SCFont: Structure-guided Chinese Font Generation via Deep Stacked Networks

Yue Jiang, Zhouhui Lian*, Yingmin Tang, Jianguo Xiao
Institute of Computer Science and Technology, Peking University, Beijing, P.R. China
{yue.jiang, lianzhouhui, tangyingmin, xiaojianguo}@pku.edu.cn

Abstract

Automatic generation of Chinese fonts that consist of large numbers of glyphs with complicated structures is now still a challenging and ongoing problem in areas of AI and Computer Graphics (CG). Traditional CG-based methods typically rely heavily on manual interventions, while recently-popularized deep learning-based end-to-end approaches often obtain synthesis results with incorrect structures and/or serious artifacts. To address those problems, this paper proposes a structure-guided Chinese font generation system, SCFont, by using deep stacked networks. The key idea is to integrate the domain knowledge of Chinese characters with deep generative networks to ensure that high-quality glyphs with correct structures can be synthesized. More specifically, we first apply a CNN model to learn how to transfer the writing trajectories with separated strokes in the reference font style into those in the target style. Then, we train another CNN model learning how to recover shape details on the contour for synthesized writing trajectories. Experimental results validate the superiority of the proposed SCFont compared to the state of the art in both visual and quantitative assessments.

Introduction

Compared to uniform-looking glyphs, now more and more people prefer using personalized fonts, especially those in distinguished handwriting styles, in many scenarios. On the one hand, handwriting styles are flexible to express personality and endow texts with writers' distinctive characteristics and hallmarks. On the other hand, glyphs in personalized handwriting styles bring about dynamic visual perceptions, which are able to attract more attentions in various social networking media.

However, creating a handwriting Chinese font library is still a time-consuming and laborious work with long production cycle. The reasons are threefold: 1) Chinese characters have complex structures and thus yield great variance in writing styles. 2) Unlike the English or Latin typefaces that only contain a small number of glyphs, even the most commonly used Chinese charset (i.e., GB2312) is composed of 6763 characters. It is hard for ordinary people to write out such large amounts of characters while still maintaining the style consistency. 3) The handwriting fonts now available on electronic devices are mainly produced by professional font designers in commercial companies. They rely heavily on elaborate adjustments for each glyph, which is impractical and infeasible for fast font generation aiming at common customers. Thereby, generating Chinese handwriting fonts for ordinary people is still a tough task.

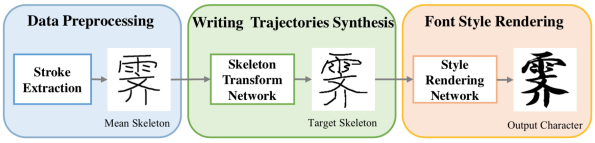


Figure 1: An overview of our system.

The majority of traditional CG-based methods take advantage of structural correlations among Chinese characters to reuse parts of input glyphs. Typically, the input character images are first decomposed into pre-defined strokes or radicals. Then, unseen characters are synthesized by properly assembling the extracted components. Nevertheless, the qualities of many synthesized glyphs are unsatisfactory such that manual interventions are typically required.

Recently, the utilization of deep learning-based approaches enables the font generation problem to be resolved in an end-to-end manner. For one thing, the font generation can be regarded as a style transfer problem where characters in the reference style are transferred to a specific style while maintaining the consistency of the contents. For another, with the advances of generative adversarial networks (GAN), more realistic and higher-quality glyphs can be synthesized. However, the entire generative process is uncontrollable and unpredictable. Blur and ghosting artifacts are often contained in the generated glyphs. Besides, for characters with complicated structures and/or in cursive handwriting styles, those end-to-end approaches often produce results with unreasonable strokes or/and incorrect structures.

In this paper, we propose a structure-guided Chinese font generation system, SCFont, which integrates the prior domain knowledge of Chinese characters with deep stacked networks to synthesize visually-pleasing character images with correct contents. We decompose the font generation task into two separate procedures, namely writing trajecto-

*Corresponding author.

ries synthesis and font style rendering. The whole pipeline is shown in Figure 1. In the first phase, each character is represented as a series of writing trajectories with separated strokes, which is also named as the skeleton of character in this paper. We utilize a multi-stage CNN model to transfer writing trajectories in the reference style into those in the target style. In the second phase, synthesized skeletons of characters are rendered with a specific handwriting style via a GAN model to recover shape details on the contour of glyphs. Finally, the complete font library composed of 6763 characters in the writer's handwriting style can be obtained.

Compared with existing methods in the literature, the proposed SCFont is more capable of handling two crucial requirements in font generation, i.e., structure correctness and style consistency. To be specific, writing trajectories can be synthesized by the learned skeleton flow vectors which indicate how to map the corresponding pixels in the input to those in the output. In this manner, we not only make the learning problem more tractable, which avoids learning to generate writing trajectories from scratches, but also provide a natural way of preserving the identity and structure of glyphs. Additionally, to further enhance style similarity with the original handwritings, a stacked generation network is designed to refine the ambiguous parts or/and artifacts and precisely recover the stylish details on the contour of glyphs.

