MLFont: Few-Shot Chinese Font Generation via Deep Meta-Learning

Xu Chen
School of software, Shandong University
listening@mail.sdu.edu.cn

Lei Wu*
School of software, Shandong University
i_lily@sdu.edu.cn

Minggang He
Shandong Survey and Design Institute of Water Conservancy
hmg668@163.com

Lei Meng*
School of software, Shandong University
Key Laboratory of Shandong Province for Software Engineering
lmeng@sdu.edu.cn

Xiangxu Meng
School of software, Shandong University
mxx@sdu.edu.cn

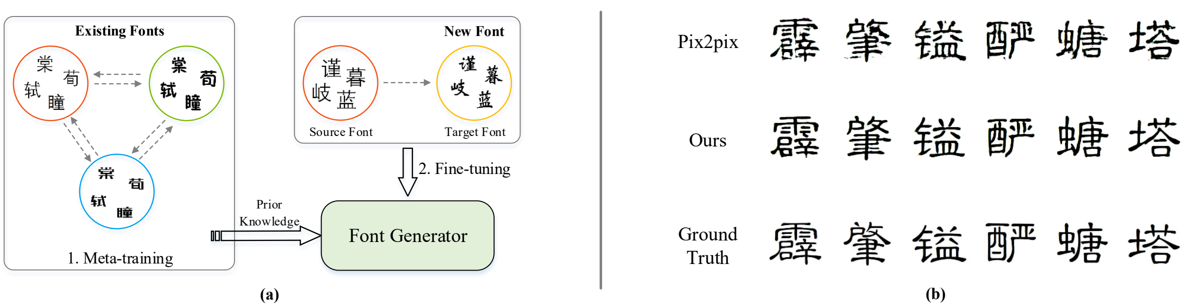

Figure 1: (a) An overview of our method, circles with different colors denote different fonts, dotted lines denote the mapping between font domains. The font generator learns prior knowledge by meta-training, and then quickly adapts to the generation of new fonts by fine-tuning. (b) Examples of Chinese character images generated by our method and pix2pix[11].

ABSTRACT

The automatic generation of Chinese fonts is challenging due to the large quantity and complex structure of Chinese characters. When there are insufficient reference samples for the target font, existing deep learning-based methods cannot avoid overfitting caused by too few samples, resulting in blurred glyphs and incomplete strokes. To address these problems, this paper proposes a novel deep meta-learning-based font generation method (MLFont) for few-shot Chinese font generation, which leverages existing fonts to improve the generalization capability of the model for new fonts. Existing deep meta-learning methods mainly focus on few-shot image classification. To apply meta-learning to font generation, we present a meta-training strategy based on Model-Agnostic Meta-Learning (MAML) and a task organization method for font generation. The meta-training makes the font generator easy to fine-tune for new font generation tasks. Through random font generation tasks and extraction of glyph content and style separately, the font generator learns the prior knowledge of character structure in the meta-training stage, and then quickly adapts to the generation of new fonts with a few samples by fine-tuning of adversarial training. Extensive experiments demonstrate that our method outperforms the state-of-the-art methods with more complete strokes and less noise in the generated character images.

CCS CONCEPTS

• **Computing methodologies → Computer vision.**

KEYWORDS

Font generation; font style transfer; meta-learning; few-shot learning

ACM Reference Format:
Xu Chen, Lei Wu, Minggang He, Lei Meng, and Xiangxu Meng. 2021. MLFont: Few-Shot Chinese Font Generation via Deep Meta-Learning. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463606

*Corresponding author.

ICMR '21, August 21–24, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463606

1 INTRODUCTION

The font library is a collection of electronic characters and is widely used on various terminal devices. The number of font products has grown rapidly in the past two decades, but the existing font resources still cannot meet the diverse and personalized needs of the digital age. Currently, the production of the font library mainly relies on the manual work of designers. Unlike the English, which contains only a small number of characters, the Chinese charset contains a huge number of characters (70,244 characters in GB18030). Except for large numbers, some Chinese characters are comprised of lots of strokes, which result in complex shapes and structures. Chinese characters are the most time-consuming part of the font library generation. The automatic generation of Chinese fonts remains a difficult problem.

Most computer graphics-based methods generate a new font library by reusing and assembling radicals[18, 39, 41]. Typically, existing reference glyphs are first decomposed into predefined radicals. Then, the remaining glyphs are generated by the assembly of predefined radicals. Such methods inevitably require manual intervention and the domain knowledge of Chinese characters. Deep learning-based methods generate new fonts in an end-to-end manner by learning a mapping from an existing font domain to the new font domain[4, 12, 13, 28, 34, 38]. They rely on a lot of reference samples of new fonts to learn the mapping that can generalize a large number of characters. In the case of few samples, these methods cannot avoid the problem of overfitting and generate blurred and incorrect strokes.

