Few-shot Font Style Transfer between Different Languages

Chenhao Li, Yuta Taniguchi, Min Lu, Shin'ichi Konomi
HDI Lab, Kyushu University
{li.chenhao.995@s, taniguchi@ait, lu@artsci, konomi@artsci}.kyushu-u.ac.jp

Abstract

In this paper, we propose a novel model FTransGAN that can transfer font styles between different languages by observing only a few samples. The automatic generation of a new font library is a challenging task and has been attracting many researchers' interests. Most previous works addressed this problem by transferring the style of the given subset to the content of unseen ones. Nevertheless, they only focused on the font style transfer in the same language. In many tasks, we need to learn the font information from one language and then apply it to other languages. It's difficult for the existing methods to do such tasks. To solve this problem, we specifically design our network into a multi-level attention form to capture both local and global features of the style images. To verify the generative ability of our model, we construct an experimental font dataset which includes 847 fonts, each of them containing English and Chinese characters with the same style. Experimental results show that compared with the state-of-the-art models, our model generates 80.3% of all user preferred images.

1. Introduction

Fonts are significant visual design that can deliver information like whether the current content is serious or casual, some artistic fonts can even create a scary or playful atmosphere. However, designing a new font is a time-consuming job because many factors like stroke, decoration, effect should be considered. Besides, all characters in the same font must be designed in a coherent style and suitable size. Many large font libraries may contain thousands of characters from multiple languages (*e.g.*, Microsoft YaHei contains more than 20,000 characters composed by Chinese, Japanese, Korean, Latin, Greek...). Artists spend a long time maintaining a coherent style among these characters to make them visually compatible. This labor-intensive process may cause many problems, especially in many scenarios, designers will only design glyph images in one language, it may take a lot of time to extend the style to other languages later.

Figure 1. Several application examples. The English letters on the red background are style images, and the Chinese characters on the green background are content images. The rest are the images generated by our proposed FTransGAN. It extracts font information from a few observed English letters and applies extracted style to the given Chinese characters automatically. The Chinese in the figure means "WACV".

With the rise of deep neural networks, Automatic font generation without human intervention becomes possible. They considered the transfer of shape and texture at the same time in an end-to-end manner. Early methods [4, 17, 23, 29] have been proposed to generate the entire font library by observing a subset of it. These methods consider font style transfer as an image-to-image translation [16] or a cGAN [9, 24] task. While these image-to-image translation based methods have shown the remarkable generative ability, they still have several significant disadvantages. First, the training process is divided into two stages. Usually, they pre-train models on a large dataset, then to some specific task, these models must be fine-tuned, which makes them less practical when computing resources are limited. Second, during the fine-tuning stage, hundreds of training samples are required. Creating these training samples is another labor-intensive job. Recently, several few-shot learning methods have been proposed [1, 3, 7, 21, 27, 35, 37], their models can build a high-quality font library by observing only a few samples.

Nevertheless, all the above methods only transfer style in the same language. In the real world, artists usually design a font for only one language, and extending the style to other languages is a time-consuming work. The title of movie

poster is a good example, an international movie requires posters in many languages with the same style, but it takes artists the same time and energy to design each of them. Thus, a model that can learn style information from another language is necessary. However, the characters of different languages may be dramatically dissimilar. More specifically, some components of Chinese characters are complicated and do not appear in English letters. Furthermore, unlike artistic style transfer tasks [8, 18] that only need to consider global style information, the style of the font is often composed of local (*e.g*., decoration, stroke, thickness) and global features (*e.g*., shape, effect). Therefore, learning style from characters of another language is difficult and requires a model that can learn some essential characteristics.

To address this problem, we propose a novel model, **FTransGAN** (Font Translator GAN) that can generate a high-quality font library by observing only a few samples from other languages. Unlike existing methods [7, 37] that rely heavily on fine-tuning, our model only needs a forward pass to generate images when testing. This means that it can be applied to some real-time systems. We use two encoders to extract the style and content representation respectively. Then we simply concatenate and input them into a decoder. Besides, two discriminators are designed to check the matching degree from the style and content perspective separately. We specifically designed the style encoder by using two modules: Context-aware Attention Network and Layer Attention Network, they work together to capture both the local and global style features. We believe this multi-level attention design can make our model more flexible when dealing with an arbitrary style input. Experimental results on the collected multi-language dataset show a high visual quality for both handwritten fonts and printing fonts. We illustrate some application examples in Figure 1. The main contributions of this work are summarized as follows:

- We develop a novel model, FTransGAN, and first apply an end-to-end solution to cross-language font style transfer.
- We introduce two novel modules, Context-aware Attention Network and Layer Attention Network to capture both local and global style features simultaneously.
- The architecture of the proposed model allows an arbitrary number of input style images, so it can transfer styles between any languages without being limited by the number of characters.
- We construct a new multi-language glyph image dataset which consists of 847 fonts, each with 52 English letters and more than 1000 Chinese characters.