Related Work

Up to now, large numbers of methods on font synthesis have been proposed. Generally speaking, those existing methods can be classified into two categories: Computer Graphics-based methods and Deep Learning-based methods.

Computer Graphics-based Methods (Xu et al. 2005) proposed the shape grammar to represent each character in a multi-level manner and generate calligraphy via a constraint-based reasoning system. Considering personal handwriting style biases, the shape grammar was augmented by a statistical modeling for a specific writer (Xu et al. 2009). Later, the Chinese Character Radical Composition Model (Zhou, Wang, and Chen 2011) and StrokeBank (Zong and Zhu 2014) were proposed to synthesize characters by means of mapping the standard font components, i.e., strokes or radicals, to their handwritten counterparts. (Lin et al. 2015) also designed an algorithm to synthesize a given character in the target style through assembling extracted components according to the human labeled position and size information. However, expert knowledge is always required to manually define basic constructive elements. Besides, human supervision and fine-tuning are inevitable to attain perfect stroke/radical extraction results. More recently, (Lian, Zhao, and Xiao 2016) developed the "FontSL" system to learn stroke shape and layout separately in which human intervention is avoided. But, the composition rules are based on strong assumptions of the reference font style, which are not well suited for handwriting fonts with huge shape difference compared to the reference data.

Deep Learning-based Methods Glyphs of Chinese characters can be represented as the combination of the given content and a specific font style. Many researchers attempted to synthesize Chinese characters using models adapted from style transfer (Gatys, Ecker, and Bethge 2015; Johnson, Alahi, and Fei-Fei 2016; Chen et al. 2017). "Rewrite" was proposed by (Tian 2016) to transfer a given character from the standard font style to a target style. In addition, font generation can be considered as an instance of image-to-image translation problem (Isola et al. 2017; Zhu et al. 2017b; 2017a; Kim et al. 2017; Yi et al. 2017), which transforms the image from one domain to another while preserving the content consistency.

With the help of generative adversarial networks (Goodfellow et al. 2014; Mirza and Osindero 2014; Radford, Metz, and Chintala 2015; Odena, Olah, and Shlens 2017), more realistic and higher-quality character images can be synthesized. For instance, "zi2zi" (Tian 2017) exploited a conditional GAN-based model to generate characters with the category embedding added to both the generator and discriminator. "DCFont" (Jiang et al. 2017) employed a font feature reconstruction network to estimate the features of unseen characters and then synthesized a given content character via a generative network. After that, many font generation networks (Azadi et al. 2018; Lyu et al. 2017; Sun et al. 2017; Chang, Gu, and Zhang 2017; Chang et al. 2018; Zhang, Zhang, and Cai 2018) were proposed, which often utilized CNN-based models to capture the style of a specific writer and then apply it to unseen characters. Yet, the low-quality synthesis results hinder the practical uses of these approaches. For one thing, ghosting artifacts and blurred strokes are common problems existing in the generated characters. For another, those deep neural networks often produce incomplete or unreasonable strokes for characters with complicated structures.

To address the problems mentioned above, we follow the idea of flow prediction to generate high-quality writing trajectories and then recover contour details on them. As for flow prediction, (Dosovitskiy et al. 2015) proposed FlowNet to predict the flow field from a pair of images. More recently, FlowNet2.0 (Ilg et al. 2017), stacked networks of FlowNet, was designed to solve small displacements and noisy artifacts in the estimated flow. There also exist some works (Zhou et al. 2016; Park et al. 2017) utilizing the predicted flow to synthesize novel views of objects via learning pixel-to-pixel correspondence between the input and output views.

Method Description

In our SCFont system, the font generation task is decoupled into two steps: writing trajectories synthesis and font style rendering. In the first step, we use SkelNet, which can learn the flow of points on the writing trajectories from the reference style to the target style, to predict the target trajectories of characters. The input of SkelNet is the mean writing trajectory of a given character and the output is the predicted flow. After applying the flow to the reference trajectory, an image of writing trajectory in the desired style is obtained. In the second step, StyleNet is trained for enriching the trajectories with a specific font style on contours. The entire pipeline requires only a few samples in the training stage

