CLF-Net: A Few-Shot Cross-Language Font Generation Method

Qianqian Jin, Fazhi He(✉), and Wei Tang

School of Computer Science, Wuhan University, Wuhan, Hubei, China
fzhe@whu.edu.cn

Abstract. Designing a font library takes a lot of time and effort. Few-shot font generation aims to generate a new font library by referring to only a few character samples. Accordingly, it significantly reduces labor costs and has attracted many researchers' interest in recent years. Existing works mostly focus on font generation in the same language and lack the capability to support cross-language font generation due to the abstraction of style and language differences. However, in the context of internationalization, the cross-language font generation task is necessary. Therefore, this paper presents a novel few-shot cross-language font generation network called CLF-Net. We specifically design a Multi-scale External Attention Module (MEAM) to address the issue that previous works simply consider the intra-image connections within a single reference image, and ignore the potential inter-image correlations between all reference images, thus failing to fully exploit the style information in another language. The MEAM models the inter-image relationships and enables the model to learn essential style features at different scales of characters from another language. Furthermore, to solve the problem that previous approaches usually generate characters with missing or duplicated strokes and blurry stroke edges, we define an Edge Loss to constrain the model to focus more on the edges of characters and make the outlines of generated results clearer. Experimental results show that our CLF-Net is outstanding for cross-language font generation and generates better images than the state-of-the-art methods.

Keywords: Cross-language Font generation · External attention · Few-shot learning

1 Introduction

Font style is the art of visual representation of text and plays a crucial role in conveying information. It can even deliver deeper meaning, such as whether the current content is delightful or horrible. Designing a font is very time-consuming and requires the highly professional ability of the designer. The designer has to make proper artistic effects for strokes so that the font not only conveys the artistic style but also guarantees the original content of the character. In addition, when designing a large font library of multiple languages, the designer needs to

S. Rudinac et al. (Eds.): MMM 2024, LNCS 14555, pp. 127–140, 2024.
https://doi.org/10.1007/978-3-031-53308-2_10

spend a lot of time and effort to keep the characters of different languages in the same style, which not only demands professional knowledge and skills but also requires the designer to be proficient in different languages.

Therefore, automatic font generation via neural networks has attracted the attention of researchers, and many GAN [5]-based models for automatic font generation have been proposed. Early models [13,14,24,25] need to be pre-trained on large datasets and then fine-tuned for specific tasks, which requires many computational resources and much effort to collect training samples. Recently, many few-shot learning methods [3,9,11,19,20,23,31,32] have been proposed specifically for the font generation task, and these models can generate complete font libraries of the same language based on a small number of samples.

Nevertheless, in many scenarios, such as designing novel covers in different translations, movie promotional posters for different countries, and user interfaces for international users, it is necessary to keep characters of different languages having the same font style. At the same time, characters of different languages vary greatly in their glyph structure, e.g., the strokes and structures of English letters are very different from those of Chinese characters. Specifically, many components of Chinese characters have no counterparts in English letters, which leads to the fact that learning the style of characters from another language is difficult and requires the model to learn high-level style characteristics. Thus, some efforts [15] attempt to use self-attention mechanism to capture style patterns in another language. However, they ignore the potential inter-image correlations between different reference images and thus fail to learn sufficiently essential features in another language. Therefore, we propose to learn better style representation in another language by analyzing the inter-image relationships between all reference images rather than simply considering the intra-image connections.

In this paper, we propose a novel model named CLF-Net. Its core idea is to learn essential style features in another language by modeling the inter-image relationships between all reference images. Specifically, we design a Multi-scale External Attention Module (MEAM) to capture style features at different scales. The MEAM not only considers the intra-image connections between different regions of a single reference image but also implicitly explores the potential inter-image correlations between the overall style images, which makes it possible to extract the geometric and structural patterns that are consistently present in the style images and thus learn the unified essential style information at different scales in another language. In addition, considering that boundary pixels play a key role in determining the overall style of Chinese characters, we define an Edge Loss to compel the model to preserve more edge information and ensure the generated characters have sharper edges with less blur. Combining these components, we have achieved high-quality cross-language font generation.