Our FTransGAN code and dataset are available at https://github.com/ligoudaner377/font_translator_GAN.

2. Related Work

2.1. Style Transfer

Style transfer methods [8, 18] apply the style from one image to another one. The input usually consists of a content image and a style image. Then, they obtain the output by optimizing content loss and style loss. Content loss is the distance of activations at each location of two feature maps. Style loss can be calculated by comparing the summary statistics of each layer. Huang and Belongie proposed the adaptive instance normalization [14] that can adapt to arbitrary new styles. Recently Gu *et al*. [10] introduced an arbitrary style transfer method by reshuffling deep features of the style image. However, these methods are mainly designed for artworks and they usually define style as a set of colors and textures. As mentioned before, the style of font is more abstract and composed of local and global features. Hence, these methods can't be applied to the font style transfer directly.

2.2. Image-to-Image Translation

Pix2Pix [16] and CycleGAN [38] proposed a general image-to-image translation frame that aims to learn the mapping between two domains. The data of the former is paired, while the latter is unpaired. Liu and Tuzel proposed the coupled GAN (CoGAN) [22] which learns a joint distribution of two domains by weight sharing. However, these models can only translate the image between two domains. Moreover, these methods usually require a large number of training data, which is less practical. starGAN [5] solved the first problem by adding domain information to the generator. But the generative ability of their model is limited to several domains, in other words, it cannot generate images of unknown domains. Recently, FUNIT [21] proposed a few-shot unsupervised image generation method. They used an encoder to extract domain information from several images instead of adding it to the generator directly.

2.3. Attention Mechanism

Attention mechanism [2] was first proposed in NLP field to alleviate the damage caused by a fixed-length vector in the encoder-decoder architecture. This mechanism allows the machine to focus on certain words. Xu *et al*. [31] applied an attention network to the visual field, tackling the image captioning problem. Yu *et al*. [33] proposed a multi-level attention network to obtain both spatial information and semantic information from one image. More recently, Vaswani *et al*. [30] made many breakthroughs in the NLP field by leveraging the self-attention module. SAGAN [34] successfully applied the self-attention layer to the image generation task later.

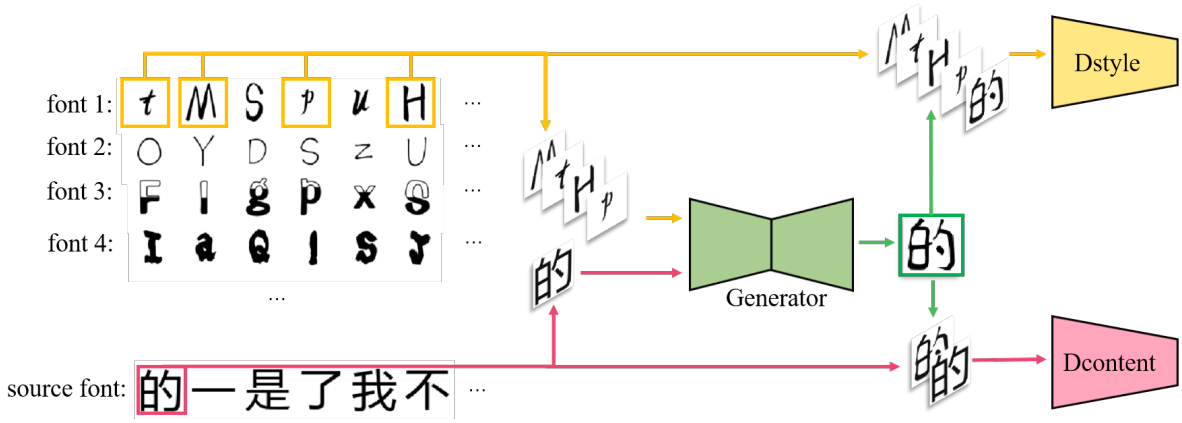

Figure 2. Overview of our proposed FTransGAN. Each time we randomly select K style images from the dataset as the style input, and one content image from the source font as the content input, content images and style images are from different languages, then input them into the generator. Two discriminators check the matching degree of the style perspective and content perspective, respectively.

