Few Shot Chinese Glyph Stylization

2021 ESTR4999 Technical Report

ZHANG YUECHEN, SID 1155091988, The Chinese University of Hong Kong, Faculty of Engineering
WONG TIEN-TSIN, Supervisor, The Chinese University of Hong Kong, Faculty of Engineering

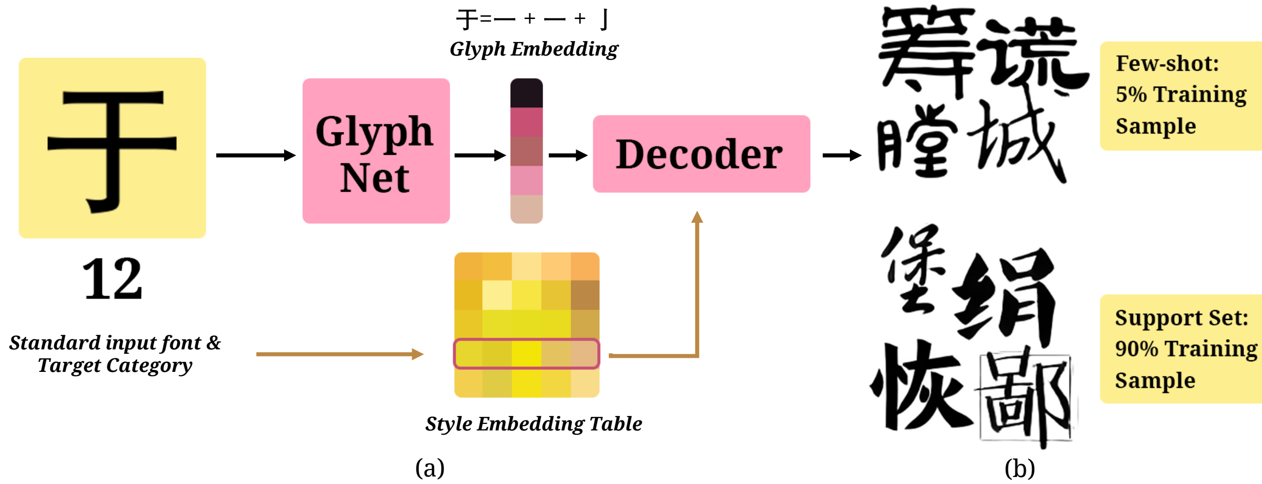


Fig. 1. The brief inference procedure of our method. a). Given an input image with a standard font style, a glyph embedding could be extracted, containing Chinese glyph representations of the given image. Meanwhile, a target style embedding is selected. Both glyph and style embedding are feed into a decoder to generate a target stylized glyph. b). The training set is combined by a support font set and a few shot font sets, each few shot font only contains 175 training samples.

High-quality font design, especially Chinese font design is a laborious task for professional designers, who need to work on the stylization process on a large glyph set. In this report, we introduced a novel method to the few-shot glyph stylization problem, which could generate a set of Chinese 3500 glyph images with only hundreds of stylized training samples. Our approach keeps the glyph consistency by a Chinese glyph component feature embedding, which could significantly increase the visual results, as well as reduce the number of network parameters when training on a large glyph classification. Besides, instead of pixel-wise loss functions, we introduce a multi-level perceptual loss to supervise the training process. This multi-level semantic representation could hold high-resolution features without strict spatial constraints. Comparing with other methods, our method could achieve a better visual result, as well as high-frequency detail preservation. The training process is done by much fewer training samples.

CCS Concepts: • **Computing methodologies** → **Computer vision representations**; **Image based rendering**.