Our contributions can be summarized as follows:

1) We first implicitly consider the inter-image associations and propose a novel few-shot cross-language font generation network called CLF-Net instead of simply considering the intra-image connections.

2) We design a Multi-scale External Attention Module (MEAM) to learn the unified essential style information at different scales of characters from another language, which solves the problem that the existing font generation models can not fully exploit style information in another language.
3) We introduce an Edge Loss function to make the model generate characters with sharper edges.
4) By modeling the inter-image relationships, our approach achieves significantly better results than state-of-the-art methods.

2 Related Works

2.1 Image-to-Image Translation

Image-to-image (I2I) translation aims to learn a mapping function from the target domain to the source domain. Pix2pix [12] uses a conditional GAN-based network that requires a large amount of paired data for training. To alleviate the problem of obtaining paired data, the CycleGAN [33] introduces cycle consistency constraints, which allow I2I methods to train cross-domain translations without paired data. FUNIT [16] proposes a few-shot unsupervised image generation method to accomplish the I2I translation task by encoding content images and style images separately and combining them with Adaptive Instance Normalization (AdaIN) [10]. Intuitively, font generation is a typical I2I translation task that maps a source font to a target font while preserving the original character structure. Therefore, many font generation methods are based on I2I translation methods.

2.2 Automatic Font Generation

We categorize automatic font generation methods into two classes: many-shot and few-shot font generation methods. Many-shot font generation methods [13, 14, 24, 25] aim to learn the mapping function between source fonts and target fonts. Although these methods are effective, they are not practical because these methods often first train a translation model and fine-tune the translation model with many reference glyphs, e.g., 775 for [13, 14].

Based on different kinds of feature representation, few-shot font generation methods can be divided into two main categories: global feature representation [1, 4, 27, 31] and component-based feature representation [3, 9, 11, 19, 20, 28, 32]. The global feature representation methods, such as EMD [31] and AGIS-Net [4], synthesize a new glyph by combining a style vector and a content vector together, but they show worse synthesizing quality for unseen style fonts. Since the style of glyphs is highly complex and fine-grained, it is very difficult to generate the font utilizing global feature statistics. Instead, works related to component-based feature representation focus on designing a feature representation that is associated with glyphs' components or localized features. LF-Font [19] designs a component-based style encoder that extracts component-wise features from reference images. MX-Font [20] designs multiple localized encoders

and utilizes component labels as weak supervision to guide each encoder to obtain different local style patterns. DFS [32] proposes the Deep Feature Similarity architecture to calculate the feature similarity between the input content images and style images to generate the target images.

In addition, some efforts [9,11,15,21,23] attempt to use the attention mechanism [26] for the font generation task. RD-GAN [11] utilizes the attention mechanism to extract rough radicals from content images. FTransGAN [15] captures the local and global style features based on self-attention mechanism [29]. Our Multi-scale External Attention Module (MEAM), motivated by external attention mechanism [7], extracts essential style features at different scales for cross-language font generation.