In the real world, relying on the knowledge of character structure they have learned before, human beings can easily infer the appearance of the remaining glyphs of a new font from a few samples. Inspired by this idea, we propose a novel deep meta-learning-based font generation method called MLFont that leverages the prior knowledge of the character structure learned from existing font sets to generate new fonts with a few samples. Meta-learning aims to transfer meta-knowledge from similar tasks to new tasks through task-level learning, which mainly focuses on few-shot image classification. In order to apply meta-learning to font generation, we present a task organization method for font generation.

Our font generator consists of two encoders to extract the content and style of glyphs separately, and a decoder for mixing content and style features to generate the characters of the target font. We present a meta-training strategy based on MAML, which trains the font generator through a large number of random font generation tasks. After meta-training, only a few samples of new fonts are needed, and the font generator can adapt to the generation of new fonts by fine-tuning of adversarial training.

We evaluated the proposed MLFont through the generation of multiple new fonts and compare it with existing methods. The experimental results demonstrate that our method significantly outperforms the state-of-the-art methods with more complete strokes and less noise in the generated character images. We analyzed the interpretability and effectiveness of our method through the ablation study. Besides, the case study shows that our method has good generalization performance for characters that have never appeared in the meta-training. The main contributions of our work are as follows:

- We propose a novel deep meta-learning-based font generation method (MLFont) for few-shot Chinese font generation, which leverages existing fonts to improve the generalization capability of the model for new fonts.
- We propose a general framework for applying meta-learning to font generation, which consists of a meta-training strategy based on MAML and a task organization method for font generation. To the best of our knowledge, our proposed MLFont is the first font generation method based on meta-learning.
- Extensive experiments demonstrate that our method significantly outperforms the state-of-the-art methods with more complete strokes and less noise in the generated character images.

2 RELATED WORK

2.1 Image-to-Image Translation

The image-to-image translation aims to transfer the input image from the source domain to a target domain. It has a wide range of applications in many image processing problems, such as semantic segmentation[19, 23], style transfer[10, 14], pose estimation[29, 30].

With the development of Generative Adversarial Networks [8, 21, 22, 24], many classic image-to-image translation models have been proposed[11, 40]. Isola *et al*. proposed pix2pix[11] based on conditional GANs, which is a strong baseline image-to-image translation model. The training of pix2pix requires paired images. To solve the training problem without paired images, cycleGAN is proposed for unpaired image-to-image translation[40]. Cyclegan can generate images of the target domain in an unsupervised manner.

At the same time, as a kind of image-to-image translation task, style transfer has attracted the attention of many researchers in recent years. Gatys *et al*. first use convolutional neural networks for style transfer by minimizing the difference of Gram matrices[7]. Since then, many feed-forward networks for style transfer have been proposed[10, 14, 31], some methods can generate high-quality images[20, 32]. For the generation of Chinese fonts, the generated image must have complete strokes and sharp glyphs, and the overall glyphs should be consistent with the target font style. While ordinary style transfer only needs to check style consistency from the perspective of visual art.

2.2 Font Generation

Most existing methods generate new fonts through the mapping from the source font domain to the target font domain. Zi2zi[28] is the first method to generate Chinese glyph images using GANs. Similar to zi2zi, both DCFont[12] and SCFont[13] use conditional GANs with font category embedding for Chinese font generation. DenseNetCycleGAN[4] uses cycleGAN for Chinese characters generation. With pix2pix as the base model, MTfontGAN[34] generates multiple new fonts in parallel by multi-task learning.

Besides, glyphs can be regarded as a combination of semantic content and font style, and several methods deal with font generation in the form of style transfer. EMD[38] is a font style transfer model with a bilinear mixer. SA-VAE[27] is a Chinese character Variational Auto Encoder based on the domain knowledge of character structure.