2.4. Font Generation

Font generation can be considered as a special case of image style transfer. Traditional methods [28] mainly relied on the shape modeling of outlines. Zhou *et al*. [36] proposed a Chinese character radical composition model to generate handwritten fonts.

More recently, Deep Neural Networks (DNNs) [19] based methods [4, 17, 23, 29] have achieved significant progress. They directly adopted the image-to-image translation architecture. Where the condition is one character from a source font A, and the output should match the corresponding character in target font B. Lian *et al*. [20] used a DNN to extract the stroke from the user's handwritten font and apply it to the left characters. Also, several models [1, 3, 7, 27, 35, 37] have been reported for few-shot font generation. Their models can generate an entire font library by using only a few samples, MC-GAN [1] proposed the first end-to-end solution to synthesizing artistic font. However, the number of input and output images is fixed (26 English capital letters), which means that their method cannot handle a large font library (*e.g*., Chinese) due to the limitation of this architecture. AGIS-Net [7] and EMD [35] can also be considered as an image-to-image translation task. The difference is these models take 2 conditions, which are style and content images. The output glyph should be the combination of two sets of conditions. Recently, Cha *et al*. [3] achieved significant progress by utilizing the compositionality of compositional scripts. Zhu *et al*. proposed the Deep Feature Similarity architecture [37] by leveraging the feature similarity between the input content images and style images to generate the target image. However, they can't perform well to the constructed multi-language dataset, as observed in our experimental results.

3. Method Description

Like other few-shot font generation methods, our goal is to generate glyph images by taking two conditions which are style and content images. We regard the glyph image generation as the process of solving a conditional probability $p(x|s, c)$. Where x is the target image, c is the content image in a standard style (*e.g*., Microsoft YaHei) and s is the set of style images $\{s^1, s^2, \ldots, s^K\}$, they share the same style but different content. The content images are only used for indexing the category of characters. Previous works [7, 37] have verified that the style or font of the content images will not significantly change the final results. The reason why there is only one content image, but multiple style images is because usually humans only need one image to identify the content, but we need several images to extract a common style. Considering our task is to learn font style information between different languages, the content and style images should come from different languages. For example, if the content image is a Chinese character, the style images should be composed of English letters, vice versa. To train our FTransGAN, we randomly select a set of style images and one content image from our dataset and input them to the model. During testing, we provide the model a few images with a novel style or content. We consider that the size of all input and output images is 64×64 in gray-scale.

As shown in Figure 2 and Figure 3, our model has a Generator G and two discriminators: Content Discriminator D_{content}, and Style Discriminator D_{style}. Two discriminators follow the design of PatchGAN [16] to check the real and fake patches locally. The Generator G consists of a style encoder, a content encoder, and a decoder. Two encoders extract the style representation and content representation respectively, then the decoder will take the extracted infor-

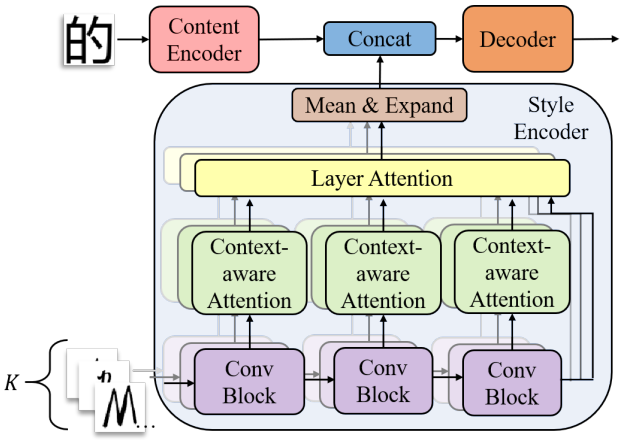

Figure 3. Overview of the proposed Generator G.

mation and generate the target image. The content encoder is made of three convolutional layers with each follow by BatchNorm [15] and ReLU, and the decoder consists of six ResNet [11] blocks and three up-convolutional layers likewise followed by BatchNorm and ReLU. Inspired by Yang *et al*. [32] and Yu *et al*. [33], we specifically design the style encoder into a multi-level attention form by using two different attention modules, Context-aware Attention Network and Layer Attention Network to capture both local and global style features. It first maps each style image $\{s^1, s^2, \ldots, s^K\}$ to a feature vector and then computes the mean of them to get the final style feature vector z_s. More details are given in section 3.1 and 3.2.