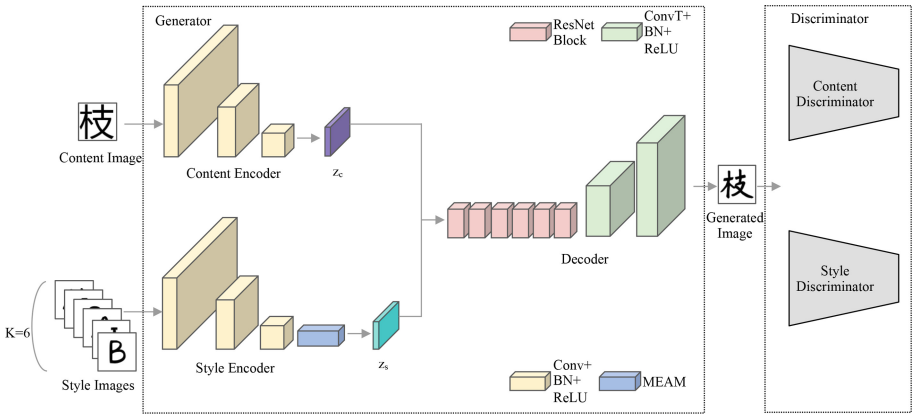

Fig. 1. Architecture overview of the CLF-Net. z_c/z_s denotes the content/style latent feature. Conv denotes a convolutional layer. BN denotes BatchNorm. MEAM denotes the Multi-scale External Attention Module. ConvT denotes a transposed convolutional layer.

3 Method Description

This section describes our method for few-shot cross-language font generation, named CLF-Net. Given a content image and several stylized images, our model aims to generate the character of the content image with the font of the style images. The general structure of CLF-Net is shown in Fig. 1. Like other few-shot font generation methods, CLF-Net adopts the framework of GAN, including a Generator G and two discriminators: content discriminator D_c and style discriminator D_s. Moreover, to make the model show enough generalization ability to learn both local and global essential style features in another language, we propose a Multi-scale External Attention Module (MEAM). More details are given in Sect. 3.2.

3.1 Network Overview

We regard the few-shot font generation task as solving the conditional probability $p_{gt}(x|I_c, I_s)$, where I_c is a content image in the standard style (e.g., Microsoft YaHei), I_s is a few style images having the same style but different contents, and x denotes the target image with the same character as I_c and with the similar style as I_s. Considering that our task is cross-language font generation, I_c and I_s should be from different languages. Therefore, we choose a Chinese character as the content image and a few English letters as the style images to train our CLF-Net. The generator G consists of two encoders and a decoder. The content encoder e_c is used to capture the structural features of the character content. The style encoder e_s is used to learn the style features of the given stylized font. Two encoders extract the style latent feature and content latent feature, respectively. Then the decoder d will take the extracted information and generate the target image $\hat{x}$. The generation process can be formulated as:

$$z_c = e_c\left(I_c\right), z_s = e_s\left(I_s\right), \tag{1}$$

$$\hat{x} = G\left(I_c, I_s\right) = d\left(z_c, z_s\right), \tag{2}$$

where z_c and z_s represent the content latent feature and style latent feature.

The content encoder consists of three convolutional blocks, each of which includes a convolutional layer followed by BatchNorm and ReLU. The kernel sizes of the convolutional layers are 7, 3, and 3, respectively.

The style encoder has the same structure as the content encoder, including three convolutional blocks. Moreover, inspired by FTransGAN [15] and external attention [7], we design a Multi-scale External Attention Module (MEAM) after the above layers to capture essential style features at different scales. More details are given in Sect. 3.2.

The decoder takes the content feature z_c and style feature z_s as input and outputs the generated image $\hat{x}$. The decoder consists of six ResNet blocks [8] and two transposed convolutional layers that upsample the spatial dimensions of the feature maps. Each transposed convolutional layer is followed by BatchNorm and ReLU.

The discriminators include a content discriminator and a style discriminator, which are used to check the matching degree from the style and content perspective separately. Following the design of PatchGAN [12], two patch discriminators utilize image patches to check the features of the real images and the fake images both locally and globally.

3.2 Multi-scale External Attention Module

Since self-attention mechanism [29] is applicable to the GAN [5] framework, both generators and discriminators are able to model relationships between spatial regions that are widely separated. However, self-attention only considers the relationships between elements within a data sample and ignores the potential

relationships between elements in different references, which may limit the ability and flexibility of self-attention. It is not difficult to see that incorporating correlations between different style reference images belonging to the same font helps to contribute to a better feature representation for cross-language font generation.