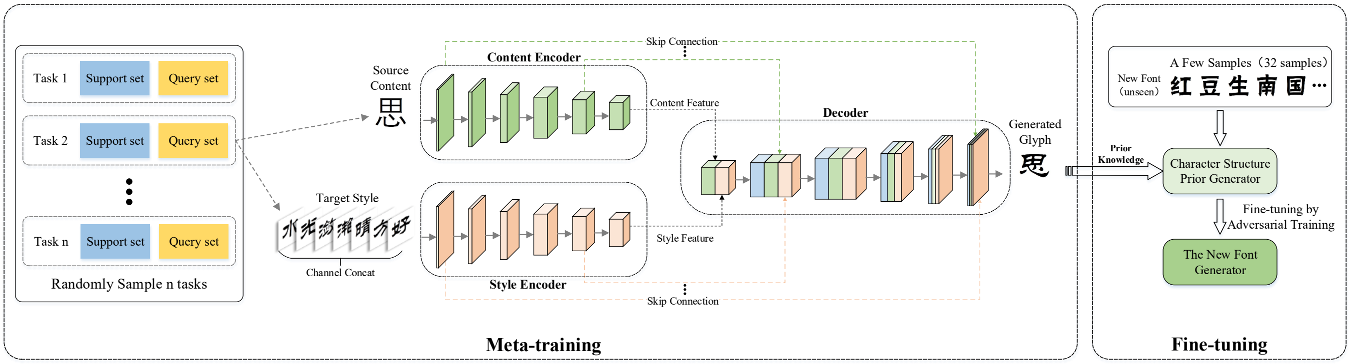

Figure 2: The overview of our proposed method for few-shot Chinese font generation. In the meta-training stage, the model parameters are continuously updated by meta-optimization, and each meta-optimization is performed using n random font generation tasks. The specific process and task organization are introduced in section 4.2. After obtaining the prior model, we fine-tune it to quickly adapt to the new font generation with a few samples by adversarial training.

Artistic font generation (so-called text effects transfer) focus on the texture transfer of glyph images, which is related to our work. Multi-ContentGAN[3] is the first method for the English artistic font generation, which cannot handle the Chinese characters. Both TET-GAN[35] and AGIS-Net[6] are proposed for the Chinese artistic font generation. Existing text effects transfer methods focus on the transfer of texture styles, and cannot cope with the task of font generation for geometric style transfer.

2.3 Meta-Learning

Deep meta-learning attempts to solve the limitations of deep learning in the case of few samples from the perspective of "learning how to learn"[9]. Meta-learning has been shown to achieve promising results for few-shot image classification, the classifier can recognize the image of unseen categories based on a few samples. Existing meta-learning methods are usually divided into three categories: optimization-based methods[5, 25], model-based methods[1, 26], and metric-based methods[16, 33]. The optimization-based methods are easier to integrate with other tasks because they do not rely on specific models. Several methods have proposed combining meta-learning with GANs to further improve the performance of few-shot image classification[2, 37]. Recently, several works use meta-learning for generative tasks, such as music generation[17] and talking head generation[36]. While our work deals with the task of the Chinese font generation. To the best of our knowledge, we are the first to use meta-learning for font generation.

3 PROBLEM FORMULATION

Font generation refers to generating all glyph images of the target font library based on some reference samples in an end-to-end manner. First, a source font is used to form a training set with the reference sample of the target font. Then the font generator learns a mapping from the source font domain to the target font domain. Subsequently, the mapping is used to generate all glyph images of the target font except the reference sample.

The generation of a glyph image is formulated as $\{x_i^p \rightarrow x_i^q\}$, where x denotes a glyph image, the subscript i denotes the i-th character in the font library, the superscript p and q denote the source font and the target font respectively. This mapping can be further formulated as $\{x_i^p, x_k^q \rightarrow x_i^q\}$, where x_k^q can also be multiple glyph images to provides the guidance of the target font style. The glyph image generation is an image-to-image translation process conditioned on the target font style.

4 METHOD

4.1 Overview

In this section, we detail our deep meta-learning method for few-shot Chinese font generation. Our approach leverages meta-learning to seek the general rules for font generation among random font generation tasks. That is, we utilize existing font sets to learn the prior knowledge of character structure for the new font generation. The overview of our approach is shown in Figure 2, which is composed of two stages, meta-training and fine-tuning. In the first stage, we first randomly sample to form font generation tasks, and then divide the task into a support set and a query set. We train the font generator by meta-optimization. In the second stage, we fine-tune the font generator on a new font with a few samples (as few as 32 samples) by adversarial training, to obtain the new font generator. The new font generator can generate all glyph images of the new font library (20,902 characters).

In our method, the model must be able to cope with multi-domain font generation tasks. As shown in Figure 2, the font generator consists of a content encoder, a style encoder, and a decoder. The content encoder extracts the character content of a glyph image. The style encoder extracts the font style of several glyph images concatenated in channel dimensions. The decoder mixes the extracted content and style features to output the target glyph image. The content skip-connection guarantees the accuracy of the glyph skeleton at all scales, the style skip-connection provides style guidance to the decoder layer-by-layer. In the fine-tuning stage, an additional discriminator is used for adversarial training to improve the quality of the generated glyph images.

