h(h)}[log(1 - Ds(x, G(h)))], \end{aligned} \tag{1}$$

$$\begin{aligned} L_{GANc} =& E_{x,y \sim p_{data(x,y)}}[logDc(x,y)]+ \\ & E_{x \sim p_{data(x)}, h \sim p_h(h)}[logDc(x, G(h))]. \end{aligned} \tag{2}$$

For the sake of resemblance in both content and style in pixel space, we calculate L_{pixel} (L1 distance) to measure the similarity between output and target character images which encourages to generate sharper and clearer images:

$$L_{pixel} = E_{x,y \sim p_{data(x,y)}, h \sim p_h}[\|y - G(h)\|_1]. \tag{3}$$

In order to ensure the style constancy in deep feature space, we also compute the Mean Square Error (MSE) between the activations of generated and target characters, including $relu_{2\_2}$, $relu_{3\_3}$, $relu_{4\_3}$, to get

$$\begin{aligned} L_{style} = & E_{x,y \sim p_{data(x,y)}, h \sim p_h}[\|\phi_{relu_{2\_2}}(y) - \phi_{relu_{2\_2}}(\tilde{y})\|^2 \\ & + \|\phi_{relu_{3\_3}}(y) - \phi_{relu_{3\_3}}(\tilde{y})\|^2 + \|\phi_{relu_{4\_3}}(y) - \phi_{relu_{4\_3}}(\tilde{y})\|^2]. \end{aligned} \tag{4}$$

Finally, the loss function for G is defined as

$$L = \alpha(L_{GANs} - L_{GANc}) + \beta L_{pixel} + \gamma L_{style}. \tag{5}$$

2.2.3 Implementation Details. In our neural network, the input and output character images are all in resolution 224×224. The content encoder contains four down-sampling layers. Each layer is composed of a 5×5 stride 2 convolution, batch normalization and LeakyRelu except the last layer. Then, the combined hidden vector is to be sent to five residual blocks, each consisting of two stacked BN-Relu-Convolution architecture. The decoder consists of four upsampling layers, which contain a 5×5 stride 2 deconvolution, Batch Normalization and Relu except for the last one. We use the tanh activation function after the last deconvolution layer. Additionally, we adapt our discriminator architecture from [Isola et al. 2016].

To accelerate the speed of convergence of our model and improve the quality of generated characters, we pre-trained the network with 20 various fonts and for each font there are 2000 commonly-used Chinese characters. It took about three days for pre-training on a single Pascal Titan X GPU. When learning a specific handwriting style, the network can be finetuned from the pre-trained model in less than two hours. For the sake of fair comparison, we also pre-trained the “zi2zi” method with the same dataset.

3 EXPERIMENTAL RESULTS

In our experiment, we select handwritten data in three different styles to evaluate the effectiveness of our system. The input set consists of 775 commonly-used Chinese characters, as suggested in [Lian et al. 2016], which are able to cover all kinds of strokes appearing in the GB2313 character set.

3.1 Reconstruction Capability

To evaluate the capability of the font feature reconstruction model, we calculate the MSE loss between the inferred features obtained by our model and the original features extracted from the pre-trained VGG16 net. As shown in Table 1, the small reconstruction loss demonstrates the high estimation accuracy of our reconstruction model.

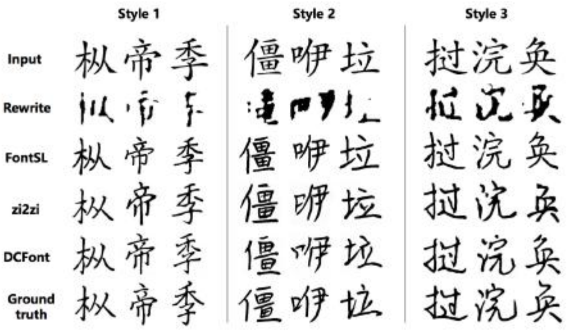


Figure 4: Comparison of synthesis results in three handwriting styles obtained by DCFont and other existing methods.

Table 1: Reconstruction losses computed for three styles.

font	style 1	style 2	style 3
MSE	0.00753	0.00859	0.00942

3.2 Style Transferring Capability

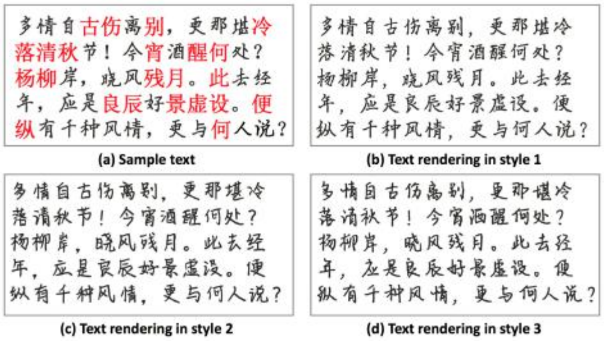
We compare the results of our method with other three recently proposed approaches: "Rewite" [Tian 2016], "FontSL" [Lian et al. 2016] and "zi2zi" [Tian 2017]. As shown in Figure 4, our system shows superiority over others. "Rewite" can only generate blurred images and it is difficult to recognize the content of characters. The characters generated by "FontSL" are quite similar and fail in capturing the tiny differences between similar handwriting styles, although contents are correctly preserved. Synthesis results of "zi2zi" possess some ghosting artifacts and unreasonable strokes in style 2 and 3. On the contrary, our "DCFont" system is able to produce realistic synthesis results and properly preserve important handwriting details such as the start, turn and end regions of strokes.

