ABSTRACT

Previous works on font generation mainly focus on the standard print fonts where character’s shape is stable and strokes are clearly separated. There is rare research on brush handwriting font generation, which involves holistic structure changes and complex strokes transfer. To address this issue, we propose a novel GAN-based image translation model by integrating the skeleton information. We first extract the skeleton from training images, then design an image encoder and a skeleton encoder to extract corresponding features. A self-attentive refined attention module is devised to guide the model to learn distinctive features between different domains. A skeleton discriminator is involved to first synthesize the skeleton image from the generated image with a pre-trained generator, then to judge its realness to the target one. We also contribute a large-scale brush handwriting font image dataset with six styles and 15,000 high-resolution images. Both quantitative and qualitative experimental results demonstrate the competitiveness of our proposed model.

Index Terms— Font Generation, Generative Adversarial Network, Brush Handwriting Font Dataset

1. INTRODUCTION

During thousands-year history of Chinese calligraphy, many styles of writing or chirography came into being. The chirography style can be defined as the skeleton structure and stroke style. The skeleton contains the basic information of character, such as the composition and position of strokes, writing direction, etc., while the stroke style means the deformation of the skeleton, such as the thickness, shape, writing strength, etc. Intuitively, it’s essential to ensure structure correct and keep style consistent when generating brush handwriting font automatically.

Recent works [1–5] formulate the Chinese font generation as an image style transfer problem, where characters in the reference style are transferred to a specific style. As it’s time-consuming and labor-intensive to create a handwriting Chinese font library, most of the researches focus on Standard Print Font Generation (SPFG), however, there is rare research on Brush Handwriting Font Generation (BHFG). As shown in

Fig. 1. Illustration of SPFG and BHFG.

1 Corresponding author, email: chenmeng20@jd.com.
The contributions of this paper can be summarized as follows: 1) We propose a novel end-to-end GAN-based model (SE-GAN) for BHFG. Besides, a novel Self-attentive Refined Attention Module (SAttRAM) is devised and applied in the generator to extract and fuse the features from source image and skeleton image effectively. 2) Extensive experiments were conducted on six different styles of Chinese font generation tasks. Both automatic and human evaluations demonstrate the efficacy of SE-GAN compared to strong baselines. 3) To facilitate future research on BHFG, we also contribute a large-scale font image dataset and will release it soon.

2. RELATED WORK

Image-to-Image Translation. Since the propose of Generative Adversarial Network (GAN) [6], many works have been proposed for image-to-image translation tasks. Pix2pix [7] is a conditional GAN based on supervised learning with paired data. Then unsupervised image translation models such as CycleGAN [8] is proposed to use the cycle consistency to improve the training stability. Apart from one-to-one domain image translation, StarGAN [9] introduces a domain classifier or a shared multi-domain embedding to achieve many-to-many domain translation with a single model. Furthermore, U-GAT-IT [10] proves the effectiveness of attention module in image translation task, which can guide the model to focus on more important regions between different domains. However, these works are formulated as pixel-to-pixel translation where the source and target images contain little deformations, which cannot be directly applied to font generation tasks with huge holistic changes.

Chinese Font Generation. Most previous works formulate character image generation as image translation tasks [1, 11–13]. Zi2zi [1] generates stylized Chinese font generation with paired data. However, it requires large-scale font-pair corpus for pre-training and fine-tuning. To overcome the challenge of insufficient data, [14] apply cycle consistency to generate fonts from unpaired data, and [11, 12] separate content and style as two irrelevant domains and learn the latent style from font images. Recently, [15] utilize StarGAN [9] equipped with diversity regularizer to achieve multi-style Chinese font generation. Some other works [5, 16–18] try to integrate more domain knowledge of character into multi-stage model for character generation. Most of these works concentrate on the SPF, and the images from source and target domain usually share very similar styles or shapes. In this paper, we focus on the BHFG task, which contains large stroke style difference and layout changes. We propose a one-stage model instead of multi-stage model, thus all modules are trained jointly and the generation process is more efficient.

