A method for generating various style Chinese fonts in the absence of training data

1st Kang Shi
School of Computer Science and Software Engineering
East China Normal University
Shanghai, PR China
51184501142@stu.ecnu.edu.cn

2th Tian-Ming Bu*
Shanghai Key Laboratory of Trustworthy Computing
East China Normal University
Shanghai, PR China
tmbu@sei.ecnu.edu.cn

***Abstract*—In recent years, the generation of arbitrary style fonts has drawn broad attention. At present, the common idea of most methods is to train the model through a large number of text samples of a specific style, so that the model can learn the font style, and then automatically generate all the text of the style. However, in the case of insufficient sample data, almost all the current methods fail. In this paper, we investigate how to generate arbitrary style Chinese fonts automatically, especially when the sample data of specific style is insufficient. In order to solve the problem, we propose a new frame. The method can be used to generate the fonts of Chinese ancient calligraphers whose relics are rare. The experimental results show the effectiveness of the new frame.**

***Index Terms*—text style transfer, generative adversarial network, convolutional neural network**

I. Introduction

The text style transfer is an end-to-end image conversion, which is to generate a new image combining the content of the source image and the style of the target style image. In the case that there are enough samples of the target style, after seeing different characters of the same style font, the generative model can learn the target style, and then generate all the characters of the style. But how to automatically generate all the characters in the target style when there are not enough samples, is a problem that has not been solved yet.

In order to solve the problem,this paper proposes a new framework, which is composed of two parts: the first part is a model which is called the GTPD model, used to get the probability distribution of the font type of training sample, and the second part is the text style transfer model. The GTPD model rebuild training data set based on the font type probability distribution of a small number of samples of a specific style. The training data set constructed by the GTPD model is then used for the training of the text style transfer model. Further, because the existing text style transfer models perform not very well, we improve the existing style transfer models and design a multi-style transfer model with better performance. The structure of the new framework is shown in Fig. 1. In summary, the main contributions of this paper are in two aspects:

- we optimize the existing text style transfer model and propose a multi-style transfer model. The new model can not only generate more realistic text, but also learn multiple different styles of fonts at the same time. In addition, the new model can also generate the same font of traditional Chinese characters by learning the style of simplified Chinese characters.
- we propose a new framework for text style transfer in the absence of training data.

II. RELATED WORK

Gatys et al. [1] successfully applied the Convolutional Neural Networks(CNNs) [2] to neural style transfer, breaking the bottleneck that one program can only transfer one style.

When the CNNs were applied to neural style transfer, some researchers attempted to apply the CNNs to text style transfer. Yunchen Tian et al. [3] established an open source project on Github and proposed the Rewrite model. But the performance was not good. Pengyuan Lyu et al. [4], based on the Auto-Encoder model, proposed the Auto-Encoder Guided GAN(AEGAN) model for generating Chinese calligraphy characters. Compared AEGAN with pix2pix, the loss function of AEGAN model contains the reconstruct loss function, which is aim to guide the transfer to learn the detailed stroke information from auto-encoder's low level features. Samaneh Azadi et al. [5] proposed the multi-content GAN model, and tried to learn the style through a few Latin characters with a specific style, and then generated all the Latin characters of that style. Shuai Yang et al. [6] proposed a scale-controllable module to empower a single network to continuously characterize the multi-scale shape features of the style image and transfer these features to the target text. In Anna Zhu's paper [7], the output text font essentially is the same as the input, except the decoration added on the output text font. While in our work, the input and output are different fonts. We solve the problem of style transfer of

This work are supported by National Key Research and Development Programs (2019YFB2102600, 2019YFA0706404).

*Corresponding author.

DOI reference number: 10.18293/SEKE2021-114.