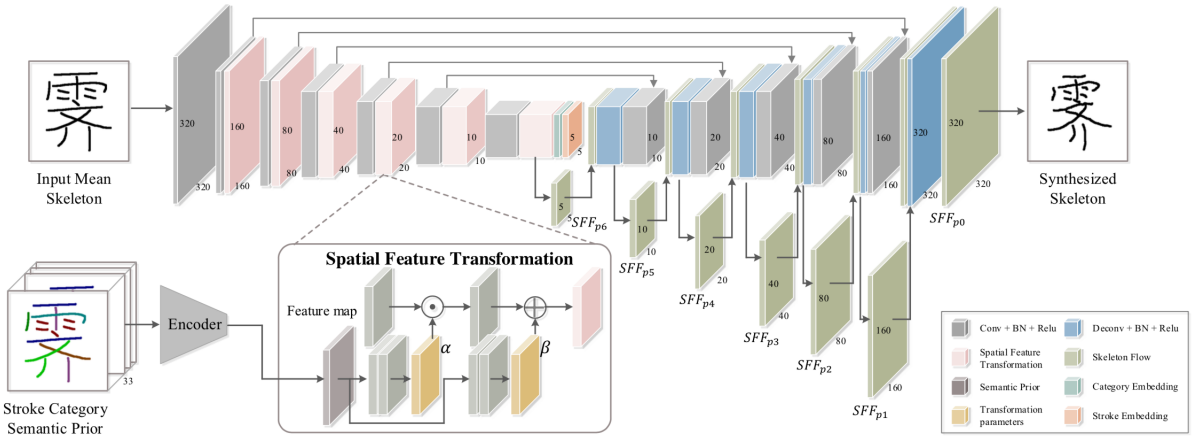


Figure 2: The architecture of the first-stage netwok in the SkelNet. It predicts the skeleton flow, in a coarse to fine way, to build correspondence between pixels in the reference and target writing trajectories so as to synthesize the target skeleton image.

and then other unseen characters can be synthesized to obtain the complete font library.

Data Preprocessing

We adopt the method presented in (Lian, Zhao, and Xiao 2016) to automatically extract strokes. Followed by that, in order to establish the correspondence between different font skeletons of a given stroke, a non-uniform sampling method is used to sample the same number of key points for each stroke in the writing trajectories containing the connection, ending and important turning points, to preserve the structure information as much as possible. The key points are connected to form a single-pixel wide skeleton, and then expanded to the image with certain width skeleton. In addition, we collected a set of Chinese fonts in different styles and extracted the writing trajectory for every glyph, and then calculated the mean value of coordinates in the sampled points on the strokes to get the mean skeleton for each character, which is used as our reference writing trajectory.

More specifically, the skeleton flow field is defined as SFF, where each element $SFF^{(i)}$ is a 2-D coordinate vector specifying the correspondence between $(x_r^{(i)}, y_r^{(i)})$ in the reference writing trajectories I_r and its counterpart $(x_t^{(i)}, y_t^{(i)})$ in the target style I_t

$$SFF^{(i)} = (x_f^{(i)}, y_f^{(i)}) = (x_t^{(i)} - x_r^{(i)}, y_t^{(i)} - y_r^{(i)}), \quad (1)$$

where $i \in I_r$. Considering that it is difficult for CNNs to effectively extract features of such single-pixel skeleton images, we broadcast the single-pixel wide skeletons to those with certain width (e.g., 4 pixels wide).

Skeleton Transformation Network (SkelNet)

Given a mean skeleton image, the goal of SkelNet is to predict the dense skeleton flow of each pixel from the mean writing trajectory to that in the target writing trajectory. The basic architecture of our SkeNet is similar to the FlowNet (Dosovitskiy et al. 2015). Here, we incorporate the stroke category information into the model to guide the transformation. Moreover, stacked networks are also employed to further improve the performance of flow estimation.

Network Architecture. As shown in Figure 2, our SkelNet contains two parts: contracting and expanding. The input skeleton image is first shrunk into high-level semantic features conditioned on the stroke category information. Then, the expanding part estimates the dense flow values in a coarse-to-fine way.

The contracting part contains a series of convolutional layers with a stride of 2, resulting in a total downsampling factor of 64. The expanding part of the network gradually and nonlinearly upsamples the encoded result, taking into account features from the contractive part. To disentangle different font styles and learn the features of different stroke categories, we also concatenate the encoded result of the contracting part with the category embedding h_f and stroke embedding h_s, which indicate the font category and categories of strokes appearing in the target character, respectively.

We adopt a hierarchical prediction architecture and begin estimating the skeleton flow field from a low-resolution (5×5) map. As illustrated in Figure 2, each time we concatenate the deconvolution result with the feature map from the corresponding contrasting layer and the upsampled coarse flow estimation. In this manner, the network is able to preserve more details and refine the estimation results progressively. Besides, we add an additional convolutional layer with stride one after each concatenation operation to make the estimated flows as smooth as possible.

Furthermore, we stack the second stage flow estimation network to further refine the predicted flow values, which is similar to the first stage except without spatial feature layers.

Loss Function. We train the entire stacked network in an end-to-end manner and supervise on two stages simultaneously. The multi-scale losses are accumulated, and the total loss is defined as the sum of weighted losses in all dimensions