3.1. Context-aware Attention Network

As shown in Figure 3, the style encoder has three parallel Context-aware Attention Blocks. They have 13×13, 21×21, 37×37 receptive fields respectively. So, the shallower layer can only see local features, while the deeper layer can see almost the entire image. Figure 4 shows the details of the Context-aware Attention Block. The input is a feature map with a size of C×H×W given by the last convolutional layer. Where C, H, W denote the number of channels, height, and width respectively. Here, we denote each region of the feature map as $\{v_r, r = 1, 2, \ldots, H \times W\}$. First, unlike previous works [32, 33] using an LSTM [13] or GRU [6] block to obtain context information recurrently, for better computing efficiency, we incorporate the context information into the feature map from each region by a Self-Attention [34] layer, which is given by:

$$h_r = SA(v_r), \tag{1}$$

where SA denotes the Self-Attention layer, the new features vectors h_r contain not only the information limited to their receptive fields but also the contextual information from other regions.

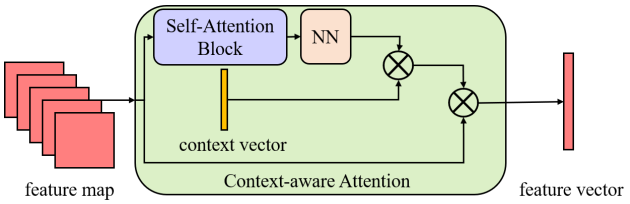

Figure 4. Architecture of the proposed Context-aware Attention Network.

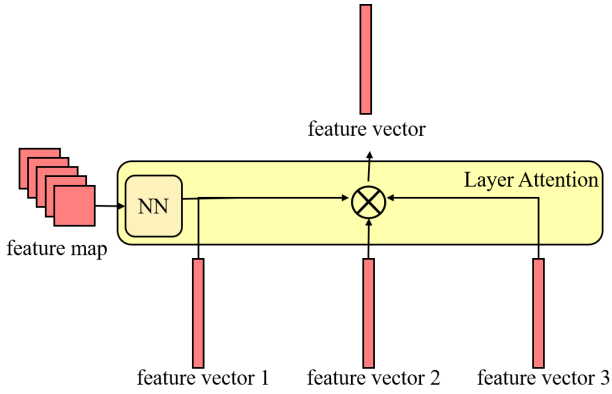

Figure 5. Architecture of the proposed Layer Attention Network.

Second, we introduce the attention mechanism to assign each region a score, because we believe that not all regions have the same contribution. Specifically,

$$u_r = \tanh(W_c h_r + b_c), \tag{2}$$

$$a_r = \mathrm{softmax}(u_r^T u_c), \tag{3}$$

$$f = \sum_{H \times W} a_r v_r. \tag{4}$$

That is, we input the contextual vector h_r into a one-layer neural network to get u_r as a latent representation of h_r. Then, a context vector u_c is employed to measure the importance of the current region, u_c is randomly initialized and jointly trained with the entire model. After that, we can obtain the normalized attention score by a softmax layer. Last, we compute a feature vector f as a weighted sum of each region v_r. Note that we have three parallel Context-aware Attention Networks, so finally we can obtain three feature vectors f_1, f_2, f_3.

3.2. Layer Attention Network

Given a style image, should the machine focus on the local or global feature? We believe this depends on the image itself. Based on this assumption, we design the Layer Attention Network.

As shown in Figure 5, it takes four inputs, they are feature map f_m, given by the last convolutional layer, and three

feature vectors f_1, f_2, f_3 given by Context-aware Attention Networks. We employ a one-layer neural network here to assign each feature vector a score. These scores explicitly indicate which feature level the model should focus on. Specifically,

$$w_1, w_2, w_3 = \text{softmax}(\tanh(W_l f_m + b_l)), \tag{5}$$

$$z = \sum_{i=1}^{3} w_i f_i, \tag{6}$$

where w_1,w_2,w_3 are three normalized scores given by a neural network, and z is the weighted sum of three feature vectors. Note that each time, the style encoder will accept K images, so the final latent code z_s is the mean of all vector z:

$$z_s = \frac{1}{K} \sum_{K} z^k. \tag{7}$$

Besides, the size of the content code is C×H×W, but the style code z_s is a C-dimensional vector. We copy z_s several times to match their size. After obtaining the content code z_c and expanded style code z_s, we simply concatenate and input them into the decoder. The decoder will generate images based on the style code z_s and the content code z_c.