4.2 Meta-training Stage

We adopt an optimization-based meta-learning method for the meta-training, which is a bilevel optimization strategy using multiple font generation tasks. This training strategy makes the font generator easier to adapt to new font generation tasks while alleviating over-fitting on few samples. The bilevel optimization requires the dataset to be organized into different tasks with support set and query set. The task organization we designed for the font generation is given below.

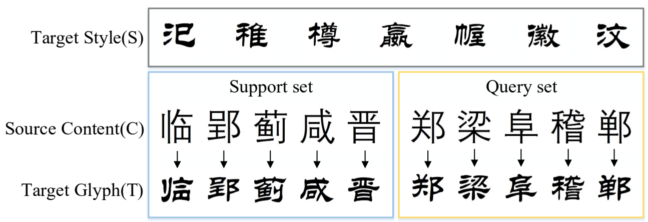

Figure 3: A specific task with ten target glyph images. The arrow denotes the mapping from the source font (C) to the target font (T) conditioned on the target style (S). The support set is used to learn the mapping for the current task, and then the query set is used to evaluate the generalization performance of the mapping.

Task Organization. In the meta-training stage, a font generation task refers to generating multiple target glyph images with the same font style. One task is formulated as $\mathcal{T} = \{C, S \rightarrow T\}$, where C and T refer broadly to several source and target glyph images respectively, S represents a set of target style images. A specific task with ten target glyph images is shown in Figure 3. C comes from the source font, S and T come from the target font. S consists of seven glyph images. There are ten source and target glyph images in C and T respectively.

$\mathcal{T}$ is divided into support set and query set. We use the support set to calculate the adapted parameters of the model for the current font generation task and evaluate its generalization performance on the query set. Note that each $\mathcal{T}$ is dynamic and random in the meta-training, which means that the source font and target font are randomly selected without repetition. The characters in $\mathcal{T}$ are also random. Random tasks can improve the generalization capability of the model.

Meta-optimization. In the meta-training, the parameters of the font generator are continuously updated by meta-optimization until convergence. One meta-optimization is performed through a batch of font generation tasks $\{\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_n\}$, where $\mathcal{T} = \{C, S, T\}$, each task comes from random sampling on the meta-training set. We use G_θ to represent the font generator with parameters θ. When adapting to the task $\mathcal{T}_i$, the parameters θ become θ'_i, where $i \in \{1, 2, ..., n\}$. The adapted parameters θ'_i is computed using m gradient descent updates on the support set. For one of the m gradient updates,

$$\mathcal{L}^i_{support} = \| G_{\theta'_i}(C^i_{support}, S^i) - T^i_{support} \|_1, \tag{1}$$

$$\theta'_i = \theta'_i - \alpha \nabla_{\theta'_i} \mathcal{L}^i_{support}(G_{\theta'_i}), \tag{2}$$

Algorithm 1: MLFont

Input: Meta-training set $\mathcal{D}_{meta}$, font generator G, gradient update steps m, number of tasks n, learning rate α, β, and γ, new font training set $\mathcal{D}_{new}$
Output: The new font generator G'
`/* meta-training */`
Initialize Parameters θ
while *not done* **do**
  Randomly sample n tasks ($\mathcal{T}_{1\sim n}$) from $\mathcal{D}_{meta}$
  $\mathcal{L}_{meta} = 0$
  for $i \in \{1,2,...,n\}$ **do**
    $\theta'_i = \theta$
    for $j \in \{1,2,...,m\}$ **do**
      Evaluate $\mathcal{L}^i_{support}$ by Eq. 1
      Evaluate $\nabla_{\theta'_i} \mathcal{L}^i_{support}(G_{\theta'_i})$
      $\theta'_i = \theta'_i - \alpha \nabla_{\theta'_i} \mathcal{L}^i_{support}(G_{\theta'_i})$
    end
    Evaluate $\mathcal{L}^i_{query}$ by Eq. 3
    $\mathcal{L}_{meta} = \mathcal{L}_{meta} + \mathcal{L}^i_{query}$
  end
  Optimize θ via Adam($\mathcal{L}_{meta}$, θ, β)
end
`/*` θ `become` θ' `after meta-training */`
`/* fine-tuning */`
for *samples in* $\mathcal{D}_{new}$ **do**
  Evaluate $\mathcal{L}_{new}$ by Eq. 7
  Optimize θ' via Adam($\mathcal{L}_{new}$, θ', γ)
end

where α is the learning rate of the gradient descent. After getting the adapted parameters θ'_i, we evaluate its generalization performance on the query set. The loss of query set is as follows:

$$\mathcal{L}^i_{query} = \| G_{\theta'_i}(C^i_{query}, S^i) - T^i_{query} \|_1 . \tag{3}$$