Additional Key Words and Phrases: font completion, perceptual loss, glyph embedding, few-shot learning

1 INTRODUCTION

Novel font design is a problem that has a high threshold for professional ability. It is also a time-consuming task when the character set is large. Like Chinese and Japanese, both have a large character set with a huge glyph variance. Most Chinese fonts include at least 3500 frequently-used characters, this number becomes even larger for standard fonts. Thus, it is laborious and hard to re-design a Chinese font and maintain the style consistency between thousands of glyph. Nowadays, with the development of digital writing devices, people will have a demand to create new fonts with their customized styles. Base on this circumstance, our method is aiming at glyph stylization with a few training examples. We believe that our method significantly reduces the workload of the novel font style design workflow.

With the rapid development of deep learning and neural network, approaches are trying to solve the problem by feeding training glyph samples into a convolutional neural network [Jiang et al. 2019; Lian et al. 2016; Tian and Chong 2016]. Most of them could get a reasonable result by applying stroke extraction or style embedding. However, training strokes needs manual extraction and is only suitable for a limited number of handwriting styles. Besides, some methods need to obtain a large number of training samples to get high-quality results, which is not feasible for the real world condition. In this case, our proposed method could solve the few-shot problem without additional handcraft training data, which is more practical for font design applications.

The few-shot setting for the target style may not cover most of the glyph cases. In this paper, we apply a glyph embedding to represent the glyph feature of each character. We also design a hybrid dataset to enrich the glyph representation during training. The Chinese

Authors' addresses: ZHANG Yuechen, SID 1155091988, The Chinese University of Hong Kong, Faculty of Engineering; WONG Tien-Tsin, Supervisor, The Chinese University of Hong Kong, Faculty of Engineering.

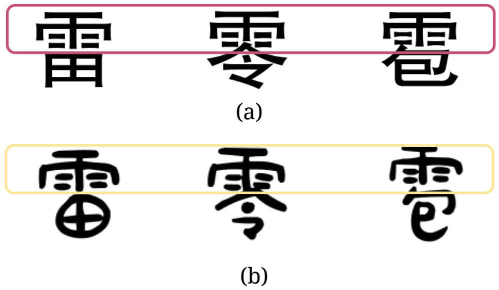


Fig. 2. An example of style inconsistency with the same glyph component. a) In a standard font style, the header part of the glyph has spatial consistency. b) A handwriting target font style with the same header glyph. There exist spatial inconsistencies within the same component.

Glyph embedding is predefined, reusable for other font styles. Another challenge for font completion is to handle the inconsistency of the handwriting font style in glyph components. Figure 2 shown an example of this inconsistency, which may introduce noise when supervise the generator pixel-wisely, resulting in wired distortions and artifacts. To solve this problem, we proposed a novel form of perceptual loss (Multi-level Perceptual Loss, MPL), which can give supervision on multilevel features with a weak spatial constrain. With the MPL term, the high-quality stylized glyph can be generated with fewer pixel-wise artifacts and structural deformations.

In brief, our work has the following major contributions:

- We propose a novel framework to do the font completion task with few training glyph samples. This framework could generate high-quality results without additional data.
- We design a Chinese glyph representation, which contains its basic glyph structure and components' combination. With this embedding, the network could get a meaningful result efficiently with arbitrary Chinese font styles.
- We propose a multi-level perceptual loss term to supervise the generation result with a weak spatial constraint, which could avoid the component inconsistency problem, preserve semantic meanings in the generation process.

In this report, we show how our method could get stylized results that surpass the current glyph stylization methods, and how the above method works for diverse Chinese font styles.

2 RELATED WORKS

As a sub-task of the image stylizing, previous font style transfer applied specified attributes from font images, and most of them successfully got a breakthrough result.