3.3. End-to-End Training

As mentioned before, our model consists of two discriminators D_{content} and D_{style}. They have almost the same architecture that consists of several convolutional layers. D_{content} takes generated image and content image and checks whether they are the same character, while D_{style} takes generated image and style image and checks whether they are the same style or not. We directly concatenate these images in the channel dimension to input to the discriminators. The entire model is jointly trained in an end-to-end manner.

The objective function of our model consists of three terms: L_1 loss, style loss L_{style}, content loss L_{content},

$$L = \lambda_1 L_1 + \lambda_s L_{\text{style}} + \lambda_c L_{\text{content}}, \tag{8}$$

where λ_1, λ_s, λ_c are three weights for balancing these terms. For higher quality results and to stabilize GAN training, both L_{content} and L_{style} use hinge loss [25] functions:

$$\begin{aligned} L_{\text{content}} &= L_{\text{contentD}} + L_{\text{contentG}}, \\ L_{\text{contentG}} &= -E_{x,c\sim P(x,c)}[D_{\text{content}}(x, c)], \\ L_{\text{contentD}} &= -E_{\hat{x},c\sim P(\hat{x},c)}[min(0, D_{\text{content}}(\hat{x}, c) - 1)] \\ &\quad - E_{x,c\sim P(x,c)}[min(0, -D_{\text{content}}(x, c) - 1)], \end{aligned} \tag{9}$$

$$\begin{aligned} L_{\text{style}} &= L_{\text{styleD}} + L_{\text{styleG}}, \\ L_{\text{styleG}} &= -E_{x,s\sim P(x,s)}[D_{\text{style}}(x, s)], \\ L_{\text{styleD}} &= -E_{\hat{x},s\sim P(\hat{x},s)}[min(0, D_{\text{style}}(\hat{x}, s) - 1)] \\ &\quad - E_{x,s\sim P(x,s)}[min(0, -D_{\text{style}}(x, s) - 1)], \end{aligned} \tag{10}$$

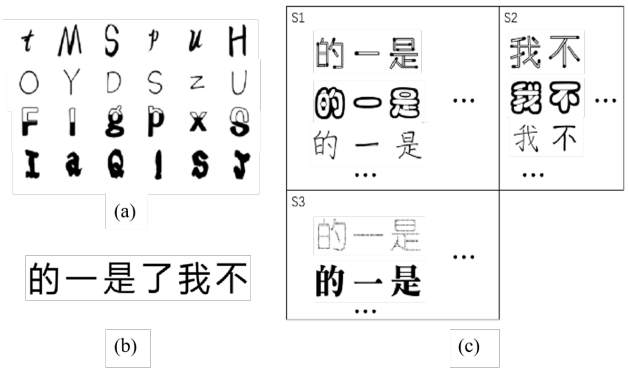

Figure 6. Examples of the font dataset that we constructed for our experiments. (a) Style images from several fonts, (b) Several content images from an ordinary font, (c) Ground truth images, S1 is used for training, S2 and S3 are used for testing unknown content and unknown style respectively.

where $\hat{x}$ is the ground truth images, and x is the generated images, c and s denote the content and style images respectively. To stabilize our training, we also use an L_1 loss in our objective function to compute the pixel-wise error between generated images and the ground truth images:

$$L_1 = E_{\hat{x},x\sim P(\hat{x},x)}[\|x - \hat{x}\|_1]. \tag{11}$$

4. Experiments

Next, we demonstrate the generative ability of the proposed FTransGAN compared with several state-of-the-art models. In the following experiments, we have set λ_1 = 100, λ_c = λ_s = 1 and K = 6. For both our model and competitors, we have trained them for 20 epochs with the learning rate lr = 0.0002.

4.1. Font Dataset

To evaluate the performance of our model, we construct a dataset including 847 gray-scale fonts (style input) each with approximately 1000 commonly used Chinese characters and 52 Latin letters in the same style. As mentioned before, we also use an ordinary font (*e.g.*, Microsoft YaHei) as our content input, it is only used for indexing the category of the character that we want to synthesize. We process the dataset by finding a bounding box around each glyph and resize it so that the larger dimension reaches 64 pixels, then pad to create 64×64 glyph images.