3. APPROACH

3.1. Overall Framework

Figure 2 shows our framework with two encoders: the image encoder $E_i$ and the skeleton encoder $E_s$, which are composed of four residual blocks. The $E_i$ is employed to extract character image features from $X_i$, including content and style information from source domain whereas the skeleton encoder $E_s$ aims to preserve the structure feature from $X_s$. To enhance the feature extraction and fusion of two different features, a novel self-attentive refined attention module (SAttRAM) with two variants, SRAM and CRAM, are stacked sequentially to extract the skeleton enhanced image representations. Then, the generator takes the refined image representations which include both content and style information as input to generate the target style image. Following the adversarial training strategy in GANs, we employ two discriminators. The first $D_i$ is used to discriminate the target image and generated image. The second $D_s$ tries to detect whether the skeleton of generated image is coherent to the skeleton image extracted from the target domain character image. For skeleton extraction, inspired by [19], we adopt an simple but effective skeletonization algorithm to extract the skeleton image by eroding and dilating the binarized character image iteratively.
3.2. Self-attentive Refined Attention Module

The previous works [10, 20, 21] demonstrate the effectiveness of class activation map (CAM) in localizing the important regions of input images in both image generation and classification tasks. Inspired by this, CAM can be used to extract style-discriminative attention heatmaps in the font generation task. To obtain style-discriminative features $M(x)$, the decoded feature maps $F(x) \in \mathbb{R}^{W \times H \times C}$ of source font $x$ are first fed into the classifier, a fully connected (FC) layer with weights $\Omega \in \mathbb{R}^C$, for domain classification. Then CAM computes the attention heatmaps by linearly weighted summation of all channels:

$$M(x) = \sum_{c=k} C \Omega_k F(x)$$  \hspace{1cm} (1)

where $M(x) \in \mathbb{R}^{W \times H \times C}$ indicates the attention heatmap at spatial location $H, W$, $\Omega_k$ represents the weight for channel $k$ in feature maps, and $F(x) \in \mathbb{R}^{W \times H}$ represents the feature map of channel $k$ from the last convolutional layer at spatial location $HW$.

However, the CAM lacks the spatial attention and usually leads to over-activation issues for feature capturing [22]. Moreover, it’s difficult to integrate multi-modal features with a CAM module. To refine the style-discriminative feature and integrate multi-modal features, we propose a self-attentive refined attention module (SAttRAM) as shown in Fig 3. We bring self attention as an efficient module to refine the pixel-wise attention heatmaps by capturing spatial dependency. Hence, the refined feature maps can be defined as follows:

$$\hat{M}(x) = f(F(x), F(x))g(M(x)) + M(x) \hspace{1cm} (2)$$

$$f(F(x), F(x)) = \sigma(\theta(F(x))^T \phi(F(x))) \hspace{1cm} (3)$$

Here $f$ is a pairwise embedding function that computes dot-product pixel affinity as self-refined attention weight normalized by softmax function $\sigma$ in an embedding space. The embedding functions $\theta, \phi$ are implemented by individual $1 \times 1$ convolution layers, where $\theta(F(x)) \in \mathbb{R}^{C1 \times WH}$ and $\phi(F(x)) \in \mathbb{R}^{C1 \times WH}$. The function $g$ reshapes the input feature $F(x)$ to $g(F(x)) \in \mathbb{R}^{WH}$, all of which are aggregated with this similarity weights given by function $f(F(x), F(x)) \in \mathbb{R}^{C \times WH}$. This self-refined attention weight is normalised by softmax function $\sigma$ and the output is $\hat{M}(x) \in \mathbb{R}^{WH \times C}$.