The font generator parameters are optimized in the direction of good generalization performance on the query set of $\mathcal{T}_{1\sim n}$. The meta-objective is as follows:

$$\mathcal{L}_{meta} = \sum_{i=1}^{n} \mathcal{L}^i_{query}. \tag{4}$$

The meta-optimization is performed via Adam with the learning rate β. Note that the meta-optimization aims to optimize the parameters θ, whereas the meta-objective is computed using the adapted parameters θ'_i.

4.3 Few-shot Learning by Fine-tuning

After the meta-training converges well, the font generator can quickly adapt to the generation of new fonts with a few samples by fine-tuning of adversarial training. Unlike the meta-training stage, we train the font generator with a new font generation task that fixes the source font and target font, where any font in the meta-training can be used as the source font and a new font as the target font.

Figure 4: Comparison of generated glyphs in four different styles obtained by our MLFont and other five existing methods. HZS, JTJ, JYH, and FLB are four different Chinese fonts.

Table 1: Quantitative evaluations of our MLFont and other five existing methods on four new fonts.

Font / Method	HZS			JTJ			JYH			FLB		
	L1 loss↓	IOU↑	SSIM↑	L1 loss↓	IOU↑	SSIM↑	L1 loss↓	IOU↑	SSIM↑	L1 loss↓	IOU↑	SSIM↑
Pix2pix	0.1988	0.5515	0.7015	0.1958	0.5704	0.7118	0.1809	0.5952	0.7207	0.1984	0.6406	0.7602
Zi2zi	0.3851	0.4632	0.6123	**0.0971**	0.5591	**0.8178**	0.1880	0.5024	**0.7346**	0.3413	0.5209	0.6635
DenseNetCycleGAN	0.2898	0.2637	0.6303	0.2618	0.5042	0.6644	0.2504	0.4393	0.6249	0.2922	0.4927	0.7023
MTfontGAN	0.1969	0.5563	0.7048	0.1922	0.5781	0.7235	0.1804	0.6003	0.7220	0.1996	0.6390	0.7606
EMD	0.3168	0.3945	0.6456	0.2586	0.4129	0.6684	0.4017	0.3956	0.5974	0.2376	0.4245	0.7115
Ours	**0.1891**	**0.5817**	**0.7129**	0.1899	**0.5936**	0.7237	**0.1797**	**0.6188**	0.7248	**0.1924**	**0.6558**	**0.7618**

We use G' to represent the font generator with parameters θ' that obtained through the meta-training. An randomly initialized discriminator D is used for the training of new fonts to further improve the quality of the generated results. The objective of a GAN can be expressed as:

$$\mathcal{L}_{GAN}(G', D) = \mathbb{E}_{C,T}[\log D(C,T)] + \mathbb{E}_{C,S}[\log(1 - D(C, G'(C,S)))], \tag{5}$$

The number of target style images in S is consistent with the meta-training. The L1 loss is

$$\mathcal{L}_{L1}(G') = \mathbb{E}_{C,S,T}[\| G'(C,S) - T \|_1]. \tag{6}$$

The final objective is

$$\mathcal{L}_{new} = \arg\min_{G'}\max_{D} \mathcal{L}_{GAN}(G', D) + \lambda\mathcal{L}_{L1}(G'). \tag{7}$$

Algorithm 1 summarizes our proposed MLFont for few-shot Chinese font generation.

5 EXPERIMENTS

5.1 Datasets

We collected a dataset containing 21 fonts, each font with 6,763 character images. All images are 256x256 in size. We randomly select 12 fonts as the meta-training set, and the remaining 9 fonts as new fonts to evaluate the performance of the few-shot font generation. The abbreviations for the names of the 9 new fonts are HZS, JTJ, JYH, FLB, ZST, LS, KT, PBT, and HXW. We randomly select 32 character images as the training set of each new font. The test set of each new font consists of the remaining 6,731 characters. The Arial (a kind of font in the meta-training set) is default as the source font in our experiments.

To evaluate the generalization performance of our method for unseen character structures, we randomly selected two of the above 9 new fonts and collected an additional dataset with 20,902 characters for these two fonts. The training set of these two fonts contains 32 character images, and the remaining 20,870 characters are used as the test set. Since the meta-training set contains 6,763 characters, there are 14,139 characters in this test set that have not appeared in the meta-training stage.