Next, we use English letters as the style input and Chinese characters as the content input. The model needs to transfer the style of English letters to Chinese characters. The reason for designing the experiment in this way is Chinese characters are usually more complicated than English letters and contain some features that English letters do not

	Pixel-level			Content-aware		Style-aware	
	↓MAE	↑SSIM	↑MS-SSIM	↑Accuracy(%)	↓mFID	↑Accuracy(%)	↓mFID
Evaluation on the unseen character images							
EMD [35]	**0.117**	0.497	0.467	81.2	116.9	24.4	597.1
DFS [37]	0.185	0.303	0.231	89.2	150.0	2.7	820.6
Ours	0.121	**0.501**	**0.493**	**97.0**	**49.8**	**58.1**	**308.9**
Evaluation on the unseen style images							
EMD [35]	**0.166**	**0.384**	**0.388**	85.5	184.4	4.4	623.2
DFS [37]	0.214	0.231	0.201	91.7	230.7	0.7	662.4
Ours	0.179	0.368	0.382	**99.8**	**97.8**	**11.7**	**418.8**

Table 1. Quantitative Evaluation on the proposed multi-language dataset. We evaluate the models on both the unseen characters and unseen styles. ↑ means larger numbers are better, ↓ means smaller numbers are better.

Figure 7. Visual comparison of our FTransGAN (**4th rows**) with EMD [35] (**2nd rows**) and DFS [37] (**3rd rows**), the observed style images are illustrated in the **1st rows** and the ground truth images are in the **5th rows**. To facilitate observation, we set the background of style images to red, ground truth images to blue, and the images generated by our model to purple. For each font, we randomly select 6 generated images as reference.

EMD [35]	10.3%
DFS [37]	9.4%
Ours	**80.3%**

Table 2. User preference data based on 390 responses.

have. This requires the model to be more flexible and robust to generate high-quality images. We randomly select 29 fonts and characters as unknown styles and contents and leave the rest part as training data. Therefore, the entire dataset is divided into three parts which are S1, images used for training, S2, images with seen styles during training but unknown contents used for testing, S3, images with known contents but unknown styles used for testing. Figure 6 shows a few examples of the font dataset as well as the partition rule. Before the following experiments, we made sure that there is no overlap between the training set and the testing set by computing the nearest neighbor for all images.

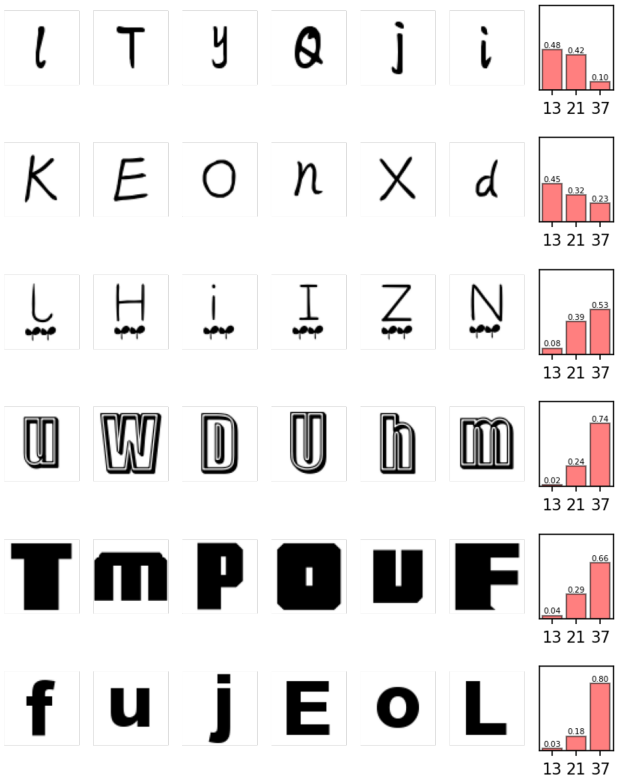

Figure 8. Analysis of the proposed Layer Attention Network, On the left are several observed style images, the bar charts on the right show the weights given by Layer Attention Network. The horizontal axis shows the receptive field of each Context-aware Attention Net, and the vertical axis shows their weights.

4.2. Competitors

To the best of our knowledge, no one has done font style transfer between different languages before. Thus, we can only choose some models that can be modified to this special task. We exclude models which are specially designed for compositional scripts [3] or can not handle a large font library [1] or originally designed for the unsupervised generation [21]. Finally, we choose EMD [35] and DFS [37] as our competitors. Here, we modify the input and output channel of DFS so that it can generate grayscale images. They are all trained in the same way as our proposed model. Note that for fairness, all models will not be fine-tuned when dealing with a new style or content image in the following experiments. The generation behavior is just a simple forward propagation.