2.1 Image Stylizing

With the development and effectiveness of generative models like Generative Adversarial Network(GAN) [Goodfellow et al. 2014], neural-based general image stylizing becomes popular. Many impressive tasks, like the horse to zebra, sketch to shoes, and GTA to cityscapes, could be solved by an image to image translation framework [Isola et al. 2017; Zhu et al. 2017]. Since this kind of framework can only do the translation to one specific domain, Choi .et.al proposed a generative network that could hold multiple domain transformations [Choi et al. 2018]. Besides these methods which require a training process, Gatys .et.al proposed a classical optimization approach that can generate a stylized image with arbitrary style reference without any training processes [Gatys et al. 2016]. Then, an optimization free method was proposed as Adaptive Instance Normalization(AdaIN), which can inference with arbitrary style references with a much faster network forward process [Karras et al. 2019]. However, since font image has its property, these general image translation methods could not extract the glyph or type information from images and styles, and cannot generate satisfying results.

The exploration of the image stylizing task also produced a widely used technique called perceptual loss [Johnson et al. 2016], which evaluates the distance of intermediate feature maps from a frozen pretrained classification network. This could help the generator network to keep an eye on semantic representations instead of pixel value distance.

2.2 Font Style Transfer

To specify the scope of the image stylizing problem, image translation between different fonts becomes a research field in recent years. Some non-parametric methods use skeleton extraction and its blending weights to stylized generate glyph symbols [Suveeranont and Igarashi 2010]. Start with pix2pix, an open-source project zi2zi is proposed to solve the font stylizing problem [Tian and Chong 2016]. However, zi2zi needs a large glyph training set to obtain a good result, it fails when few glyph examples are provided, as shown in the first column of figure 7. Typeface completion is also proved available with one-shot reference inference [Park et al. 2018], but the visual quality of the generated image is not satisfying for a transfer task to an applied Chinese font. To introduce Chinese glyph characteristics, several works use stroke or component information to improve the generation result of the Chinese Glyph [Jiang et al. 2017, 2019; Lin et al. 2015]. However, they still have limitations on unknown glyph completion and component inconsistency problem. With pre-defined attributes and semi-supervise learning technique, Attribute2Font tries to generate new font with the latent style space [Wang et al. 2020], which also worked well on Chinese glyph. But this method limited the style feature space with given attributes from the training data, so it lacks customization on creating new styles. To compare with, the method proposed in this report uses a hybrid training strategy and multi-level perceptual loss to improve the visual quality of the result. The novel glyph component embedding proposed in this method is trying to solve the glyph inconsistency problem.

2.3 Few-shot Learning

Few-shot learning is firstly addressed in image classification [Snell et al. 2017]. In this task, only a few numbers of labeled training data are provided. In this research field, prototype network is a commonly used technique [Snell et al. 2017; Wang et al. 2019], which uses a statistical prototype to represent the feature of one few-shot query set. Empirically, image translation tasks need a large dataset to get