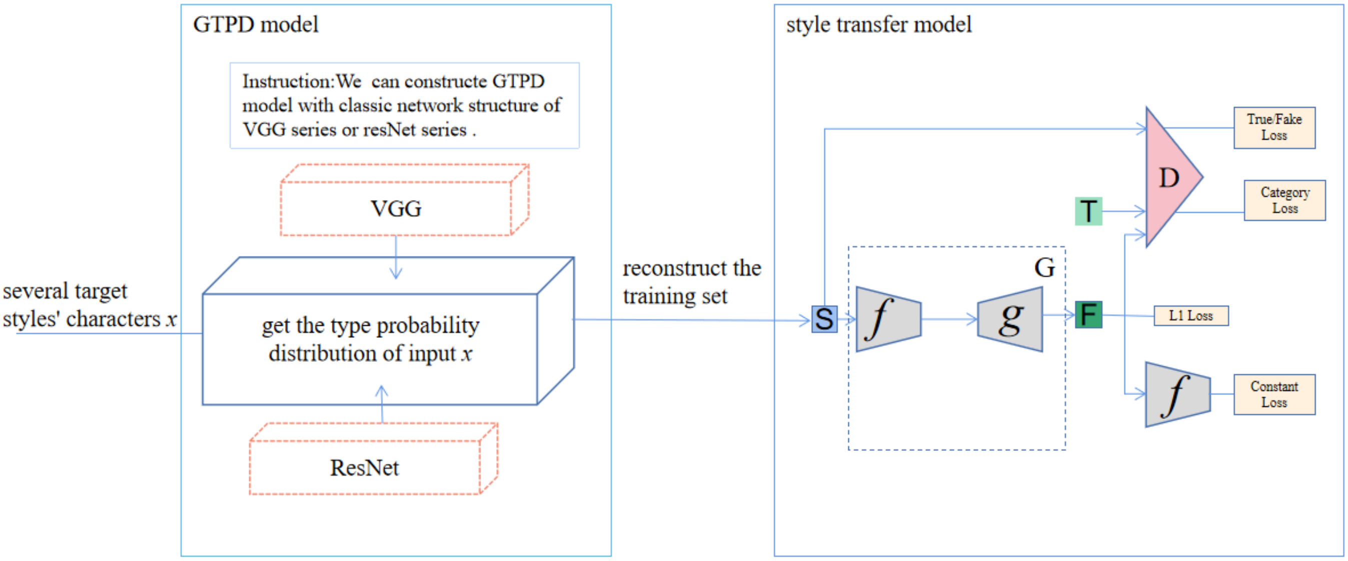

GTPD model
Instruction:We can constructe GTPD model with classic network structure of VGG series or resNet series .
VGG
get the type probability distribution of input x
ResNet
several target styles' characters x
reconstruct the training set
style transfer model
S
f
g
G
F
T
D
True/Fake Loss
Category Loss
L1 Loss
f
Constant Loss

Fig. 1. The structure of new frame.

different fonts from only a few referenced samples(output font).

III. Proposed Method

In this paper, we improve the existing text style transfer model in some aspects. Firstly, based on the pix2pix model, constant loss function and category loss function are introduced to make the generative model perform better in the generated details. Then, a new framework is proposed to solve the problem of style transfer cannot work in the absence of training samples. The multi-style transfer model and the new framework are described in detail below.

A. Constant Loss

The learning process of human beings is the process of analogy. Our generative model is actually to learn how to map the sample space to the target space, namely, learn the mapping from a input domain to an output domain. Inspired by this, we apply the constant loss function proposed by Yaniv Taigman et al. [8] to our generative model.

Given two related domains S and T, the generative model needs not only to learn to map from domain S to domain T, but also to ensure the correlation between S and T during the mapping process. Assume that element $x \in S$ and the generative model is G. Since having the same feature is the premise of analogy learning, we hope that the feature of x should have a correlation to the feature of $G(x) \in T$. In this case, we want to find a multivariate function f, which makes $f(x) = f(G(x))$.

To solve the above problems, we rebuild our generative model structure. As shown in Fig. 1, the reconstructed generative model G is composed of two parts, one is a multivariate function f, and the other is a generative function g.

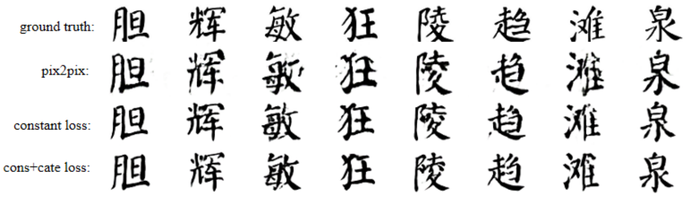


Fig. 2. The generation results of the model introduce / not introduce the constant loss function and category loss function.

