Few Shot POP Chinese Font Style Transfer using CycleGAN

Yong Yu*

1School of Management, Anshan Normal University, Anshan, China

*email: yysbs@mail.asnc.edu.cn

Abstract. The new style design of Chinese fonts is an arduous task, because there are many types of commonly used Chinese characters and the composition of Chinese characters is complicated. Therefore, the style transfer of Chinese characters based on GAN has become a research hotspot in the past two years. This line of re-search is dedicated to using a small number of artificially designed new style fonts and learning the map-ping from the source font style domain to the target style domain. However, such methods have two problems: 1. The performance on pop (point of purchase) fonts with exaggerated and random style is not satisfying. 2. Plentiful manually designed fonts are still required. In order to solve the above problems, we propose a few-shot font style transfer model based on CycleGAN. It uses meta-knowledge to reduce the use of manually designed fonts and enables each character to fully learn the knowledge contained in all new style fonts to achieve satisfying pop font style transfer effect. We also construct a dataset based on commonly used 3500 Chinese characters and verify the effectiveness of our model.

1. Introduction

In the digital age, the threshold for designing a Chinese font that can be used for computer display and paper printing is lower, but it is still a heavy work because over 3,500 characters are commonly used in daily life. This paper proposes a style transfer model based on meta learning and CycleGAN[1] to generate fonts automatically, focusing on pop (point of purchase) font style, which aims to stimulate consumption and activate the atmosphere of the store and contains more decorative elements.

However, designing pop font styles suffers a lot from strokes-deformation and randomness. As figure 1 shows, randomness is an essential part of pop font styles which highlights fashion, freedom and personality. Therefore, pop font style transfer task is more challenging, and past methods[2][3][4] have problems such as fuzzy generated characters and missing decorative elements.

Based on the above idea, we formalize pop font style transfer task as a few-shot learning problem and utilize CycleGAN combined with meta-learning methods to make font generation easier and more realistic. To summarize, our main contributions are concluded as follows:

- We propose the challenging pop font style transfer task and build a validation dataset for it.
- We propose a few-shot learning framework Meta-CycleGAN, which could be easily transferred to any GAN-based models, to greatly reduce the number of manually designed characters and perform better on pop font style transfer task simultaneously.
- We demonstrate the efficacy of Meta-CycleGAN through detailed experiments.
2. Materials and Methods

2.1. Problem Formulation

Pop font style transfer task can be formulated as learning a mapping function $F$ from source font style domain $S$ to target font style domain $T$ with samples: $\{s_i\}_{i=1}^{M}$ where $s_i \in S$ and $\{t_j\}_{j=1}^{N}$ where $t_j \in T$, $N < M$. The size of target font style samples $N$ is explored in experiment part to find a minimal value which will not induce an obvious decline in performance. Figure 2 gives an overview of proposed model structure.

2.2. Base Model

As shown in figure 2, base model is a CycleGAN. The loss function of CycleGAN contains three parts: adversarial losses, cycle consistency losses and identity loss. For generator $G_F$ and corresponding discriminator $D_F$, the adversarial loss is:

$$L_{GAN}(G_F, D_F) = E_{s \sim p(S)} \log \left(1 - D_F \left( G_F (s) \right) \right) + E_{t \sim p(T)} \log D_F \left( t \right)$$  \hspace{1cm} (1)

The adversarial loss for $G_H$ and corresponding discriminator can be defined similarly. Cycle consistency losses ensure that the cyclic transformation is able to bring the image back to the original state, which is:

$$L_{cycle} = E_{s \sim p(S)} ||G_H(G_F(s)) - s||_1 + E_{t \sim p(T)} ||G_F(G_H(t)) - t||_1$$  \hspace{1cm} (2)

Another identity loss is added to ensure real images are not changed so much after being transformed by generator. Thus the original total loss for CycleGAN is:

$$L_{cycleGAN} = L_{GAN}(G_F, D_F) + L_{GAN}(G_H, D_H) + \lambda * L_{cycle} + \lambda * L_{id}$$ \hspace{1cm} (3)
2.3. Meta Learning Procedure

The meta knowledge among episodes is prototype[5][6][7]. The meta knowledge is that characters with the same font style should stay closer in the embedding space and stay far from those with different font style. Characters which stay close the each other share similar font styles, thus generating target style characters is much easier guided by meta knowledge.