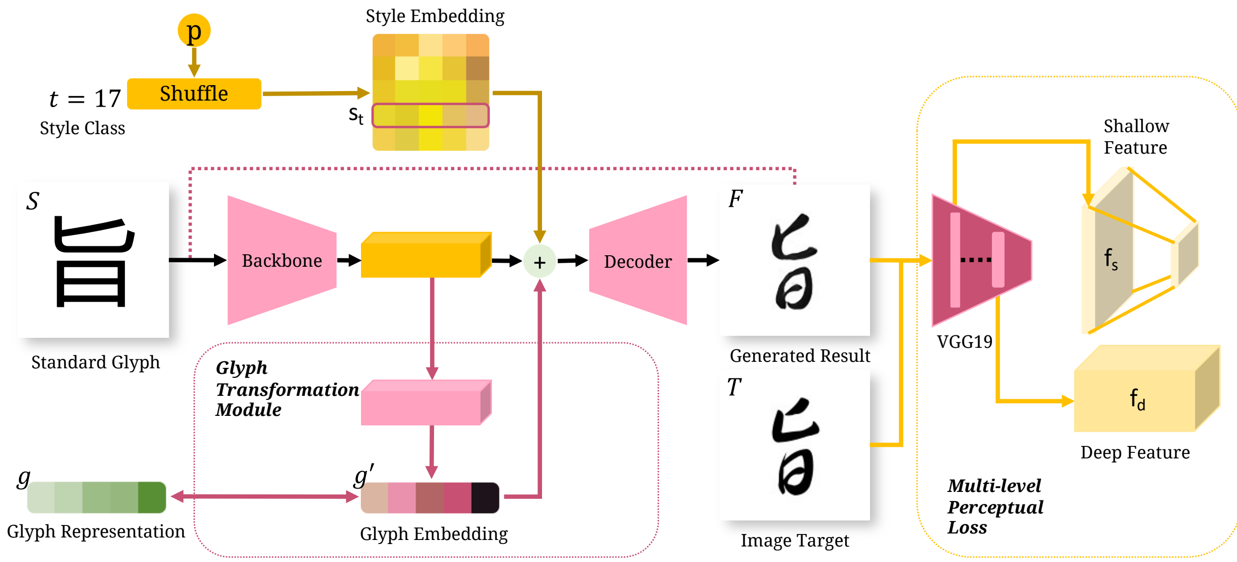


Fig. 3. The overview of our method. With the input of a standard glyph image S and its style class t, a fake result F is generated by a decoder to process the concatenation of style and glyph embedding. Meanwhile, the generated result will pass the backbone (denoted as the pink dash line) to keep the consistency of the glyph representation. The distance evaluation between F and the target image T is done by a multi-level perceptual loss module, in which, a VGG network is used as a feature extractor to get multi-stage feature representations.

a good result. Few training data may lead to overfit or gradient collapse problems. But for the font completion task, few-shot is an import property since the font stylizing is to reduce the duplication of labor.

3 METHODS

To handle problems like few-shot training and glyph inconsistency, we proposed a glyph stylizing framework as shown in figure 3. We highlight the hybrid few-shot training strategy, the glyph representation module, and the multi-level perceptual loss in this section. Since they are the main contributions of this work to improve the visual quality of glyph completion.

3.1 Overview

Figure 3 shows the overall structure of our method. We reimplement the zi2zi framework as our baseline model [Tian and Chong 2016]. The generator network includes a ResNet18 backbone and a decoder with skip connection inputs [He et al. 2016]. During the training process, an image with the standard font style S_g with its glyph index g, and its target style index t are used as the input. Noted that all glyph representations are calculated before the training process. During training, to solve the overfitting problem, we add a small probability p to shuffle the style class in batch, introduce noise on the image generator in the training process. The style embedding s_t is selected by the class index in a trainable embedding table S_E. Meanwhile, a glyph embedding $g\prime$ is extracted by a glyph transformation function, which will be described in section 3.3. Then, glyph, style, and backbone output are expanded and concatenated together, feed into the decoder, and generate the result F. Instead of using a discriminator and pixel-wise distance to identify the fake image in recent methods [Jiang et al. 2017; Park et al. 2018; Tian and Chong 2016; Wang et al. 2020], we propose a multi-level perceptual loss module to evaluate the distance between T and F, details of this module will be described in section 3.4.

3.2 Few-shot Learning

In the few-shot problem setting, we apply a hybrid learning strategy that is easy to achieve, and theoretically reasonable in terms of transfer learning. For a set of style class $t \in C_s$, we set several styles t_f as 'few-shot style set', which is our target style. For remaining font styles t_s, we set them as 'support style set', which means we already have the full font data in these styles. This setting is reasonable because we are aiming to optimize a transformation function on style spaces with the given glyph representation, the training process on the support set could obtain a good style transfer on a full glyph set, which could also benefit the transfer process on the few-shot set. It means that the decoder is learning a common transformation on a given style representation.