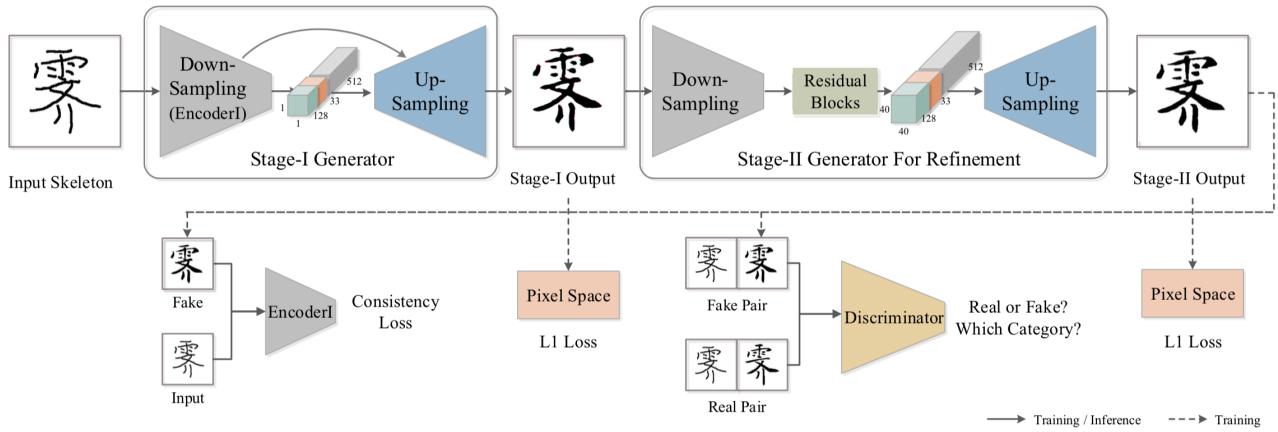


Figure 3: The architecture of StyleNet, which contains two-stage generative networks and a discriminative network. It renders the skeleton image in a specific font style to recover stylish details on the contour of glyphs.

$$L_{skel} = \sum_{i=1}^{2} \lambda_0 loss_0^i + \lambda_1 loss_1^i + \lambda_2 loss_2^i + \lambda_3 loss_3^i + \lambda_4 loss_4^i + \lambda_5 loss_5^i + \lambda_6 loss_6^i, \quad (2)$$

where the weight of each loss layer λ_j increases with the dimension, and $loss_j^i (j \in [0, 6])$ denotes the endpoint error (EPE) in the stage i, i.e., the average Euclidean distance between the predicted flow vector and the ground truth over all pixels.

Stroke Category Semantic Prior. We define 33 categories of strokes according to their shapes and meanings in the characters. The stroke category of every pixel in the image can be regarded as a type of semantic information in the Chinese character. The SkelNet uses the spatial feature transform (SFT) layer (Wang et al. 2018) following each convolution layer in the contracting part. The SFT layer tries to learn the parameters of affine transformation based on the category prior, which are then applied to the intermediate feature maps. Here, the category prior is defined as

$$S = (S_1, S_2, S_3, S_k, ..., S_K), \quad (3)$$

where S_k^i indicates whether the pixel $i (i \in I_r)$ belongs to the stroke category k and K is the total number of stroke categories. The input of j^{th} SFT layer is the feature map F_{in}^j and the conditions P^j coming from the encoded results of the stroke category map S. This layer learns parameters α^j and β^j through respective convolutional layers to produce the output with the same dimension as F_{in}^j (see Figure 2). Then, we have

$$F_{out}^j = F_{in}^j \odot \alpha^j + \beta^j, \quad (4)$$

where F_{out}^j indicates the ouput of the SFT layer and $\odot$ denotes the element-wise multiplication.

Style Rendering Network (StyleNet)

The goal of StyleNet is to render the writing trajectories in a specific handwriting style and recover shape details on the contour of glyphs. StyleNet utilizes an image-to-image transformation model with adversarial network to transform the input skeleton to the target image. Meanwhile, we try to preserve the structural consistency between the input and output images so as to make the generation process more controllable. The whole generation process contains two stages. Stage-I generator transforms the skeleton image to its corresponding character image while Stage-II generator fixes artifacts to further polish synthesis results.

Stage-I Generator. As shown in Figure 3, the input skeleton image x is sent to a sequence of downsampling layers to encode the skeleton image to high-level representation. Similar to SkelNet, we also combine the font category embedding h_f and stroke embedding h_s with the encoded feature. In this way, the network could better build the latent style space and distinguish the characteristics of different strokes. Then, deconvolution layers are used to upsample the features progressively. Meanwhile, the output of each deconvolution layer is concatenated with its corresponding layer in the encoder to propagate the low-level information directly.

Stage-II Generator for Refinement. Since the generated images of the Generator I inevitably has some ghosting artifact and blur, we employ the Stage-II generator to refine undesired results. The input $G_1(x)$ is firstly downsampled to 40×40, then sent to some residual blocks (He et al. 2016), finally up-sampled to the target resolution. The encoder-decoder architecture used in the generation network is intrinsically well-suited to fix small displacements and maintain the consistent attributes of the given input image.

Loss Function. The loss function of our StyleNet consists of adversarial loss, consistency loss and pixel space loss.