By feeding episodes into model, the model is able to learn common knowledge among episodes, and prototype encodes such knowledge in it. Such knowledge in font style contains edge shape, decoration, and other features. In one training episode, we feed training batch $B = \{(s_i, t_j)\}_{i,j=1}^K$ into CycleGAN. $K$ is the number of samples from each font style, thus we sample $K$ characters equally and repeatedly from each font style. The encoders in each generator $G_F, G_H$, which are marked as $Enc_F$ and $Enc_H$, serve as project function to embed character images in latent space. The bottleneck vector between encoder and decoder is latent embedding. First we calculate the prototype of each font style within one batch.

$$P_c = \frac{1}{K} \sum_{i=1}^{K} Enc_F(x_i)$$

Figure 3 illustrates the calculating process, from which embeddings of two font styles fall into separate clusters. The meta knowledge within episodes is the prototype of each font style. Clusters should stay far enough, and this property is encoded with a loss form:

$$L_{\text{meta-far-real}} = -\text{dist}(P_F, P_H)$$

Function $\text{dist}$ is to calculate the distance between prototypes, which is set to Euclidean distance in Meta-CycleGAN. Characters in the same cluster share similar font style. Furthermore, the prototype of each font style represents the most essential content of each style. For example, the butterfly decoration in font DieZuiQingFeng and the round edge in font ZhuLangYuanTi. By pulling the latent embedding near to the prototype, generated characters of target font style learn the essence of style knowledge and increase their randomness.

In addition, the prototype calculated by reconstructed images should be close to their original domain. Similarly, we compute the prototype of each reconstructed font style.

In the equation below, $\{f, c\} = \{F, c = S, x_i \in S\}$ or $\{f, c\} = \{H, c = T, x_i \in T\}$. $\{f, c\}$ equals to $\{F, H\}$.

$$\bar{P}_c = \frac{1}{K} \sum_{i=1}^{K} Enc_{\sim f}(G_f(x_i))$$

To encode the knowledge of reconstructed image clusters, another loss form is added.

$$L_{\text{meta-near}} = \text{dist}(P_F, \bar{P}_F) + \text{dist}(P_H, \bar{P}_H)$$

In fact we hope $\bar{P}_F$ is the same as $P_F$, which means cyclic transformation is able to bring the image back to the original state. This loss term means to enhance the cycle consistency of base model.

![Fig. 2 Model structure. We add meta-loss computing unit to original CycleGAN](image-url)
Similar as $L_{\text{meta-far-real}}$, another loss term based on $\hat{P}_F - \hat{P}_H$ is also added in order to further pull clusters far away.

\[
L_{\text{meta-far-fake}} = -\text{dist}(\hat{P}_F, \hat{P}_H)
\]  

(8)

The overall model loss is summed by CycleGAN loss terms and meta loss terms:

\[
L_{\text{total}} = L_{\text{GAN}}(G_F, D_F) + L_{\text{cycle}} + L_{\text{meta-far-real}} + L_{\text{meta-far-fake}} + L_{\text{meta-near}}
\]  

(9)

During each training episode, the calculated prototype is the estimation of real prototype representing that font style which can not be actually observed. By adding the above loss terms as regularization to font style clusters, the encoders will gradually learn a project function to embed images separately according to their style. However, at the initial stages of training the encoders are not trained well enough, so a param $\lambda_{\text{epoch}}$ dependent on epoch is added to meta loss in order to increase its weight linearly to training epochs.

\[
L_{\text{total}} = L_{\text{GAN}}(G_F, D_F) + L_{\text{cycle}} + \lambda_{\text{epoch}}(L_{\text{meta-far-real}} + L_{\text{meta-far-fake}} + L_{\text{meta-near}})
\]  

(10)

The whole training algorithm is summarized in table 1.

**Fig. 3** Calculate the prototype of two font style domains where encoder part of each generator projects character image to cluster of the other style in latent space

**Tab. 1** Meta learning with CycleGAN

| Input dataset: M character images from source style and N from target style; $K$; $\lambda_{\text{epoch}}$ initial value; epochs |
|---------------------------------------------------------------|
| Output generated character image of target font style         |

1: for e in epochs do:
2:   sample training batch as episode: $B = \{s_i, t\}_{i=1}^K$
3:   $L_{\text{GAN}}(G_F, D_F), L_{\text{cycle}}, P_S, P_T, \hat{P}_S, \hat{P}_T = \text{CycleGAN}(B)$
4:   calculate meta loss: $L_{\text{meta-far-real}}, L_{\text{meta-far-fake}}, L_{\text{meta-near}}$
5:   sum all loss terms together as $L_{\text{total}}$. Optimize parameters in CycleGAN.
6: end for

3. Experiments

3.1. Data, Baselines and Model Settings

As pop Chinese font style datasets have never been created before, we propose a dataset containing two pop Chinese font styles: *DieZuiQingFeng*, *ZhuLangYuanTi*. We choose the commonly used 3500 Chinese characters and translate characters to images of different font styles containing source font style *SimKai*. For few-shot settings, the source dataset contains 3500 samples, while target dataset contains no more than 1000 samples. Sample batch size $K$ is validated in range: \{1, 3, 5, 10, 15, 20\} and we finally choose 5.