3.3 Glyph Representation

To present both integral feature and component feature into one representation, we design the glyph representation as a combination of the four-corner index and the component decomposition, as shown in figure 4. The first five dimensions contain the four-corner index g_{cor}, which is a 5-digit index to represent a Chinese glyph by its position (four corners and center) [Wikipedia 2021]. Each digit is from 0-9, represents a class of stroke. The remaining 156 dimensions formed as a vector g_{comp} to count the appearance frequency of basic glyph components G_c. We construct G_c by recursively parsing the rule defined by the Chinese glyph decomposition dictionary,

Fig. 4. An example of how is the glyph representation generated. It contains a four-corner index and a component-based counting vector with 156 dimensions.

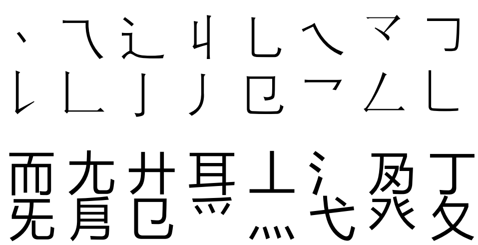


Fig. 5. Example of two types of components. The top part shows examples of 32 basic strokes in Chinese characters. The bottom part is examples of 124 sub-components, which are components that can only be composed by basic strokes or itself.

until the component 1) cannot be decomposed anymore or 2) can only be composed by the basic strokes. Figure 5 shows examples of basic strokes and components, we combine them to form the basic component set G_c. With this component set, we can decompose the glyph until it is only composed by components in G_c to get g_{comp}.

With the combined glyph representation g, a glyph estimation process is performed by several transformation functions f_n, as shown in figure 6. In the network design, the prediction of g_{cor} contains two transformation functions, implemented by multi-level perceptrons (MLPs). A 2x2 position-related embedding and a 1x1 center embedding with 10 dimensions are generated by the same processes. A global average pooling is performed before the center index embedding transformation. Meanwhile, an MLP is applied to the estimation of g_{comp}.

3.4 Loss Functions

The overall training loss L of our method is defined as following:

$$L = w_M L_M + w_D L_D + w_G L_G \tag{1}$$

In which $w_{(\cdot)}$ are the weight balancing coefficients for corresponding loss terms. Inspired by the pyramid pooling structure in the PSPNet for semantic segmentation task [Zhao et al. 2017], we design a Multi-level Perceptual Loss (MPL) to reduce the noisy negative effect on pixel-level loss supervision. For a stylizing result F and target image T, the MPL L_M is defined as follows:

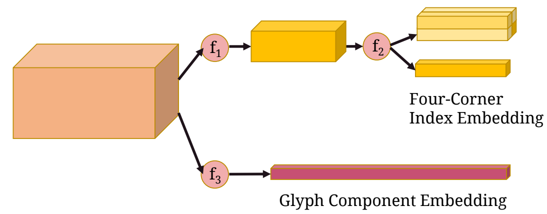


Fig. 6. Structure of the glyph transformation module. $f_{(\cdot)}$ are transformation functions implemented by MLPs.

$$L_M = \frac{w_s}{N_s} \sum \|A(f_s(F)) - A(f_s(T))\|^2 + \frac{w_d}{N_d} \sum \|f_d(F) - f_d(T)\|^2 \tag{2}$$

In which $f(\cdot)$ means a feature extraction process, we use VGG19 as our feature extractor [Simonyan and Zisserman 2014]. $\cdot_s$ and $\cdot_d$ are functions and loss weights corresponding to a shallow and a deep layers. In our experiments, we choose layer 'relu2_2' and 'relu4_3' as the shallow and deep layer, respectively. $A(\cdot)$ is the adaptive average pooling operation applied on the shallow layer, we choose 7*7 as the output size N_s. Size N_d for the deep feature map remains unchanged.

In the equation 1, the second term L_D is the direct pixel distance between F and T:

$$L_D = \frac{1}{N_p} \sum (\|T_p - F_p\|) \tag{3}$$

In which, evaluate the average distance on pixel $p \in N$ between target and fake image. Since our approach do not encourage to take much attention on the pixel-wise similarity evaluation, a distance loss with a slight weight is still benefit for the model convergence.