External attention [7] has linear complexity and implicitly considers the correlations between all references. As shown in Fig. 2a, external attention calculates an attention map between the input pixels and an external memory unit $M \in \mathbb{R}^{S\times d}$ by:

$$A = (\alpha)_{i,j} = \text{Norm}(FM^T), \tag{3}$$

$$F_{out} = AM, \tag{4}$$

and $\alpha_{i,j}$ in Eq. (3) is the similarity between the i-th pixel and the j-th row of M, where M is an input-independent learnable parameter that is a memory of the whole training dataset. A is the attention map inferred from the learned dataset-level prior knowledge.

External attention separately normalizes columns and rows using the double-normalization method proposed in [6]. The formula for this double-normalization is:

$$(\tilde{\alpha})_{i,j} = FM_k^T, \tag{5}$$

$$\hat{\alpha}_{i,j} = \exp(\tilde{\alpha}_{i,j}) / \sum_k \exp(\tilde{\alpha}_{k,j}), \tag{6}$$

$$\alpha_{i,j} = \hat{\alpha}_{i,j} / \sum_k \hat{\alpha}_{i,k}. \tag{7}$$

Finally, it updates the input features of M according to the similarities in A. In practice, it uses two different memory units, M_k and M_v, as the key and value to improve the capability of the network. This slightly alters the computation of external attention to

$$A = \text{Norm}(FM_k^T), \tag{8}$$

$$F_{out} = AM_v. \tag{9}$$

As mentioned above, the style of glyphs is complex and delicate. When designing the fonts, experts need to consider multiple levels of styles, such as component-level, radical-level, stroke-level, and even edge-level. Therefore, to improve the attention modules in FTransGAN [15], we design a Multi-scale External Attention Module (MEAM) to capture style features at different scales.

In particular, our method can model relationships between all style reference images from another language with the presence of the MEAM. With the MEAM, we can obtain high-quality essential style features at different scales. Specifically, when the style reference images go into the style encoder, whose architecture is shown in Fig. 2b, they will first go through three convolution blocks. Afterward, we feed the feature map outputted by the last convolutional block in the above layers into the MEAM. The MEAM first further extracts two

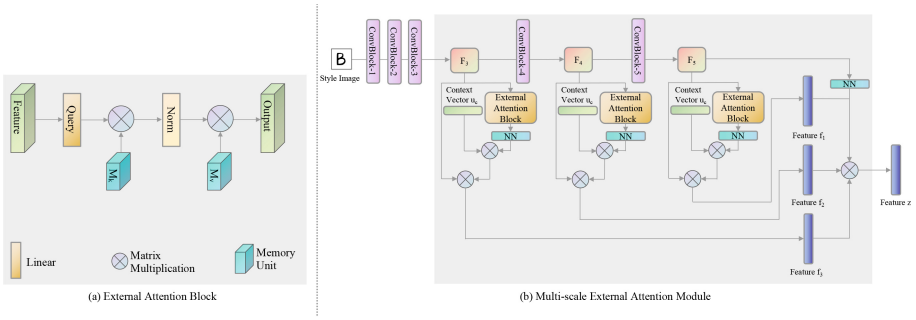

Fig. 2. The architecture of External Attention Block and the style encoder with Multi-scale External Attention Module. NN denotes a single-layer neural network. ConvBlock-1, ConvBlock-2, ConvBlock-3, ConvBlock-4, and ConvBlock-5 denote convolutional blocks, each of which includes a convolutional layer followed by BatchNorm and ReLU. F_3, F_4, and F_5 denote the feature maps with receptive fields of 13×13, 21×21, and 37×37, respectively.

feature maps separately through two consecutive convolutional blocks, each of which has a convolutional layer with kernel sizes of 3, and each convolutional layer is followed by BatchNorm and ReLU. Then the MEAM uses three juxtaposed External Attention Blocks to process the above three feature maps with receptive fields of 13×13, 21×21, and 37×37, respectively. Thus, the feature maps with different receptive fields contain the multi-scale features. The context information is obtained and incorporated into the feature map through an External Attention Block, which is computed as:

$$h_r = EA(v_r), \tag{10}$$

where EA denotes the External Attention Block, $\{v_r\}_{r=1}^{H \times W}$ denotes each region of the feature map and the new feature vector h_r contains not only the information limited to their receptive field but also the context information from other regions of other reference images.