After applying multivariate function f to generative model, we hope $\|f(x) - f(G(x))\|$ is as small as possible. Thus we have an additional component of the loss function of generative model G:

$$L_{\text{constant}} = E_{x \sim D_s} d(f(x), f(G(x))).$$

B. Category Loss

Before applying category loss function to our model, the model once can only learn one target style. Inspired by AC-GAN [9], we apply category loss function to our model, so that we can learn several font styles in one training and realize multi-style transfer model of text. To let the generative model learn multiple font styles at the same time, each font corresponds to a category label. The discriminative model not only determines the authenticity of the input sample, but also introduces another classifier to determine the category of the input sample. The loss function of the discrimination is composed of two parts: the log-likelihood of the correct source L_S, and the log-likelihood of the correct class L_C:

$$L_S = E\left[\log P\left(S = \text{real} \mid X_{\text{real}}\right)\right] + E\left[\log P\left(S = \text{fake} \mid X_{\text{fake}}\right)\right]$$

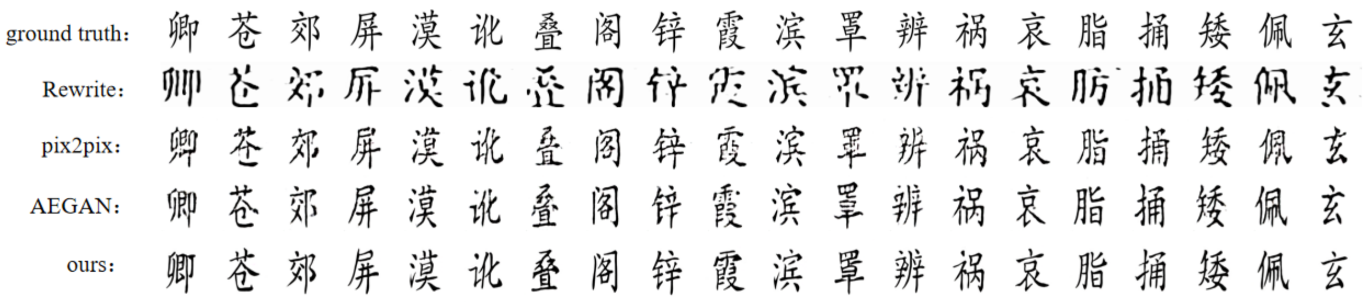


Fig. 3. The generation results of different models.

$$L_C = E\left[\log P\left(C = c \mid X_{\text{real}}\right)\right] + E\left[\log P\left(C = c \mid X_{\text{fake}}\right)\right].$$

For the generative model, we want the generative model to "deceive" the discriminative model. To make the discriminative model unable to discriminate the real and fake of the generated images, the component of the loss function of the generative model is as follows:

$$L_{\text{cheat}} = E\left[\left(\log P\left(S = \text{fake} \mid X_{\text{fake}}\right)\right].$$

Furthermore, we also hope that the generated images can "deceive" the discriminative model, so that the discriminative model fail to determine whether the font type of the generated images is correct. Therefor, the other component of the loss function of the generated model is introduced as:

$$L_{\text{fake-category-loss}} = E\left[\left(\log P\left(C = \text{c} \mid X_{\text{fake}}\right)\right].$$

C. Muti-Style Transfer Model

The loss function of the discriminative model of the multi-style transfer model we designed is:

$$L_D = L_S + \alpha L_C.$$

The loss function L_S is to make the discriminative model learn to distinguish the authenticity of the input in the training process, while the loss function L_C is to make the discriminative model learn to distinguish the type of input in the training process. α is the super parameter of the model, and our objective in the training discriminative model is to make L_D as large as possible.

The loss function of the generative model of the multi-style transfer model we designed is:

$$L_G = L_{\text{cheat}} + \lambda L_{L1}(G) + \lambda_2 L_{\text{fake-category-loss}} + \lambda_3 L_{\text{constant}},$$

where the definition of $L_{L1}(G)$ is same as it in [10].