The glyph representation loss L_G is designed as two terms, the glyph consistency loss and the glyph embedding loss with their balancing weights $w_{consist}$ and w_{embed}:

$$L_G = w_{consist} L_{G\_consist} + w_{embed} L_{G\_embed} \tag{4}$$

$$L_{G\_embed} = L_{G\_cor,g\prime} + L_{G\_comp,g\prime} \tag{5}$$

$$L_{G\_cor,g\prime} = -\frac{1}{N_{g_{cor}}} \sum (g\prime_{cor_i} log(g_{cor_i}) + (1 - g\prime_{cor_i})(1 - g_{cor_i})) \tag{6}$$

$$L_{G\_comp,g\prime} = -\frac{1}{N_{g_{comp_i}}} \sum (\|g\prime_{comp_i} - g_{comp_i}\|^2) \tag{7}$$

$$L_{G\_consist} = L_{G\_cor,g\prime\prime} + L_{G\_comp,g\prime\prime} \tag{8}$$

The embedding loss L_{G_embed} contains two parts, L_{G_cor} and L_{G_comp}. To regress two vector embedding, the Cross Entropy loss is applied on the four-corner index part as a classification task, defined in equation 6. We also calculate the Euclidean distance between $g\prime_{comp}$ and g_{comp} as the component loss L_{G_comp} in equation 7. Meanwhile, the glyph consistency loss is set to get a cycle glyph consistency on the generated image F, which is computed

Fig. 7. Visual comparisons between font completion methods. Each row is a character from a few-shot style class

in the same manner of L_{G_embed} with glyph representation of the generated image g'' as the input.

4 RESULTS

4.1 Dataset

We choose 20 fonts with 3500 Chinese glyphs each to form our dataset. In our training set, we choose 5 fonts as the few-shot set, each of them contains 175 randomly chosen images, for the other 15 fonts, we use 85% of them as the support training set. To balance the training samples, each few-shot image will have 20 times of probability to be sampled than images in the support sets. For the testing phase, we randomly sampled 7000 unused glyphs, 350 each font as the test set. We render all images with font size 128 on a 256*256 white background.

4.2 Implementation Details

In our experiments, we set $w_M = 2$, $w_s = 5$, $w_d = 1$, $w_D = 0.2$, $w_G = 0.2$, $w_{consist} = 1$, $w_{embed} = 2$ as the default setting of balancing loss weights. For the style embedding table, the dimension of the embedding vector is set to be 256. For the backbone of the network, an ImageNet pretrained weight is used as the initialized weight [Krizhevsky et al. 2012]. All trainable parameters are optimized by an Adam optimizer with learning rate 0.0001 [Kingma and Ba 2014]. Step learning rate schedule is applied, which will half the learning rate for each 5 epoch. We train the network on a NVIDIA Titan RTX with batch size 24 for 30 epochs. We set the class shuffle probability p to 0.1.

4.3 Evaluation Metrics

To evaluate our results, we implement four evaluation metrics mentioned in Attribute2Font as our quantitative evaluation metrics [Wang et al. 2020].

- Pixel Accuracy (PixAcc): Since the generated image is not binarized, we applied a threshold (0.5 in normalized range 0-1) to do a simple classification on pixels, and calculate pixel accuracy on this binary assumption, higher means better.
- Structural Similarity (SSIM): Evaluate luminance, contrast, and structural similarity by patch-wise statistics, higher means better [Wang et al. 2004].
- Learned Perceptual Image Patch Similarity (LPIPS): Calculate cosine distance between dimensions in a feature map, and

Fig. 8. More few-shot stylization results of our method. Each row represents a style with few-shot training examples. In each pair of images, the left is the generated glyph, as the right one is the testing ground truth.

Table 1. Evaluation results of some of our experiments, **bold** values shows the best result.