The adversarial networks have achieved great success in generative problems. In our model, D examines not only the genuineness $(Ds(\cdot))$, but also the font category of the image pair $(Dc(\cdot))$, namely, the input and its corresponding fake or real character image (see Figure 3). Instead of downsampling the image to one dimension, we classify whether the $N \times N$ patch is real or fake. The aver-

age of classification results for all patches in the image is the final output. During training, the discriminator tries to maximize $L_{GANs} + L_{GANc}$ while the generator minimizes $L_{GANs} - L_{GANc}$, where

$$\begin{aligned} L_{GANs} =& E_{x,y\sim p_{data(x,y)}}[logDs(x,y)]+ \\ & E_{x\sim p_{data(x)}}[log(1-Ds(x,G(x)))] \end{aligned} \tag{5}$$

$$\begin{aligned} L_{GANc} =& E_{x,y\sim p_{data(x,y)}}[logDc(x,y)]+ \\ & E_{x\sim p_{data(x)}}[logDc(x,G(x))]. \end{aligned} \tag{6}$$

Here, x and y are the input skeleton image and the target character image, respectively. G_1, G_2 denote the Stage-I and Stage-II generators, and G represents the whole two-stage generator $\{G_1, G_2\}$.

In order to make the output character image preserve structure attributes of the input skeleton, we utilize the consistency loss to penalize the difference of high-level features, which are results of the encoder Enc_I in the Stage-I generator:

$$L_{const} = E_{x\sim p_{data(x)}}[\|Enc_I(x) - Enc_I(G(x))\|_1]. \tag{7}$$

We also calculate the weighted L1 loss in the pixel space for G_1 and G_2, focusing more on pixels of the character rather than the background, to enforce the generated image to resemble the real one

$$\begin{aligned} L_{pixel} =& E_{x,y\sim p_{data(x,y)}} M \odot [\|y - G_1(x)\|_1 + \\ & \|y - G_2(x)\|_1], \end{aligned} \tag{8}$$

where $M \in \{1,5\}^{H\times W}$ is the weighted visuality map. The weight of pixels on the character is 5 times of those in the background.

Finally, the total loss function for G can be computed by

$$\begin{aligned} L_{style} =& \lambda_{ad}(L_{GANs} - L_{GANc}) + \lambda_{pix} L_{pixel} + \\ & \lambda_{con} L_{const}. \end{aligned} \tag{9}$$

Experimental Results

In this section, we firstly introduce the dataset and implementation details of our method. After that, we evaluate the performance from different aspects to verify its advantages over other approaches. Finally, the networks are fully analyzed to confirm the validity of the entire architecture.

Dataset

We select to conduct experiments on seventy Chinese font libraries in different handwriting styles as well as designing styles. Here, the trajectories of stokes in all character images have been attained by stroke extraction techniques and a few wrong extraction results have been manually corrected. We assume the correctness of stroke extraction results since our major concern is how the proposed font style transfer method works. We conduct experiments on various fonts with different degrees of difficulty, from approximately standard to cursive or stylish handwritings with huge shape deformation.

Although our method already can generate attractive results by learning quite a few (e.g., 50) samples, to ensure more stable and high-quality performance in real applications, we take the optimal input set (OptSet) presented in (Lian, Zhao, and Xiao 2016) which contains 775 characters as the input set. It covers all kinds of components in the GB2312 character set. Under this setting, the network could be able to "see" enough samples to effectively handle glyphs with complicated structures or in very cursive styles.

Implementation Details

In our experiment, the input and output character images are both of size $320 \times 320 \times 3$. We use mini-batches with size 16 and train the model with the Adam optimizer. The learning rate is initialized as 0.001 and is decayed by a half after 5 iterations. In the SkelNet, $\lambda_j = 2^{-j}$ for $j \in [0, 6]$; in the StyleNet, λ_{pix}, λ_{con} , and λ_{ad} are set to $100, 15$ and 1, respectively. We selected 25 fonts, 6000 characters randomly chosen from GB2312 charset for each font, to pre-train the entire network. When it comes to a specific font to be generated, we fine-tune the model by a small number (e.g., 775) of writing samples. This manner brings about two benefits. Firstly, compared with training from scratch, it greatly speeds up the rate of convergence of the model and enhances the quality of generated results. Secondly, the pre-trained process enables the network to "see" diverse fonts so as to learn the underlying style representation, which makes it more robust for complex handwritings with great shape difference against the reference style.

Performance Evaluation

In this subsection, we first compare SCFont with some recently proposed methods to prove the effectiveness of our method. Additionally, the results are also quantitatively analyzed. Then, a user study is carried out to measure the realism and style-similarity qualitatively. Finally, texts rendered in the styles generated by our method are illustrated to indicate its feasibility in real applications.

Comparisons. We compare the results of SCFont with 4 existing methods: "pix2pix"(Isola et al. 2017), "DCFont"(Jiang et al. 2017), "zi2zi"(Tian 2017) and "FontSL"(Lian, Zhao, and Xiao 2016). The former 3 methods are deep learning-based approaches while "FontSL" is a type of CG-based method. For fair comparison, the three deep learning-based approaches are all pre-trained with the same dataset mentioned above and fine-tuned on the glyphs in a specific font style.