D. New Framework

In order to solve the problem that text style transfer cannot work in the absence of training data, this paper proposes a new framework, which is composed of GTPD model and the multi-style transfer model mentioned before. At present, the training of the existing text style transfer models requires about 1500 text samples of the target style. When the sample data of the target style is insufficient, by using the GTPD model, we firstly get the probability distribution of the font type based on a small amount of sample data. Then we reconstruct the training data set according to the probability distribution of the font type. Next, the GTPD model will be described in detail below.

The GTPD model is essentially a multi-classification model. Unlike the classification task, we don't want the GTPD model to tell us the type of font that the input belongs to. In fact, after the GTPD model is trained, the input to the GTPD model often does not belong to any font type in the sample space. Through the probability distribution of font type got by the GTPD model, we can see the relationship between target font and fonts in our sample space to a certain extent, that the output is the probability distribution. The formula is as follows:

$$y = \frac{1}{M}\sum_{i=1}^{M}\left(p_1^i, \quad p_2^i, \cdots, p_N^i\right),$$

where y represents the average probability distribution corresponding to M images input, and N represents the number of font types contained in the GTPD model's sample space. In general, after getting y, we take the font types of the three largest probability values to build the training data set. The data set constructed contains the above three fonts, and the quantity ratio is equal to the probability ratio in the probability distribution. Namely, $m : n : q = p_x : p_y : p_z$, where m, n and q are the numbers of characters of font x, y and z in the constructed training data respectively, and p_x, p_y and p_z are the probability of font x, y and z respectively.

We use a deep neural network to construct the GTPD model. As is known to all, the structure of deep neural network directly affects the performance of the model. We measure the quality of GTPD model by its classification accuracy on the test set.

E. Implementation Details

1) Multivariate Function *f*. The generative model of CGAN consists of an encoder and a decoder. The network structure of the generative model includes an encoder and a decoder. We use encoder to fit multivariate function *f*.

Encoder is used to do feature extraction, which is very suitable for fitting multivariate function *f*.

2) GTPD Model. The loss function of the GTPD model is cross entropy function. In the case of insufficient sample data, we use the existing fonts in the GTPD model sample space to reconstruct the data set as input to the style transfer model. The GTPD model sample space should contain font types as much as possible that have similar texture to the target style font.

IV. Experiments

In this section, we complete a large number of comparative experiments. Firstly, we verify the improvement of generative model consists of constant loss and category loss function. Then, we demonstrate the effectiveness of the multi-style transfer model through the comparison experiments of different models. Finally, in the absence of samples of the target style, we carried out a lot of experiments with the new frame and used it to imitate the works of ancient calligraphers.

A. Data Set

We collect 10 kinds of common fonts and some Chinese calligrapher fonts as our training data set. In our experiments, we all use font Song (宋体) as the standard font.

B. Constant Loss and Category Loss

We take font Song as the source font (content) and font Yan (颜体) as the target style font (style), and randomly select 1500 simplified characters from these two fonts to construct a paired data set. The training set and the verification set are constructed at a ratio of 9:1 respectively. The pix2pix model, the model with constant loss function only and the model with constant loss function and category loss function introduced are respectively trained using the constructed data set. As shown in Fig. 2, introducing constant loss function and category loss function to our model leads to better generation effect.

C. Comparison with Baseline Methods

In this subsection, we compare our method with the following baselines for text style transfer.

- Rewrite [3]: Rewrite is a simple top-down Convolution network with big convolution kernel size and 1× 1stride. The network is minimized by L1 loss and total variation loss.
- Pix2pix [10]: Pix2pix is a conditional GAN based image translation network, which adopts the skip connection to connect encoder and decoder. Pix2pix is optimized by L1 distance loss and adversarial loss.
- Auto-encoder guided GAN (AEGAN) [4]: AEGAN consists of two encoder-decoder networks, one for image transfer and another acting as an auto-encoder to guide the transfer to learn detailed stroke information.

For comparison, we take font Song as the source font (content) and font Kai (楷体) as the target style font (style), and randomly select 1500 simplified characters from these two fonts to construct a paired data set. The results are shown in Fig. 3.