Experiment	PixAcc	SSIM	LPIPS	FID
zi2zi+few shot	0.9370	0.8421	0.1589	155.4955
zi2zi+hybrid strategy	0.9523	0.8647	0.1199	43.5130
GAN+perceptual loss	0.9354	0.8356	0.1467	58.7450
ours	**0.9676**	**0.8864**	**0.05631**	**10.1581**

get a perceptual similarity, lower means better [Zhang et al. 2018].

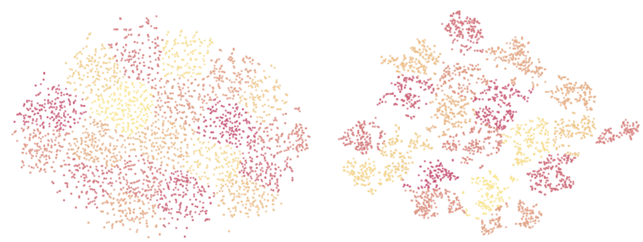
- Frechet Inception Distance (FID): Compare means and standard deviations on deep feature maps generated by pretrained InceptionV2, lower means better [Heusel et al. 2017].

The evaluation results are presented in the table 1. If a pure few-shot dataset is provided to train the zi2zi framework, the high probability of style class shuffle (0.5) and few data samples will result in a gradient collapse, resulting in unusable results. Our method achieves better quantitative evaluation results in all metrics.

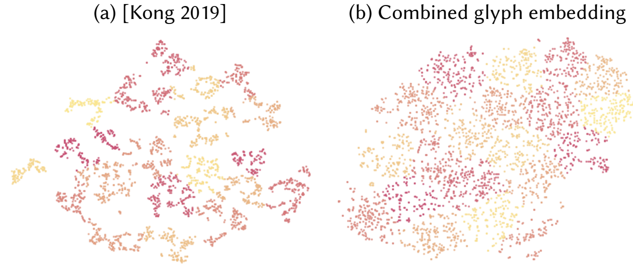
4.4 Ablation Study

There are three key contributions of our method comparing with other glyph stylizing approaches: hybrid training strategy, glyph transformation module, and MPL. The effectiveness of hybrid training strategy can be shown in the figure 7 and table 1. With the hybrid training strategy, few-shot training becomes possible.

We have tried multiple methods to represent the glyph embedding. Previously, we tried to use the embedding of pronouncing and



(a) [Kong 2019] (b) Combined glyph embedding



(c) Four-corner index (d) Glyph component

Fig. 9. A two-dimentional t-SNE visualization result of 3500 glyph embeddings. Sample points are clustered into 20 groups by the KMeans algorithm.

its four-corner index, which could be used as a glyph content representation [Kong 2019]. Now, the glyph representation used in our work is modeled by the four-corner index and the decomposition result from the Chinese decomposition dictionary. As shown in figure 9, t-SNE visualization results on the different versions of glyph

Table 2. Quantitative evaluation results to support the effectiveness of glyph representation.

Experiment	PixAcc	SSIM	LPIPS	FID
w/o. g	0.9673	0.8856	0.05746	10.5953
w. g	**0.9676**	**0.8864**	**0.05631**	**10.1581**

(a) Style with extreme deformation (b) Case with joined-up writing

Fig. 10. Failure cases of our method. In each image pair, the left is the generated image, ground truth is on the right.

embedding used in our work [Van der Maaten and Hinton 2008]. The original representation is stochastic and hard to evaluate. Oppositely, the four-corner index is less in randomness. Our combined glyph embedding in figure 9 (b) can get a reasonable representation.

To prove the effectiveness of the glyph representation, we established a new experiment without it. The evaluation metrics are shown in table 2. It seems that there is only a slight improvement shown in the quantitative result, the figure 7 presents the visualize difference between two experiments. The resulting glyph looks more stable and matches better on the target style representation.

The effectiveness of MPL is also shown in figure 7 and table 1. It remarkably reduces the negative impact on the pixel-wise distortion and glyph inconsistency problem.