Then, considering that not all regions contribute equally, we assign scores to each region. Specifically,

$$u_r = S_1(h_r), \tag{11}$$

$$a_r = \text{softmax}(u_r^T u_c), \tag{12}$$

$$f = \sum_{r=1}^{H \times W} a_r v_r. \tag{13}$$

That is, we input the feature vector h_r into a single-layer neural network S_1 and get u_r as the latent representation of h_r. Next, the importance of the current region is measured using the context vector u_c, which is randomly initialized and co-trained with the whole model. After that, we can obtain the normalized

score by a softmax layer. Finally, we compute a feature vector f as a weighted sum for each region v_r.

We also consider that features at different scales need to be given different weights. Therefore, we flatten the feature map given by the last convolutional block to obtain a feature vector f_m, which is inputted into a single-layer neural network S_2 to generate three weights, then we assign scores to three different scale feature vectors f_1, f_2, and f_3, respectively. These scores explicitly indicate which feature scale the model should focus on. Specifically,

$$w_1, w_2, w_3 = S_2(f_m), \tag{14}$$

$$z = \sum_{i=1}^{3} w_i f_i, \tag{15}$$

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

$$z_s = \frac{1}{K} \sum_{K} z^k. \tag{16}$$

Besides, we copy the style latent feature z_s seven times to match the size of the content latent feature z_c.

3.3 Loss Function

To achieve few-shot cross-language font generation, our CLF-Net employs three kinds of losses: 1) Pixel-level loss to measure the pixel-wise mismatch between generated images and the ground-truth images. 2) Edge Loss to make the model pay more attention to the edge pixels of characters and make the edges of generated images sharper. 3) Adversarial loss to solve the minimax game in the GAN framework.

Pixel-Level Loss: To learn pixel-level consistency, we use L1 loss between generated images and the ground truth images:

$$\mathcal{L}_{L1} = \mathbb{E}_{x,\hat{x} \in P_{(x,\hat{x})}}[\| x - \hat{x} \|_1]. \tag{17}$$

Edge Loss: Pixel-level loss is widely used in existing font generation models. They all estimate the consistency of the distribution of the two domains based on the per-pixel difference between the generated and real characters. However, in the font generation task, the weights of pixels in the images of Chinese characters are different. Different from pixels used as background or fill, boundary pixels play a key role in the overall style of Chinese characters. Therefore, our model needs to pay more attention to the edges of each Chinese character. To preserve more edge information of Chinese characters, we define an Edge Loss to limit our

model to generate results with sharper edges inspired by [21]. We utilize Canny algorithm [2] to extract the edges of generated images and the target images and utilize L1 loss function to measure the pixel distance between the two edges:

$$\mathcal{L}_{edge} = \mathbb{E}_{x,\hat{x}\in P_{(x,\hat{x})}}[\| Canny(x) - Canny(\hat{x}) \|_1]. \tag{18}$$

Adversarial Loss: Our proposed method uses a framework based on GAN. The optimization of GAN is essentially a game problem, and its goal is to allow generator G to generate examples that are indistinguishable from the real data to deceive the discriminator D. In CLF-Net, the generator G has to extract the information from the style images I_s and the content image I_c, and generate an image with the same content as I_c and the similar style as I_s, and then the discriminators D_c and D_s are used to determine whether the generated image has no difference with the reference images in terms of content and style. We use hinge loss [18] function to compute the adversarial loss as:

$$\mathcal{L}_{adv} = \mathcal{L}_{advc} + \mathcal{L}_{advs}, \tag{19}$$

$$\mathcal{L}_{advc} = \max_{D_c} \min_{G} \mathbb{E}_{I_c\in P_c, I_s\in P_s}[\log D_c(I_c) + \log(1 - D_c(\hat{x}))], \tag{20}$$

$$\mathcal{L}_{adv_S} = \max_{D_s} \min_{G} \mathbb{E}_{I_c\in P_c, I_s\in P_s}\ [\log D_s(I_s) + \log(1 - D_s(\hat{x}))], \tag{21}$$

where $D_c(\cdot)$ and $D_s(\cdot)$ represent the output from the content discriminator and style discriminator respectively.