5.2 Implementation Details

The two encoders of our font generator have the same architecture. The encoder consists of six convolution layers, the number of output channels of each layer is 64, 128, 256, 512, 512, and 512. The decoder consists of six upsampling layers, the number of output channels of each layer is 512, 512, 256, 128, 64, and 3. We use the discriminator in pxi2pix[11] for the training of new fonts.

In the meta-training stage, we set the number of tasks used in one meta-optimization to 3 and the number of gradient update steps for inner optimization to 1. Both meta-training and new font training use Adam[15] as the optimizer, we set all learning rates to 0.0001. In the fine-tuning stage, the weight λ is set to 10. The meta-training takes about 28 days to converge very well on an Nvidia GPU RTX2080Ti. The training of new fonts only takes a few minutes.

5.3 Comparison with Existing Methods

Comparison methods. We compare our method with five methods. A brief description of these methods is as follows.

- Pix2pix[11] is the most classic image-to-image translation model, it is the baseline for font generation.
- Zi2zi[28] is the first to use GANs for the generation of Chinese characters images.
- DenseNetCycleGAN[40] uses CycleGAN with a DenseNet module for the generation of Chinese characters.
- MTfontGAN[34] is a multi-task learning font generation method.
- EMD[38] is a font style transfer model with a bilinear mixer.

Fairness. To ensure fairness, the above five methods are pre-trained on the meta-training set of our method. Then we fine-tune them on new fonts with 32 samples. It may be that our pre-training set contains too few font categories for EMD, resulting in poor test results on the new font.

Evaluation metrics. The following three metrics are used to quantitatively evaluate the generated results. Different evaluation indicators can provide a certain reference from different angles.

- L1 loss (Mean Absolute Error) calculates the pixel error between the generated glyphs and ground truth.
- IOU (Intersection-over-Union) is a metric for object detection, where it calculates the overlap rate between the generated glyphs and the ground truth.
- SSIM (Structural Similarity) is a metric for the structural similarity of two images.

Comparison results. Figure 4 shows the visual comparison results on four fonts. Table 1 shows quantitative evaluations. The visualization results in Figure 4 clearly show that our method can accurately generate characters with complex structures. In contrast, the other five methods generate blurred and incorrect strokes. In the quantitative evaluation of Table 1, we outperform five methods on 9/12 metric data, zi2zi outperforms us on 3/12 metric data. We argue that the results of zi2zi in Figure 4 are blurred and unstable, while our results are sharp and stable. Our method significantly outperforms these five methods in few-shot Chinese font generation.

5.4 Ablation Study

5.4.1 ***Effect of Random Task Organization.*** According to previous intuition, the model is easier to train in a one-to-many manner with a fixed source font and many target fonts, just like existing font generation methods. Through experiments, we found that the one-to-many manner with random target fonts enables the style encoder to extract styles perfectly, while the content encoder is weak. We believe that the encoder needs to become more powerful to cope with the random input, which improves the generalization capability of the model. Therefore, we propose a random task organization method. The source font, target font, and characters of each font generation task are random. We use a controlled trial to verify the effectiveness of random source font. The variable of this experiment is fixed or random source font. Figure 5 shows the test results of a new font with 32 samples. Generators trained by random source fonts can generate more accurate strokes. Random tasks improve the performance of the model, and it also slows down the convergence of the meta-training.

Figure 5: Comparison of glyphs generated in the case of fixed or random source font.

Figure 6: Comparison of glyphs generated with or without inner optimization.

5.4.2 ***Effect of Inner Optimization.*** We conduct a controlled trial to verify the effectiveness of the bilevel meta-optimization strategy. Figure 6 shows the comparison of glyphs generated with or without inner optimization. Although it may not be obvious, the results of using inner optimization are closer to the ground truth. According to Algorithm 1, the inner optimization has m gradient update steps. When not using inner optimization, m is set to 0. When using inner optimization, m is set to 1. In experiments, we found that the high derivative produced by the inner optimization leads to a high computational cost. Since the results of $m = 1$ are acceptable, we set m to 1 in all experiments. More update steps yield better results but incur higher computational costs. Through the bilevel meta-optimization strategy, the model can generate more accurate glyphs.

5.4.3 ***Effect of The Number of Tasks in Meta-optimization.*** In our method, one meta-optimization is performed through n tasks.