5 LIMITATIONS AND FUTURE WORKS

Although the framework proposed in this paper could generate high-quality stylized glyph in most cases, it still has some shortcomings. Firstly, it cannot handle well with the style which has a very large deformation. It is mainly because of the spatial variance and glyph inconsistency. Besides, because our supervision is still on the spatial domain, and we cannot control what kinds of deep features benefit the optimization process, and our method tends to generate glyph with less joined-up writing. Both failure cases mentioned above are shown in the figure 10. For the deformation problem, a more advanced glyph skeleton transformation technique may help to solve this problem. As for the loss of join-up writing, the design of user-controllable attribute for style is necessary, which enables users to manipulate stylizing weight for glyph images.

The number of the glyph set in our dataset is still not enough for a commercial font. We will try to reproduce the GB2312 glyph set, which contains 6763 characters, as same as other works [Jiang et al. 2017, 2019]. Trying to prove quantitatively whether we can have a better result compared with other methods.

6 CONCLUSION

To conclude, this report proposed a framework to solve the few-shot Chinese glyph stylizing problem. We introduce a hybrid training strategy on this task to solve the few-shot learning problem, the glyph representation transformation module to decouple the glyph and style attributes, and multi-level perceptual loss to solve the glyph inconsistency problem. With these three key contributions proposed in our method, we proved that our method could significantly improve the visual and quantitative quality of the generated result. These methodologies could remarkably reduce the labor cost on the Chinese font design workflow and may have more contributions to other image generation applications like the Japanese and Korean glyph stylization.

ACKNOWLEDGMENTS

The authors would like to thank Prof. WONG Tien-Tsin for his supervision and guidance on this work. The authors would also like to thank the Chinese University of Hong Kong, which supports my substantial offline schedule in my last undergraduate semester.

REFERENCES

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

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

Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial networks. *arXiv preprint arXiv:1406.2661* (2014).

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

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *arXiv preprint arXiv:1706.08500* (2017).

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

Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2017. DCFont: an end-to-end deep Chinese font generation system. In *SIGGRAPH Asia 2017 Technical Briefs*. 1–4.

Yue Jiang, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. 2019. Scfont: Structure-guided chinese font generation via deep stacked networks. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4015–4022.

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

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4401–4410.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

Xiaoquan Kong. 2019. A Chinese character feature extractor, which extracts the features of Chinese characters (pronunciation features, glyph features) as features for deep learning. https://github.com/howl-anderson/hanzi_char_featurizer.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

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

Jeng-Wei Lin, Chian-Ya Hong, Ray-I Chang, Yu-Chun Wang, Shu-Yu Lin, and Jan-Ming Ho. 2015. Complete font generation of Chinese characters in personal handwriting style. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*. IEEE, 1–5.

Yonggyu Park, Junhyun Lee, Yookyung Koh, Inyeop Lee, Jinhyuk Lee, and Jaewoo Kang. 2018. Typeface completion with generative adversarial networks. *arXiv preprint arXiv:1811.03762* (2018).

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

Jake Snell, Kevin Swersky, and Richard S Zemel. 2017. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175* (2017).

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

Yuchen Tian and Zhe Chong. 2016. zi2zi: Master Chinese Calligraphy with Conditional Adversarial Networks. https://github.com/charlespwd/zi2zi.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

Kaixin Wang, Jun Hao Liew, Yingtian Zou, Daquan Zhou, and Jiashi Feng. 2019. Panet: Few-shot image semantic segmentation with prototype alignment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9197–9206.

Yizhi Wang, Yue Gao, and Zhouhui Lian. 2020. Attribute2Font: creating fonts you want from attributes. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 69–1.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

Wikipedia. 2021. Four-Corner Method — Wikipedia, The Free Encyclopedia. http://en.wikipedia.org/w/index.php?title=Four-Corner%20Method&oldid=1004484407. [Online; accessed 14-April-2021].

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2881–2890.

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