3.3 Generation of Font Libraries

Finally, the 775 human-written and 5988 machine-generated character images are vectorized, and then packaged together into a GB2312 font library. To visually estimate the quality of these font libraries generated by our system, we use them to render a paragraph in different font styles (see Figure 5). For better demonstration, Figure 5(a) emphasizes the machine-generated characters in red color. Figure 5(b), (c) and (d) show the rendered paragraph in their corresponding font styles. We can see that machine-generated characters are almost indistinguishable from those written by corresponding users. More results can be found in supplementary materials.

4 CONCLUSION AND FUTURE WORK

In this paper, we proposed an end-to-end system for Chinese font generation. It can automatically generate a complete font library in a specific handwriting style based on a small number of human-written characters. Compared with other existing approaches, our system is capable of producing handwritings that are of higher quality and look more realistic. However, as shown in Figure 6, for characters written in very cursive styles, our system still cannot generate satisfactory results. For future work, we are planning to study how to combine the deep neural network with domain knowledge in Chinese characters, so as to generate more reasonable cursive handwritings that correspond well with people's natural writing habits.


多情自古伤离别，更那堪冷落清秋节！今宵酒醒何处？杨柳岸，晓风残月。此去经年，应是良辰好景虚设。便纵有千种风情，更与何人说？

(a) Sample text

多情自古伤离别，更那堪冷落清秋节！今宵酒醒何处？杨柳岸，晓风残月。此去经年，应是良辰好景虚设。便纵有千种风情，更与何人说？

(b) Text rendering in style 1

多情自古伤离别，更那堪冷落清秋节！今宵酒醒何处？杨柳岸，晓风残月。此去经年，应是良辰好景虚设。便纵有千种风情，更与何人说？

(c) Text rendering in style 2

多情自古伤离别，更那堪冷落清秋节！今宵酒醒何处？杨柳岸，晓风残月。此去经年，应是良辰好景虚设。便纵有千种风情，更与何人说？

(d) Text rendering in style 3

Figure 5: Texts rendered using three different font libraries (b-d) generated by our system. (a) shows the text rendered using the font library with only 775 handwritten characters in style 1, where unwritten characters are colored in red.

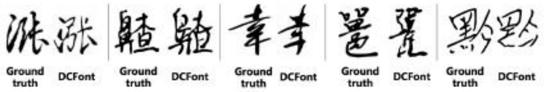


Figure 6: Some unsatisfactory synthesis results.

ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China (Grant No.: 61672043, 61472015 and 61672056), Beijing Natural Science Foundation (Grant No.: 4152022), National Language Committee of China (Grant No.: ZDI135-9), and Key Laboratory of Science, Technology and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology).

REFERENCES

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576* (2015).

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, and David Warde-Farley. 2014. Generative Adversarial Nets. In *NIPS*. 2672–2680.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2016. Image-to-image translation with conditional adversarial networks. *arXiv preprint arXiv:1611.07004* (2016).

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*. 694–711.

Zhouhui Lian, Bo Zhao, and Jianguo Xiao. 2016. Automatic generation of large-scale handwriting fonts via style learning. In *SIGGRAPH ASIA 2016 TB*. 12.

Pengyuan Lyu, Xiang Bai, Cong Yao, Zhen Zhu, Tengteng Huang, and Wenyu Liu. 2017. Auto-Encoder Guided GAN for Chinese Calligraphy Synthesis. *arXiv preprint arXiv:1706.08789* (2017).

Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

Yuchen Tian. 2016. Rewrite: Neural Style Transfer For Chinese Fonts. (2016). Retrieved Nov 23, 2016 from https://github.com/kaonashi-tyc/Rewrite

Yuchen Tian. 2017. zi2zi: Master Chinese Calligraphy with Conditional Adversarial Networks. (2017). Retrieved Jun 3, 2017 from https://github.com/kaonashi-tyc/zi2zi

Baoyao Zhou, Weihong Wang, and Zhanghui Chen. 2011. Easy generation of personal Chinese handwritten fonts. In *ICME*. 1–6.

Alfred Zong and Yuke Zhu. 2014. StrokeBank: Automating Personalized Chinese Handwriting Generation.. In *AAAI*. 3024–3030.