As depicted in Figure 4, our method produces realistic and high-quality results showing great superiority over others in visual appearance. Although the deep learning based methods ("pix2pix", "DCFont", and "zi2zi") seem to be able to transfer the overall font style, the generated results are still in low-quality. When zooming in the details, ambiguity and severe artifacts often exist in the synthesized glyphs especially for those with complicated structures (see Figure 4 (a), (c), (e)). As for CG-based methods (e.g.,"FontSL") that make full use of domain knowledge of Chinese characters, they could, to some extent, guarantee the structure correctness. However, it fails to precisely capture the overall characteristics as well as local details and thus lacks style-similarity (see Figure 4(b)). When glyphs in the target font style (e.g., FZTLJW) look dramatically different against those in the

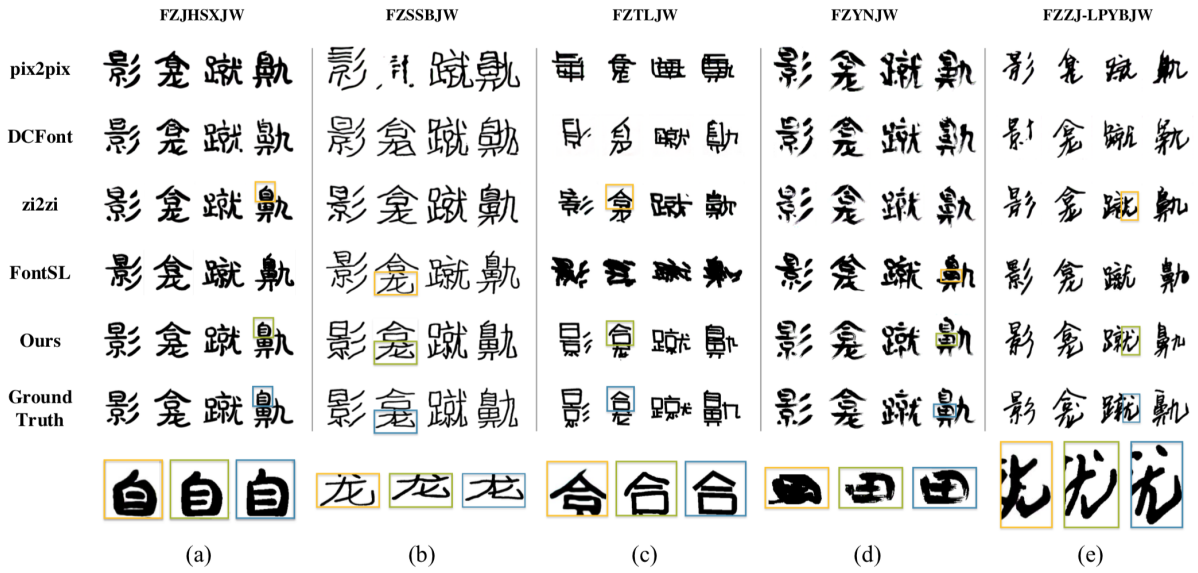


Figure 4: Comparison of synthesized glyphs in five different styles obtained using our SCFont and other four existing methods.

Table 1: Quantitative evaluations of our SCFont and other 4 methods.

Method	FZJHSXJW		FZSSBJW		FZTLJW		FZYNJW		FZZJ-LPYBJW	
	L1 loss	IOU	L1 loss	IOU	L1 loss	IOU	L1 loss	IOU	L1 loss	IOU
pix2pix	0.1851	0.4007	0.2290	0.1846	0.1687	0.2048	0.1782	0.4135	0.1491	0.2331
DCFont	0.1630	0.4125	0.1906	0.1672	0.1459	0.1223	0.1640	0.3908	0.1327	0.1808
zi2zi	0.1483	0.4682	0.1936	0.2598	0.1558	0.2431	0.1527	0.4328	0.1400	0.2688
FontSL	0.1943	0.4087	0.2395	0.1928	0.1560	0.1839	0.2272	0.3609	0.1492	0.2379
SCFont	**0.1173**	**0.5459**	**0.1627**	**0.3163**	**0.1188**	**0.3574**	**0.1245**	**0.5442**	**0.1191**	**0.3197**

Table 2: User study results.

Font	Accuracy
FZJHSXJW	0.5207
FZSSBJW	0.4971
FZTLJW	0.5119
FZYNJW	0.4986
FZZJ-LPYBJW	0.5299
Average	**0.5116**

reference style, poor-quality synthesis results will be generated by both types of methods mentioned above (see Figure 4(c)). On the contrary, the proposed method not only guarantees the correctness in structures, but also preserves stylish details on the contour.

In addition to visual appearance, SCFont also outperforms others in quantitative measurements. We calculate the L1 loss, IOU (Intersection over Union) between synthesized images and the corresponding ground truth. As we can see from Table 1, our method achieves the lowest L1 loss and the highest IOU accuracy, which clearly demonstrate the superiority of the method over others.

User Study. We also conduct a user study to examine the realism and style-similarity of synthesis results generated by our method. To be specific, each participant was presented with an online questionnaire composed of 100 ground-truth and 100 synthesized character images in the same font style, which are both randomly selected and placed in the table each time. Meanwhile, 50 randomly chosen ground-truth character images are also shown to participants as the selecting guidance. Participants were required to find out all character images which they think are machine-generated.