To handle the multimodal input features for image generation, we build two variant attention units on top of the refined attention feature maps, namely the self-refined attention module (SRAM) unit and the cross-refined attention module (CRAM) unit. SRAM takes a group of images features $x_i = E_i(x)$ as input to obtain attended features, since image features are basic information in image translation. CRAM catches intra-modal interactions between image features $x_i = E_i(X_i)$ and skeleton features $x_s = E_s(X_s)$, which can further refine the feature maps extracted from character images.

$$\hat{M}(x_i, x_s) = f(F(x_i), F(x_s))g(M(x_i)) + M(x_i) \hspace{1cm} (4)$$

In Equation 4, the character encoding feature is embedded into the residual space by function $g$. Whereas $f$ represents the pixel-level feature aggregation between image feature $x_i$ and skeleton feature $x_s$.

![Fig. 3. The proposed SAttRAM. There are two variant modules: SRAM takes one group of input features $X_i$. CRAM takes two groups of input features $X_i$ and $X_s$.](image)

3.3. Discriminators

We design two discriminators for learning target font style. Following the traditional setting, the first discriminator $D_i$ is employed to calculate how similar the generated image is to the font image of target domain, and the encoder for $D_i$ also exploits the refined attention feature maps as mentioned in previous section. Under the assumption that the generated character image should have similar skeleton to its target font image, we also design an extra skeleton discriminator $D_s$. Considering that the skeletonization is a non-differential function, we first apply a pre-trained skeleton generator (skeletonGAN) to generate the corresponding skeleton from a given character image. Then the skeleton discriminator $D_s$ is applied to distinguish the skeleton of generated image from that of ground truth image. For skeletonGAN, we train a simple CycleGAN model [8] without SAttRAM module, as skeletonization task appears to be much easier than the character generation task. With the help of skeleton discriminator $D_s$, the model is encouraged to preserve the content and structure of character during training, meanwhile, $D_s$ is also served as a regularizer for SE-GAN to prevent model from overfitting.

3.4. Loss Design

The loss function of SE-GAN contains content loss, adversarial loss, cycle-consistency loss and classification loss. The content loss consists $L_{con}$ of two parts: the pixel loss $L_{pix}$ which forces the generated image $G_p(X_i, X_s)$ to be similar to the target image $Y_t$, and the skeleton consistency
loss $L_{sc}$ which ensures the skeletons consistency between $SG(G_F(X_i, X_s))$ and $SG(Y_i)$.

$$L_{pix}(G_F) = \mathbb{E}_X[| |G_F(X_i, X_s) - Y_i||_1]$$  \hspace{1cm} (5)

$$L_{ac}(G_F, SG) = \mathbb{E}_X[| |SG(G_F(X_i, X_s)) - SG(Y_i)||_1|]$$  \hspace{1cm} (6)

where the $SG$ represents the pre-trained skeletonGAN.

The cycle-consistency loss $L_{cycle}$ is identical to that used in [8], which guarantees that the cycle transformation is able to bring the image back to the original state. In order to distinguish that the image $X_i$ belongs to source or target style $Y_i$, and promote the style transformation in the refined attention module, we use an auxiliary loss $L_{cls}$ following [10].

Besides, the adversarial loss $L_{adv}$ combines discriminative loss $L_{D_i}$ and skeleton-level discriminative loss $L_{D_s}$, and these two losses aim to catch different properties of the desired generated image $X_i$.

$$L_{D_i}(G_F, D_i) = \mathbb{E}_Y[logD_i(G_F(X_i, X_s))] + \mathbb{E}_X[log(1 - D_i(G_F(X_i, X_s)))]$$  \hspace{1cm} (7)

$$L_{D_s}(G_F, D_s) = \mathbb{E}_Y[logD_s(Y_i)] + \mathbb{E}_X[log(1 - D_s(SG(G_F(X_i, X_s))))]$$  \hspace{1cm} (8)

where $D_i$, $D_s$ are pixel-level discriminator and skeleton discriminator respectively.