Baseline models are CycleGAN proposed which is widely used in Chinese font style transfer task and serves as a strong baseline. Pre-trained HCCR-GoogLeNet model[8] is widely used to evaluate the quality of the generation tasks. If the generated characters are realistic, the pre-trained HCCR-GoogLeNet will also be able to classify the generated characters with high accuracy.
3.2. Model Performance on Tasks

Figure 4 shows a direct comparison between original CycleGAN and Meta-CycleGAN on DieZuiQingFeng task. Target font style datasets have 500 samples, and models are trained to the same epoch. In few-shot settings, original CycleGAN performs not well, and it neither learns decoration nor encodes randomness. For contrast, Meta-CycleGAN learns parts of decoration and really encodes randomness in font styles. For example, in the first three lines all character images learn the butterfly decoration, which is the representative part of DieZuiQingFeng style. However, none of them has such decoration in the original target style, and the first character image even changes its original heart-like decoration to butterfly and puts it at another position. Edge shapes are much easier to learn compared to decoration, as both models perform equally well on edge shape transfer. As sample number is limited, there are still some features not well learned by Meta-CycleGAN which is shown at figure 4.

On another two tasks, the direct comparison is shown in figure 7. These two tasks are easier than DieZuiQingFeng, thus all models perform better. Table 2 uses HCCR-GoogLeNet test accuracy to compare models, where Meta-CycleGAN beats CycleGAN by a large margin. Accuracy also indicates the difficulty of each task.

|                  | Meta-CycleGAN | CycleGAN |
|------------------|---------------|----------|
| DieZuiQingFeng   | 65.23%        | 55.77%   |
| ZhuMengYuanTi    | 93.15%        | 90.41%   |

![Fig. 4](image_url) Meta-CycleGAN VS CycleGAN on few-shot data set from SimKai style to DieZuiQingFeng pop style (left) and ZhuLangYuanTi pop style (right)

![Fig. 5](image_url) t-SNE visualization on two transfer tasks. Left column shows bottleneck vectors from Meta-CycleGAN, and right column shows vectors from CycleGAN
3.3. Parameter Sensitivity

In this part size of target font style dataset is discussed. Table 3 shows the change of top-5 accuracy with varied dataset size on DieZuiQingFeng. On the most severe case where target set has 200 fonts, our model still receives 52.1% accuracy.

| Dataset Size | Top-5 Accuracy |
|--------------|----------------|
| 1000         | 90.15%         |
| 900          | 88.27%         |
| 800          | 84.3%          |
| 700          | 80.42%         |
| 600          | 75.61%         |
| 500          | 70.17%         |
| 400          | 65.23%         |
| 300          | 56.17%         |
| 200          | 52.1%          |

3.4. Visualization for Bottleneck Vectors

In this part we declare that bottleneck vectors from source and target font styles are projected into different clusters in latent space. Bottleneck vectors from generator are 64-dim, so we use t-SNE[8] as dimension reduction and visualization method to project these vectors into 2-dim space. Each vector represents a character. Figure 5 gives visualization result on two task: DieZuiQingFeng and ZhuLangYuanTi. The margin between clusters reduce as the difficulty of tasks decrease. Left column shows visualization from Meta-CycleGAN, from which we could draw a separate line for two classes, while samples on the right column could not be separated clearly.

4. Conclusion

In this paper, we first propose the pop Chinese font style transfer task with few-shot settings, then utilize episodic learning and meta knowledge to solve this task with proposed model Meta-CycleGAN. Experiments are conducted to show model performance on different difficulty-level tasks. Future works include improving model structure to gain better performance and introduce attention mechanism to calculate prototype more precisely.

Acknowledgments

This paper is funded by Anshan Normal University (Project No.18kyxm018). The title of the project is “Font Style Generation via Deep Learning”.

References

[1] Zhu J Y, Park T, Isola P, Efros A A 2017 Unpaired image-to-image translation using cycle-consistent adversarial networks Proceedings of the IEEE International Conference on Computer Vision Venice pp 2223-2232

[2] Chang B, Zhang Q, Pan S, Meng L 2018 Generating handwritten chinese characters using cyclegan 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) IEEE pp 199-207 https://arxiv.org/pdf/1801.08624.pdf

[3] Chang J, Gu Y, Zhang Y, Wang Y F, Innovation C M 2018 Chinese Handwriting Imitation with Hierarchical Generative Adversarial Network BMVC p 290 http://bmvc2018.org/contents/papers/1141.pdf

[4] Lyu P, Bai X, Yao C, Zhu Z, Huang T, Liu W 2017 Auto-encoder guided gan for chinese calligraphy synthesis. In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) IEEE vol. 1 pp 1095-1100 https://arxiv.org/pdf/1706.08789.pdf

[5] Wang Y, Yao Q, Kwok J T, Ni L M 2020 Generalizing from a few examples: A survey on few-shot learning ACM Computing Surveys (CSUR) 53(3) pp 1-34 DOI: 10.1145/3386252

[6] Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D 2016 Matching networks for one shot learning Proceedings of the 30th International Conference on Neural Information Processing Systems Barcelona Spain pp 3637-3645

[7] Snell J, Swersky K, Zemel R 2017 Prototypical networks for few-shot learning Proceedings of
the 31st International Conference on Neural Information Processing Systems (NIPS’17) USA pp 4080–4090

[8] Zhong Z, Jin L, Xie Z High performance offline handwritten chinese character recognition using googlenet and directional feature maps 2015 13th International Conference on Document Analysis and Recognition (ICDAR) https://arxiv.org/ftp/arxiv/papers/1505/1505.04925.pdf