Combining all losses mentioned above, we train the whole model by the following objective:

$$\mathcal{L} = \lambda_{L1}\mathcal{L}_{L1} + \lambda_{edge}\mathcal{L}_{edge} + \lambda_{adv}\mathcal{L}_{adv}, \tag{22}$$

where λ_{L1}, λ_{edge}, and λ_{adv} are the weights for controlling these terms.

4 Experiments

4.1 Datasets

For a fair comparison, our experiments use the public dataset of FTransGAN [15], which contains 847 grayscale fonts (stylized inputs), each font with about 1000 commonly used Chinese characters and 52 English letters of the same style. The test set consists of two parts: images with known contents but unknown styles and images with known styles but unknown contents. They randomly select 29 characters and fonts as unknown contents and styles and leave the rest as training data.

4.2 Training Details

We trained CLF-Net on Nvidia RTX 3090 with the following parameters on the above dataset. For experiments, we use Chinese characters as the content input and English letters as the style input. We set $\lambda_{L1} = 100$, $\lambda_{edge} = 10$, $\lambda_{adv} = 1$, and $K = 6$.

4.3 Competitors

To comprehensively evaluate the model, we chose the following three models, EMD [31], DFS [32], and FTransGAN [15] as our competitors. As mentioned above, previous works usually focus on font generation for a specific language, and there are few works on cross-language font generation. Therefore, in addition to FTransGAN being specifically designed for the cross-language font generation task, EMD and DFS are both designed for monolingual font generation, and we make them suitable for the cross-language task according to the modifications made by the authors of FTransGAN.

Table 1. Quantitative evaluation on the test set. The bold numbers indicate the best, and the underlined numbers represent the second best.

	Content-aware		Style-aware		Pixel-level	
	Accuracy↑	mFID↓	Accuracy↑	mFID↑	MAE↓	SSIM↑
Evaluation on the unseen character images						
EMD	81.2	116.9	24.4	597.1	**0.117**	0.497
DFS	89.2	150.0	2.7	820.6	0.185	0.303
FTransGAN	<u>97.0</u>	<u>49.8</u>	<u>58.1</u>	<u>308.9</u>	<u>0.121</u>	<u>0.501</u>
Ours	**97.5**	**45.8**	**61.6**	**294.1**	<u>0.121</u>	**0.503**
Evaluation on the unseen style images						
EMD	85.5	184.4	4.4	623.2	**0.166**	**0.384**
DFS	91.7	230.7	0.7	662.4	0.214	0.231
FTransGAN	<u>99.8</u>	<u>97.8</u>	<u>11.7</u>	**418.8**	<u>0.179</u>	0.368
Ours	**99.9**	**96.9**	**11.9**	<u>427.5</u>	0.180	<u>0.369</u>

4.4 Quantitative Evaluation

Quantitative evaluation of generative models is inherently difficult because there are no generalized rules for comparing ground truths and generated images. Recently, several evaluation metrics [30] based on different assumptions have been proposed to measure the performance of generative models, but they remain controversial. In this paper, we evaluate the models using various similarity metrics from pixel-level to perceptual-level. As shown in Table 1, our model outperforms existing methods in most metrics.

Pixel-Level Evaluation. A simple way to quantitatively evaluate the model is to calculate the distance between generated images and the ground truths. The pixel-wise assessment is to compare the pixels that are at the same position in the ground truths and generated images. Here, we use the following two metrics: mean absolute error (MAE) and structural similarity (SSIM).

Fig. 3. Visual comparison of our proposed model and its competitor.