Finally, we jointly train the generators, discriminators, and classifiers by using the full objective as follows:

$$\min_{G_F, G_B, RA, D_1, D_2} \max_{\lambda_1, \lambda_2, \lambda_3, \lambda_4} L_{adv} + \lambda_2 L_{cycle} + \lambda_3 L_{cls} + \lambda_4 L_{con}$$  \hspace{1cm} (9)

where $\lambda_i$ controls the weights of different losses. Here, we omit the corresponding backward loss functions for simplicity because they are also defined in the same way.

### 4. EXPERIMENTS

#### 4.1. Experimental Setup

As the deficiency of public brush handwriting font generation dataset, we collect a large-scale image dataset for experiments with six different styles. The statistics of each style is presented in Table 1. The total number of images is 15,799, and the size of each subset ranges from 1,419 to 3,958. During experiments, we split each subset into train/dev/test set by ratio of 8:1:1. We choose the standard print font Liukai\(^1\) as the source domain, and take each of the six styles as the target domain respectively. We adopt content accuracy [3] and Fréchet Inception Distance (FID) [23] as evaluation metrics. For baselines, we compare our model with four representative font generation models, including zi2zi [1], CycleGAN [14], StarGAN [15], and DF FONT [12].

#### 4.2. Experimental Results

**Quantitative analysis.** We calculate both the Top 1 content accuracy (Acc) and FID score for all baselines and our proposed model on six chirography styles. Table 1 illustrates that our proposed model SE-GAN achieves the best accuracy and the lowest FID score for nearly all six styles. Compared with supervised models like zi2zi, our model can still generate high-quality images. Compared with unsupervised models CycleGAN and StarGAN, which have unstable performances among different styles, SE-GAN still has very competitive performance and significant improvements. For DF-Font, we notice that it is inclined to generate similar font images to the source images, however, the distinctive styles are not well learnt. Although the content accuracy of DF-Font is high, the FID score is the worst.

**Qualitative analysis.** We show some generated examples of all models in Fig. 4 for case study. Generally, the font images from SE-GAN are much easier to recognize and the chirography styles are more consistent with the original styles mentioned. For other baselines, there exists obvious flaws in the generated font images. For zi2zi, there are missing strokes for Style 2 and Style 6, and the structures are tied to each other for Style 1 and Style 5. For CycleGAN, the structures are wrong.

\(^1\)https://www.foundertype.com/index.php/FontInfo/index/id/197

---

### Table 1. Evaluation on font image generation. Acc represents content accuracy of character recognition.

<table>
<thead>
<tr>
<th>Models</th>
<th>Style 1</th>
<th>Style 2</th>
<th>Style 3</th>
<th>Style 4</th>
<th>Style 5</th>
<th>Style 6</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>FID</td>
<td>Acc</td>
<td>FID</td>
<td>Acc</td>
<td>FID</td>
</tr>
<tr>
<td>zi2zi [1]</td>
<td>0.201</td>
<td>93.32</td>
<td>0.291</td>
<td>84.01</td>
<td>0.443</td>
<td>70.79</td>
</tr>
<tr>
<td>CycleGAN [14]</td>
<td>0.348</td>
<td>77.16</td>
<td>0.277</td>
<td>76.05</td>
<td>0.432</td>
<td>76.51</td>
</tr>
<tr>
<td>StarGAN [15]</td>
<td>0.257</td>
<td>83.68</td>
<td>0.222</td>
<td>93.74</td>
<td>0.437</td>
<td>67.88</td>
</tr>
<tr>
<td>DF-Font [12]</td>
<td>0.421</td>
<td>103.82</td>
<td>0.317</td>
<td>96.85</td>
<td>0.414</td>
<td>80.75</td>
</tr>
<tr>
<td>SE-GAN</td>
<td>0.434</td>
<td>62.58</td>
<td>0.486</td>
<td>60.43</td>
<td>0.513</td>
<td>50.40</td>
</tr>
<tr>
<td>SE-GAN w/o $E_s$</td>
<td>0.406</td>
<td>76.51</td>
<td>0.369</td>
<td>80.15</td>
<td>0.475</td>
<td>66.87</td>
</tr>
<tr>
<td>SE-GAN w/o $D_s$</td>
<td>0.427</td>
<td>72.14</td>
<td>0.388</td>
<td>74.86</td>
<td>0.491</td>
<td>57.64</td>
</tr>
<tr>
<td>Human</td>
<td>0.465</td>
<td>-</td>
<td>0.521</td>
<td>-</td>
<td>0.604</td>
<td>-</td>
</tr>
</tbody>
</table>