4.3. Quantitative Evaluation

It is inherently difficult to quantitatively evaluate a generative model because there is no universal rule to compare the ground truth images and generated images. Moreover, tasks like artistic style transfer don't even have a standard answer. Recently, several evaluation metrics [12, 18, 26] have been proposed for measuring the performance of generative models based on different assumptions, but they are still controversial. In this paper, we use three different aspects which are pixel-level, perceptual-level, human-level accuracy to evaluate the models. As shown in Table 1, our model outperforms the existing methods on most metrics.

4.3.1 Pixel-level Evaluation

Pixel-wise evaluation compares pixels at the same position between the ground truth image and generated image. We employ the mean absolute error (MAE), structural similarity (SSIM), and multi-scale structural similarity (MS-SSIM). But pixel-level evaluation metrics often go against human intuition. Hence, we also employ metrics of the other two levels to comprehensively evaluate all models.

4.3.2 Perceptual-level Evaluation

Recently, Salimans *et al*. [26] proposed a method to evaluate generative models by computing the Fréchet Inception Distance (FID) between the feature maps of the ground truth images and generated images. Liu *et al*. [21] modified it to a conditional version (mFID) by averaging FID for each target class. In this paper, we want to evaluate the generated images from both the style and content perspective. To do this, we trained two ResNet-50 [11] networks on our proposed dataset to classify content (character) and style (font) respectively. We report the top-1 accuracy and mean FID (mFID) based on these two networks. Therefore, the perceptual-level evaluation metric consists of 4 parts, they are style-aware accuracy, content-aware accuracy, style-aware mFID, and content-aware mFID.

4.3.3 Human-level Evaluation

Our final goal is to generate images that satisfy users. So, we randomly select 39 sets of images from the output of both methods and ask users their preferred images when given the content reference and the style reference. We let users comprehensively evaluate the generated images from two perspectives witch are style matching degree and content recognizability. All experiments are completely anonymous, and the generated images are randomly shuffled so that participants cannot see which model the image comes from. We collect a total of 390 valid responses from 10 people who are both proficient in English and Chinese. Table 2 shows most participants prefer images generated by our model.

4.4. Visual Quality Evaluation

As shown in Figure 7, We randomly select some outputs from three groups of our model and other competitors. The

	Pixel-level			Content-aware		Style-aware	
	↓MAE	↑SSIM	↑MS-SSIM	↑Accuracy(%)	↓mFID	↑Accuracy(%)	↓mFID
Evaluation on the unseen character images							
CAT	0.129	0.478	0.465	96.1	59.3	38.1	406.9
w/o LA and CA	0.127	0.482	0.471	**97.1**	50.7	46.5	361.3
w/o LA	0.123	0.499	0.489	96.7	51.4	56.6	322.5
Full model	**0.121**	**0.501**	**0.493**	**97.1**	**49.8**	**58.1**	**308.9**
Evaluation on the unseen style images							
CAT	0.180	0.360	0.372	99.7	106.5	9.1	442.1
w/o LA and CA	0.181	0.360	0.367	99.7	106.9	10.9	**417.2**
w/o LA	0.186	0.353	0.360	99.6	105.6	10.0	455.4
Full model	**0.179**	**0.368**	**0.382**	**99.8**	**97.8**	**11.7**	418.8

Table 3. Ablation study on the proposed multi-language dataset. We evaluate the models on both the unseen characters and unseen styles. w/o denotes without, LA denotes Layer Attention Block, CA denotes Context-aware Block. ↑ means larger numbers are better, ↓ means smaller numbers are better.

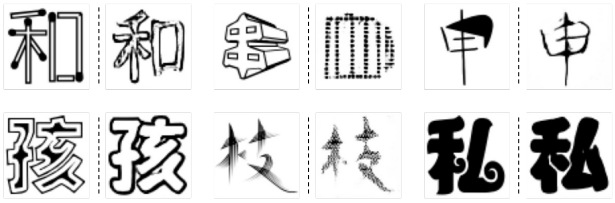

Figure 9. Failure cases of our model for some highly artistic font, the left is ground truth images, the right is generated images in each group.

first group is handwritten fonts, the second group is printing fonts, and the third group is highly artistic fonts. We can see that EMD [35] erases some thinner fonts and it can't perform well on highly artistic fonts. DFS [37] perform poorly on printed fonts. Our method can generate high-quality images of various fonts.

4.5. Ablation Study

In order to effectively evaluate the contribution of each component in the proposed FTransGAN. We gradually remove some modules and show the results. In addition, we also implement a simple method to replace our multi-level attention module by concatenating all style images in the channel axis and inputting it to a style encoder. We call it CAT model. In Table 3, we demonstrate several evaluation metrics based on the ablation study. The values of these metrics clearly show that the proposed Layer Attention Network and Context-aware Attention Network play an important role in our model. Both pixel-level and perceptual-level metrics drop quickly when removing these modules.