Table 2. Effect of different components in our method. The bold numbers indicate the best, and the underlined numbers represent the second best.

	Content-aware		Style-aware		Pixel-level	
	Accuracy ↑	mFID ↓	Accuracy ↑	mFID ↑	MAE ↓	SSIM ↑
Evaluation on the unseen character images						
FM-$\mathcal{L}_{edge}$	<u>97.4</u>	<u>48.4</u>	<u>58.3</u>	<u>311.7</u>	**0.121**	<u>0.502</u>
FM-$\mathcal{L}_{edge}$-MEAM	97.1	50.7	46.5	361.3	<u>0.127</u>	0.482
FM	**97.5**	**45.8**	**61.6**	**294.1**	**0.121**	**0.503**
Evaluation on the unseen style images						
FM-$\mathcal{L}_{edge}$	**99.9**	<u>98.9</u>	<u>10.9</u>	428.5	**0.180**	<u>0.367</u>
FM-$\mathcal{L}_{edge}$-MEAM	<u>99.7</u>	106.9	<u>10.9</u>	**417.2**	<u>0.181</u>	0.360
FM	**99.9**	**96.9**	**11.9**	<u>427.5</u>	**0.180**	**0.369**

Perceptual-Level Evaluation. However, pixel-level evaluation metrics often go against human intuition. Therefore, we also adopt perceptual-level evaluation metrics to comprehensively evaluate all models. Drawing on FTransGAN [15], we use the Fréchet Inception Distance (FID) proposed in [22] to compute the feature map distance between generated images and the ground truths. This metric evaluates the performance of the network rather than simply comparing generated results. In this way, we can evaluate the performance of the content encoder and the style encoder separately. The score is calculated from the top-1 accuracy and the mean Fréchet Inception Distance (mFID) proposed by [17].

4.5 Visual Quality Evaluation

In this section, we qualitatively compare our method with the above methods. The results are shown in Fig. 3. We have randomly selected some outputs from three groups of our model and other competitors. In Fig. 3, the first group is handwriting fonts, the second group is printing fonts, and the third group is highly artistic fonts. We can see that EMD [31] erases some fonts with thinner strokes and works worse on highly artistic fonts. DFS [32] performs poorly on most fonts. FTransGAN [15] ignores fine-grained local styles and is not detailed enough in dealing with the style patterns of the stroke ends on highly artistic fonts, which causes artifacts and black spots in generated images. Our approach generates high-quality images of various fonts and achieves satisfactory results.

4.6 Ablation Study

Edge Loss. As shown in Table 2, after stripping out the Edge Loss from full model(FM), we find that Edge Loss significantly improves the classification accuracy of style and content labels. From the mFID [17] scores, we can observe that the feature distribution of the images generated by the model trained with Edge Loss is closer to the real images.

Multi-scale External Attention Module. Continue taking out the Multi-scale External Attention Module (MEAM), according to Table 2, both pixel-level and perceptual-level metrics drop rapidly.

5 Conclusion

In this paper, we propose an effective few-shot cross-language font generation method called CLF-Net by learning the inter-image relationships between all style reference images. In CLF-Net, we design a Multi-scale External Attention Module for extracting essential style features at different scales in another language and introduce an Edge Loss function that produces results with less blur and sharper edges. Experimental results show that our proposed CLF-Net is highly capable of cross-language font generation and achieves superior performance compared to state-of-the-art methods. In the future, we plan to extend the model to the task of font generation across multiple languages.

Acknowledgement. This work was supported by China Yunnan province major science and technology special plan project 202202AF080004. The numerical calculations have been done on the Supercomputing Center of Wuhan University.