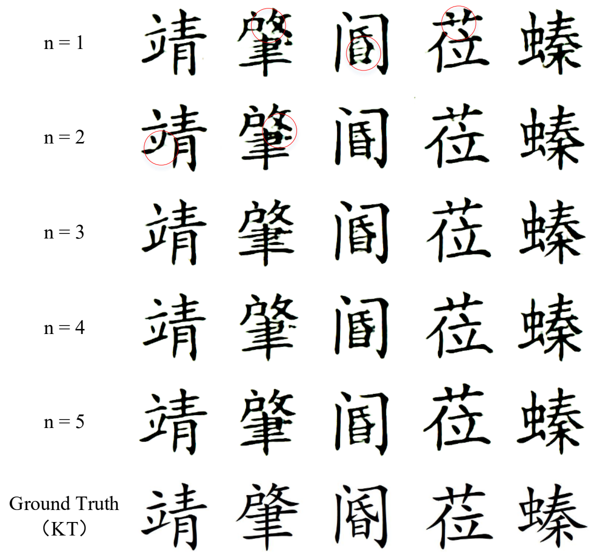

Figure 7: The effect of the number of tasks in a meta-optimization.

We conduct a controlled trial with different *n*. In the meta-training of different *n*, the number of samples seen by the font generator cannot be guaranteed to be the same. Here we control the same number of meta-optimization for the meta-training of different *n*. The results in Figure 7 shows that the strokes of the characters are incorrect when *n* is *1* and *2*. When *n* reaches *3*, the generated result becomes clear and stable. This experiment shows that the meta-optimization using multiple tasks makes the font generator learn better.

Figure 8: Comparison of glyphs generated before and after fine-tuning. The model generates correct character structures before fine-tuning, and gradually learned the style of the new font (PBT) through a few epochs.

*5.4.4 **Character Structure Learned by Meta-training**.* Figure 8 shows the comparison of generated results before and after fine-tuning. We can see that the model can generate the correct character structure before fine-tuning. For the training of new fonts, the model focuses on learning font style, so it can quickly adapt to the new font generation. This experiment verifies that the generator has learned the prior knowledge of character structure through meta-training.

5.5 Case Study

Figure 9: The effect of new font training set size. The changes of generated results when the size of the new font training set is 8, 16, 32, 64, 128, 256, and 512.

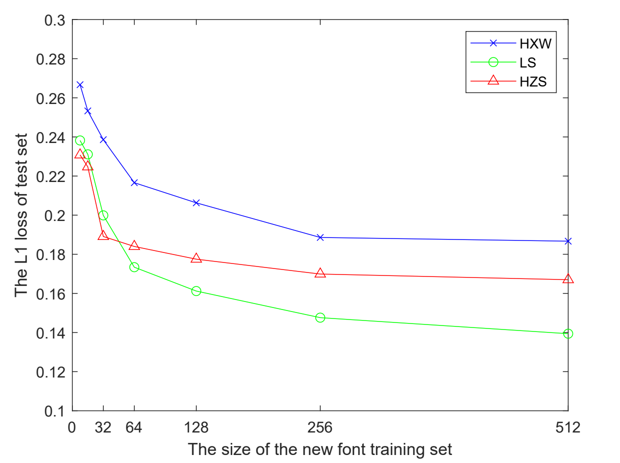

Figure 10: The trend of L1 loss on the test set as the training set size increases. HXW, LS, and HZS represent three different new fonts.

*5.5.1 **Effect of New Font Training Set Size**.* Figure 9 shows the changes of generated results when the size of the new font training set is 8, 16, 32, 64, 128, 256, and 512. It can be seen that the details of the strokes become accurate as the number of reference samples increases. Note that although the results of 8 samples are not good, the semantic content of characters is clear. This is the specialty of our method that differs from other methods. The reason for this specialty is that our model adapts to the new font generation by the adjustment on strokes-level, but 8 reference samples provide too few reference strokes to generate accurate font style. When the number of reference samples reaches 32 or 64, the result gradually becomes acceptable. Therefore, compared to other deep learning-based methods, our method can produce good results in the case of few samples. The line chart in Figure 10 shows the variation trend of the results as the number of reference samples increases. The performance of our method improves as the number of reference samples increases. This implies that our method will have better performance on the new font with more reference samples.

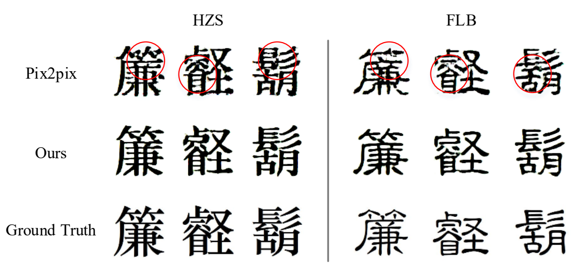

Figure 11: The test results of unseen characters generated by our MLFont and pix2pix on two new fonts (HZS and HLB), the unseen character here refers to the character structures that do not appear in the meta-training.