---

### Table 2. Statistics of six styles in our dataset.

<table>
<thead>
<tr>
<th>Number</th>
<th>Style 1</th>
<th>Style 2</th>
<th>Style 3</th>
<th>Style 4</th>
<th>Style 5</th>
<th>Style 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1885</td>
<td>3008</td>
<td>3958</td>
<td>1419</td>
<td>2356</td>
<td>3175</td>
<td></td>
</tr>
</tbody>
</table>

---

\(^1\)https://www.foundertype.com/index.php/FontInfo/index/id/197
and the characters are hard to recognize for Style 1 and 3. For StarGAN, there are some extra and erroneous strokes in Style 2 and 3, which damage the overall appearance of characters. All above examples demonstrate that the skeleton information can facilitate the font generation.

**Ablation Study.** We conduct ablation experiments by removing the skeleton input $E_s$ from generator and removing the skeleton discriminator $D_s$ separately. As shown in Table 1, after removing $E_s$, the accuracy drops evidently compared with SE-GAN, which indicates the skeleton information is helpful for learning the structure of characters. However, it still outperforms CycleGAN for most of the styles, proving the effectiveness of SAttRAM. Besides, removing $D_s$ also degrades the model performance over all styles, indicating the necessity of skeleton discriminator. Fig. 5 (a) illustrates the generated images of different ablated models. Due to the lack of skeleton information, some strokes in the generated characters are either missing or over exaggerated. When $D_s$ is removed, some components are mixed together, hurting the readability of the generated images. Fig. 5 (b) compares the heated attention maps of CAM, SRAM and CRAM. It’s observed that, compared with CAM, SRAM and CRAM have fewer over-activations and more complete activation coverage. Besides, the font shape learnt by CRAM is more accurate than SRAM and closer to the ground-truth font image, which verified the contribution of skeleton.

**User Study.** We conduct two kinds of user study (we skip evaluating DF-Font considering its bad performance in Table 1). The first is to evaluate preference score for the generated font images of different models. The second is to pick out the more visually pleasing font image by mixing the generated images with human-written font images. Totally sixty students majored in fine arts with more than 3-year experience of calligraphy writing were invited to finish the human evaluation. Table 3 reveals our model obtains the highest user preference score and winning rate from the human experts.

<table>
<thead>
<tr>
<th>Models</th>
<th>Pref score</th>
<th>Win rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>zi2zi</td>
<td>2.3</td>
<td>14%</td>
</tr>
<tr>
<td>CycleGAN</td>
<td>3.1</td>
<td>18%</td>
</tr>
<tr>
<td>StarGAN</td>
<td>2.9</td>
<td>35%</td>
</tr>
<tr>
<td>SE-GAN</td>
<td>4.2</td>
<td>57%</td>
</tr>
</tbody>
</table>

5. **CONCLUSION AND FUTURE WORK**

In this paper, we propose SE-GAN, a novel end-to-end framework for brush handwriting font generation. As the task involves holistic structure changes and complex stroke transfer, we propose to integrate the skeleton information for character image generation. We design two encoders to extract the character and skeleton features respectively. To efficiently fuse the information from both sides, a novel self-attentive attention module is devised in the generator. Besides, we also employ a skeleton discriminator to ensure the content consistency between generated and target images. The experiments demonstrate the advantages of our proposed model over several strong baseline methods. In the future, we will explore the pre-trained image translation models to facilitate this task.
6. REFERENCES