In total, 124 educated people of different age groups took part in this test. The average accuracy of distinguishing machine-generated glyphs from the ground truth in five different font styles is 51.16% (see Table 2). The value is approximate to random selection (50%) indicating that it is struggling for participants to determine whether a given character image is machine-generated or not.

Text Rendering Effects. To illustrate the effectiveness of our method in real applications, we vectorize the original human-created and machine-generated character images together and package them into a GB2312 font library. As shown in Figure 5, the same texts are rendered in different font styles. To indicate which characters are generated by our method, Figure 5(a) marks the machine-generated characters in red color. The entire text rendered using the font library generated by our SCFont is consistent in style and it is quite hard to distinguish the machine-generated glyphs from human-created ones.

Analysis of Network Architecture

Stroke Category Prior in the SkelNet. In the SkelNet, we incorporate the stroke category prior into the contrasting part by spatial feature transformation layers. To verify its effectiveness, we compare it with a SkelNet variant without using the prior information. Figure 6 indicates that with the guidance of stroke category prior, severe distortions on the synthesized writing trajectories can be greatly alleviated, especially on the stroke overlapping regions. Besides, the

Figure 5: Texts rendered in 5 different font libraries generated by our SCFont. Synthesized characters in (b-f) are marked in red in (a).

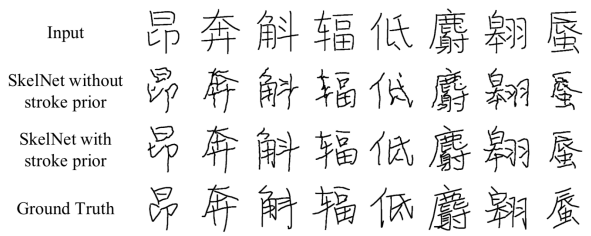


Figure 6: The effect of stroke category prior knowledge. With the guidance of stroke category prior, distortions can be greatly alleviated.

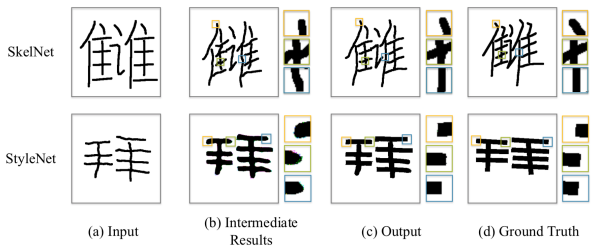
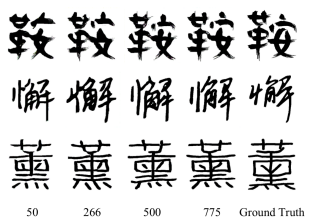


Figure 7: The effect of stacked architectures used in both SkelNet and StyleNet.



N_{train}	L1 loss	IOU
50	0.1543	0.3121
100	0.1488	0.3346
200	0.1458	0.3466
266	0.1442	0.3570
300	0.1404	0.3632
400	0.1374	0.3727
500	0.1353	0.3858
600	0.1335	0.3914
775	**0.1285**	**0.4163**

Figure 8: The effect of training set size.

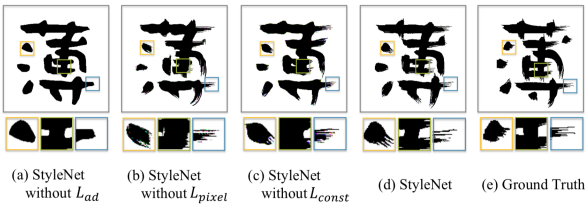


Figure 9: The ablation study of loss functions.

stroke category prior enables the network to better learn the correlations between strokes in the same category so as to further smooth the writing trajectories.

Stacked Network Design. In our model, we utilize the stacked networks to modify the small displacements and further refine the details in both SkelNet and StyleNet. Figure 7 shows the intermediate results in seperate networks. In the SkelNet, the second-stage network effectively reduces distortions and jitters on the writing trajectory and makes the entire skeleton more smooth. In the StyleNet, the stacked architecture enhances the quality of generated images and enriches the contour details in the character images while alleviating noises and distortions.

Effect of Training Set Size. To investigate the relationship of the quality of generated characters with the size of training set, we evaluate the performance of our method with a series of training sets with various sizes that contain 50, 100, 200, 266, 300, 400, 500, 600, 775 glyphs, respectively. The dataset with 266 and 775 images are the MinSet and OptSet, respectively, introduced in (Lian, Zhao, and Xiao 2016). Images in the other input sets are randomly chosen from the OptSet (775 glyphs). As shown in Figure 8, even with only 50 input samples, our model could get impressive results. As we expected, along with the increasing size of training set, the quality of generated results gets improved accordingly.