References

1. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content GAN for few-shot font style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7564–7573 (2018)
2. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. **6**, 679–698 (1986)
3. Cha, J., Chun, S., Lee, G., Lee, B., Kim, S., Lee, H.: Few-shot compositional font generation with dual memory. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 735–751. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_43
4. Gao, Y., Guo, Y., Lian, Z., Tang, Y., Xiao, J.: Artistic glyph image synthesis via one-stage few-shot learning. ACM Trans. Graph. (TOG) **38**(6), 1–12 (2019)
5. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, vol. 27 (2014)
6. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: PCT: point cloud transformer. Comput. Vis. Media **7**, 187–199 (2021)
7. Guo, M.H., Liu, Z.N., Mu, T.J., Hu, S.M.: Beyond self-attention: external attention using two linear layers for visual tasks. IEEE Trans. Pattern Anal. Mach. Intell. **45**(5), 5436–5447 (2022)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
9. He, X., Zhu, M., Wang, N., Gao, X., Yang, H.: Few-shot font generation by learning style difference and similarity (2023). https://doi.org/10.48550/arXiv.2301.10008
10. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1501–1510 (2017)
11. Huang, Y., He, M., Jin, L., Wang, Y.: RD-GAN: few/zero-shot Chinese character style transfer via radical decomposition and rendering. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12351, pp. 156–172. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58539-6_10
12. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
13. Jiang, Y., Lian, Z., Tang, Y., Xiao, J.: DCfont: an end-to-end deep Chinese font generation system. In: SIGGRAPH Asia 2017 Technical Briefs, pp. 1–4 (2017)
14. Jiang, Y., Lian, Z., Tang, Y., Xiao, J.: SCfont: structure-guided Chinese font generation via deep stacked networks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4015–4022 (2019)
15. Li, C., Taniguchi, Y., Lu, M., Konomi, S., Nagahara, H.: Cross-language font style transfer. Appl. Intell. 1–15 (2023)
16. Liu, M.Y., et al.: Few-shot unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10551–10560 (2019)

17. Liu, M.Y., Tuzel, O.: Coupled generative adversarial networks. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
18. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. arXiv preprint arXiv:1802.05957 (2018)
19. Park, S., Chun, S., Cha, J., Lee, B., Shim, H.: Few-shot font generation with localized style representations and factorization. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2393–2402 (2021)
20. Park, S., Chun, S., Cha, J., Lee, B., Shim, H.: Multiple heads are better than one: few-shot font generation with multiple localized experts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 13900–13909 (2021)
21. Ren, C., Lyu, S., Zhan, H., Lu, Y.: SAfont: automatic font synthesis using self-attention mechanisms. Aust. J. Intell. Inf. Process. Syst. **16**(2), 19–25 (2019)
22. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training GANs. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
23. Tang, L., et al.: Few-shot font generation by learning fine-grained local styles. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7895–7904 (2022)
24. Tian, Y.: Rewrite: neural style transfer for Chinese fonts (2016). https://github.com/kaonashi-tyc/Rewrite
25. Tian, Y.: zi2zi: Master Chinese calligraphy with conditional adversarial networks (2017). https://github.com/kaonashi-tyc/zi2zi
26. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
27. Wen, Q., Li, S., Han, B., Yuan, Y.: ZiGAN: fine-grained Chinese calligraphy font generation via a few-shot style transfer approach. In: Proceedings of the 29th ACM International Conference on Multimedia, pp. 621–629 (2021)
28. Xie, Y., Chen, X., Sun, L., Lu, Y.: DG-font: deformable generative networks for unsupervised font generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5130–5140 (2021)
29. Zhang, H., Goodfellow, I., Metaxas, D., Odena, A.: Self-attention generative adversarial networks. In: International Conference on Machine Learning, pp. 7354–7363. PMLR (2019)
30. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586–595 (2018)
31. Zhang, Y., Zhang, Y., Cai, W.: Separating style and content for generalized style transfer. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8447–8455 (2018)
32. Zhu, A., Lu, X., Bai, X., Uchida, S., Iwana, B.K., Xiong, S.: Few-shot text style transfer via deep feature similarity. IEEE Trans. Image Process. **29**, 6932–6946 (2020)
33. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2223–2232 (2017)