4.6. Attention Analysis

For further analysis of the proposed model, we visualize the weights given by Layer Attention Network. As mentioned before, three Context-aware Attention Networks in our model have different receptive fields. Layers with a small receptive field can only see a small region of the original image, while layers with a big receptive field can see almost the entire image. These weights show that for current images the model should pay more attention to local features or global features. It is observed that when processing handwritten fonts, our model tends to observe local features, while when processing printing fonts or artistic fonts, our model tends to focus on a global feature. We speculate that this is because the features of handwritten fonts are mostly concentrated in some local areas (*e.g.*, line thickness, stoke), while for some artistic fonts, a global consideration is required. We randomly select some results and illustrate them in Figure 8.

5. Conclusion

We proposed an end-to-end approach to transfer font style between different languages by only using a few samples, we also built a large-scale multi-language dataset to evaluate the models. Experimental results show the high generative ability of our proposed model and the proposed Context-aware Attention Network and Layer Attention Network play an important role.

However, it still has many shortcomings. First, Although the number of style images is arbitrary when testing, it can only receive a fixed number of style images during the training period due to the architecture design of our model. Second, As shown in Figure 9, we observed that for some highly artistic fonts, our model does not perform well. Dealing these problems is an interesting and challenging direction for future research. Besides, how to apply our model to other scenarios like artistic style transfer is also a very interesting future work.

References

[1] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7564–7573, 2018.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Junbum Cha, Sanghyuk Chun, Gayoung Lee, Bado Lee, Seonghyeon Kim, and Hwalsuk Lee. Few-shot compositional font generation with dual memory. *arXiv preprint arXiv:2005.10510*, 2020.

[4] Jie Chang and Yujun Gu. Chinese typography transfer. *arXiv*, pages arXiv–1707, 2017.

[5] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] Yue Gao, Yuan Guo, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Artistic glyph image synthesis via one-stage few-shot learning. *ACM Transactions on Graphics (TOG)*, 38(6):1–12, 2019.

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[10] Shuyang Gu, Congliang Chen, Jing Liao, and Lu Yuan. Arbitrary style transfer with deep feature reshuffle. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8222–8231, 2018.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1501–1510, 2017.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[16] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[17] Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Dcfont: an end-to-end deep chinese font generation system. In *SIGGRAPH Asia 2017 Technical Briefs*, pages 1–4. 2017.

[18] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Zhouhui Lian, Bo Zhao, and Jianguo Xiao. Automatic generation of large-scale handwriting fonts via style learning. In *SIGGRAPH ASIA 2016 Technical Briefs*, pages 1–4. 2016.

[21] Ming-Yu Liu, Xun Huang, Arun Mallya, Tero Karras, Timo Aila, Jaakko Lehtinen, and Jan Kautz. Few-shot unsupervised image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10551–10560, 2019.

[22] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[23] Pengyuan Lyu, Xiang Bai, Cong Yao, Zhen Zhu, Tengteng Huang, and Wenyu Liu. Auto-encoder guided gan for chinese calligraphy synthesis. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1095–1100. IEEE, 2017.

[24] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[26] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[27] Danyang Sun, Tongzheng Ren, Chongxun Li, Hang Su, and Jun Zhu. Learning to write stylized chinese characters by reading a handful of examples. *arXiv preprint arXiv:1712.06424*, 2017.

[28] Rapee Suveeranont and Takeo Igarashi. Example-based automatic font generation. In *International Symposium on Smart Graphics*, pages 127–138. Springer, 2010.

[29] Yuchen Tian. zi2zi: Master chinese calligraphy with conditional adversarial networks, 2017. *Retrieved Jun*, 3:2017, 2017.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[31] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[32] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489, 2016.

[33] Dongfei Yu, Jianlong Fu, Tao Mei, and Yong Rui. Multi-level attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4709–4717, 2017.

[34] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[35] Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8447–8455, 2018.

[36] Baoyao Zhou, Weihong Wang, and Zhanghui Chen. Easy generation of personal chinese handwritten fonts. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6. IEEE, 2011.

[37] Anna Zhu, Xiongbo Lu, Xiang Bai, Seiichi Uchida, Brian Kenji Iwana, and Shengwu Xiong. Few-shot text style transfer via deep feature similarity. *IEEE Transactions on Image Processing*, 2020.

[38] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.