Effect of Loss functions. We also conduct an ablation study investigating the effect of components (L_{ad} , L_{pixel} , L_{const}) in the loss function of StyleNet (see Figure 9). The adversarial loss indeed encourages more stylish details in generated glyphs while the pixel space loss guides the network to obtain clearer results, and is essential to the convergence of the entire model. Besides, the consistency loss guarantees the structure correctness of glyphs and avoids unreasonable strokes. Experimental results show that the combination of these components in the loss function could take advantage of their respective benefits and thus result in the best performance.

Conclusion

This paper presented a novel structure-guided Chinese font generation system, SCFont, via deep stacked networks. Our method incorporates the benefits of deep neural networks with prior knowledge of Chinese characters to ensure structure correctness and style consistency simultaneously. Specifically, writing trajectories in the reference style are transformed to a specific style by the skeleton transformation network, and the synthesized writing trajectories are

rendered with stylish details on the contours by a novel generative model. Experimental results demonstrated our advantages in both visual perceptions and quantitative evaluations compared to other existing methods.

Acknowledgements

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

References

Azadi, S.; Fisher, M.; Kim, V.; Wang, Z.; Shechtman, E.; and Darrell, T. 2018. Multi-content gan for few-shot font style transfer. In *CVPR*, volume 11, 13.

Chang, B.; Zhang, Q.; Pan, S.; and Meng, L. 2018. Generating handwritten chinese characters using cyclegan. In *WACV*, 199–207.

Chang, J.; Gu, Y.; and Zhang, Y. 2017. Chinese typeface transformation with hierarchical adversarial network. *arXiv preprint arXiv:1711.06448*.

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G. 2017. Stylebank: An explicit representation for neural image style transfer. In *CVPR*.

Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; and Brox, T. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2758–2766.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; and Warde-Farley, D. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; and Brox, T. 2017. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, volume 2, 6.

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*, 5967–5976.

Jiang, Y.; Lian, Z.; Tang, Y.; and Xiao, J. 2017. Dcfont: an end-to-end deep chinese font generation system. In *SIGGRAPH ASIA 2017 TB*, 22.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 694–711.

Kim, T.; Cha, M.; Kim, H.; Lee, J. K.; and Kim, J. 2017. Learning to discover cross-domain relations with generative adversarial networks. In *ICML*, 1857–1865.

Lian, Z.; Zhao, B.; and Xiao, J. 2016. Automatic generation of large-scale handwriting fonts via style learning. In *SIGGRAPH ASIA 2016 TB*, 12.

Lin, J.-W.; Hong, C.-Y.; Chang, R.-I.; Wang, Y.-C.; Lin, S.-Y.; and Ho, J.-M. 2015. Complete font generation of chinese characters in personal handwriting style. In *IPCCC*, 1–5.

Lyu, P.; Bai, X.; Yao, C.; Zhu, Z.; Huang, T.; and Liu, W. 2017. Auto-encoder guided gan for chinese calligraphy synthesis. In *ICDAR*, volume 1, 1095–1100.

Mirza, M., and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Odena, A.; Olah, C.; and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. In *ICML*, 2642–2651.

Park, E.; Yang, J.; Yumer, E.; Ceylan, D.; and Berg, A. C. 2017. Transformation-grounded image generation network for novel 3d view synthesis. In *CVPR*, 702–711.

Radford, A.; Metz, L.; and Chintala, S. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Sun, D.; Ren, T.; Li, C.; Zhu, J.; and Su, H. 2017. Learning to write stylized chinese characters by reading a handful of examples. *arXiv preprint arXiv:1712.06424*.

Tian, Y. 2016. Rewrite: Neural style transfer for chinese fonts. `https://github.com/kaonashi-tyc/Rewrite`.

Tian, Y. 2017. zi2zi: Master chinese calligraphy with conditional adversarial networks. `https://github.com/kaonashi-tyc/zi2zi`.

Wang, X.; Yu, K.; Dong, C.; and Change Loy, C. 2018. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *CVPR*.

Xu, S.; Lau, F. C.; Cheung, W. K.; and Pan, Y. 2005. Automatic generation of artistic chinese calligraphy. *IEEE Intelligent Systems* 20(3):32–39.

Xu, S.; Jin, T.; Jiang, H.; and Lau, F. C. 2009. Automatic generation of personal chinese handwriting by capturing the characteristics of personal handwriting. In *IAAI*.

Yi, Z.; Zhang, H. R.; Tan, P.; and Gong, M. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2868–2876.

Zhang, Y.; Zhang, Y.; and Cai, W. 2018. Separating style and content for generalized style transfer. In *CVPR*, volume 1.

Zhou, T.; Tulsiani, S.; Sun, W.; Malik, J.; and Efros, A. A. 2016. View synthesis by appearance flow. In *ECCV*, 286–301.

Zhou, B.; Wang, W.; and Chen, Z. 2011. Easy generation of personal chinese handwritten fonts. In *ICME*, 1–6.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*.

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E. 2017b. Toward multimodal image-to-image translation. In *NIPS*, 465–476.

Zong, A., and Zhu, Y. 2014. Strokebank: Automating personalized chinese handwriting generation. In *AAAI*, 3024–3030.