5.5.2 ***Generalization Performance For Unseen Characters.*** We evaluate the generalization performance of our method on two new fonts containing 20,902 characters. This experiment only requires 32 reference samples to generate the remaining 20,870 characters. In the meta training set, each font contains 6,763 characters. The two new fonts each contain 20,902 characters, two-thirds of the characters are unseen in the meta-training stage. Figure 11 shows the test results of unseen characters generated by our MLFont and pix2pix. Our method can generate clear glyph images for unseen character structures. In contrast, the results of pix2pix are blurred. This experiment shows that our method has excellent generalization performance for unseen character structures.

6 CONCLUSION

In this paper, we propose a novel deep meta-learning-based font generation method for few-shot Chinese font generation. In our approach, the model learns the prior knowledge of character structure on existing font sets through meta-training, and then quickly adapts to the new font generation by fine-tuning of adversarial training. Only a few samples are needed to generate a new large-scale Chinese font library. Extensive experiments of multiple new font generation demonstrated that our proposed MLFont consistently outperforms existing methods in terms of both the completeness of the character radicals and the sharpness of the generated glyph images.

ACKNOWLEDGMENTS

This research is supported in part by the Special Project of Science and Technology Innovation Base of Key Laboratory of Shandong Province for Software Engineering (Project ID:11480004042015).

REFERENCES

[1] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. 2016. Learning to learn by gradient descent by gradient descent. *arXiv preprint arXiv:1606.04474* (2016).

[2] Antreas Antoniou, Amos Storkey, and Harrison Edwards. 2018. Augmenting image classifiers using data augmentation generative adversarial networks. In *International Conference on Artificial Neural Networks.* 594–603.

[3] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. 2018. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 7564–7573.

[4] Bo Chang, Qiong Zhang, Shenyi Pan, and Lili Meng. 2018. Generating handwritten chinese characters using cyclegan. In *IEEE Winter Conference on Applications of Computer Vision.* 199–207.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning.* 1126–1135.

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

[7] 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.

[8] 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).

[9] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439* (2020).

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.* 694–711.

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

[16] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, Vol. 2.

[17] Weixin Liang, Zixuan Liu, and Can Liu. 2020. Dawson: A domain adaptive few shot generation framework. *arXiv preprint arXiv:2001.00576* (2020).

[18] 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 *International Performance Computing and Communications Conference.* 1–5.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3431–3440.

[20] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. 2017. Deep photo style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 4990–4998.

[21] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision.* 2794–2802.

[22] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014).

[23] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. 2015. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision.* 1520–1528.

[24] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434* (2015).

[25] Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *International Conference on Learning Representations.*

[26] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International Conference on Machine Learning.* 1842–1850.

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

[28] Yuchen Tian. 2017. zi2zi: Master chinese calligraphy with conditional adversarial networks. https://github.com/kaonashi-tyc/zi2zi.

[29] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 648–656.

[30] Alexander Toshev and Christian Szegedy. 2014. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1653–1660.

[31] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor S Lempitsky. 2016. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *International Conference on Machine Learning*, Vol. 1. 4.

[32] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. 2017. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6924–6932.

[33] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080* (2016).

[34] Lei Wu, Xi Chen, Lei Meng, and Xiangxu Meng. 2020. Multitask Adversarial Learning for Chinese Font Style Transfer. In *International Joint Conference on Neural Networks*. 1–8.

[35] Shuai Yang, Jiaying Liu, Wenjing Wang, and Zongming Guo. 2019. Tet-gan: Text effects transfer via stylization and destylization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1238–1245.

[36] Egor Zakharov, Aliaksandra Shysheya, Egor Burkov, and Victor Lempitsky. 2019. Few-shot adversarial learning of realistic neural talking head models. In *Proceedings of the IEEE International Conference on Computer Vision*. 9459–9468.

[37] Ruixiang Zhang, Tong Che, Zoubin Ghahramani, Yoshua Bengio, and Yangqiu Song. 2018. Metagan: An adversarial approach to few-shot learning. In *Advances in Neural Information Processing Systems*. 2371–2380.

[38] Yexun Zhang, Ya Zhang, and Wenbin Cai. 2018. Separating style and content for generalized style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8447–8455.

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

[40] 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.

[41] Alfred Zong and Yuke Zhu. 2014. Strokebank: Automating personalized chinese handwriting generation. In *Twenty-Sixth IAAI Conference*.