The multi-style transfer model can generate the font of traditional Chinese characters by learning style from simplified Chinese characters. We randomly select 1500 simplified characters from different font libraries as training data. After training the model, we let the model to generate these types of traditional characters. The generated results are shown in Fig. 4. The simplified characters are already exist in the font library, and we generate the fonts of traditional characters corresponding to these simplified characters.

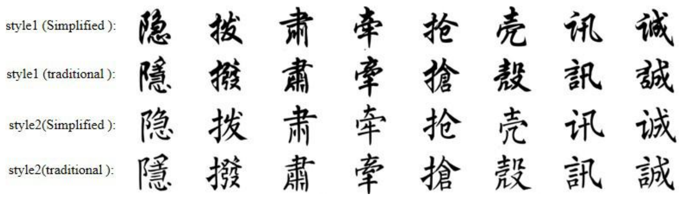

style1 (Simplified): 隐 拔 肃 牵 抢 壳 讯 诚
style1 (traditional): 隱 撥 肅 牽 搶 殼 訊 誠
style2(Simplified): 隐 拨 肃 牵 抢 壳 讯 诚
style2(traditional): 隱 撥 肅 牽 搶 殼 訊 誠

Fig. 4. The generation results of traditional Chinese characters by learning style from simplified Chinese characters.

D. New Framework

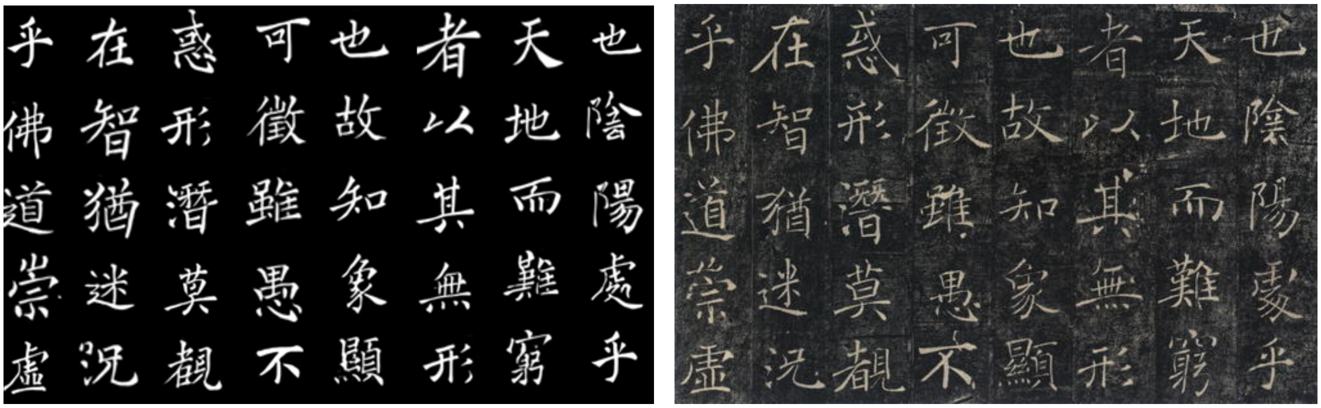
In the case of insufficient samples of the target style, we use the new framework to achieve text style transfer, and generate arbitrary characters of the target style. Firstly, we should train the GTPD model. We randomly select 2500 words from 13 common fonts (方正兰亭超细黑简体、方正舒体、毛泽东手写体、黑体、中易楷体、隶书、幼圆字体、华文楷体、华文宋体、华文行楷、手写体、李旭科书法、柳楷简体) as the training data of the GTPD model. The output of a GTPD model is a vector of 13 dimensions, each of which corresponds to a font type. A value per dimension represents the probability that the input character belongs to that font type.

After training the GTPD model, we randomly select several characters from Suiliang Chu's calligraphy work 《雁塔圣教序》 as input to the GTPD model (font Suiliang Chu is not one of the sample space fonts). Based on the output of GTPD model, we reconstruct the training set of the text style transfer model. Then we generate the whole calligraphy work based on the constructed training set. As shown in Fig. 5, the left is the characters of Suiliang Chu generated by using the new frame, the right comes from the network, which is the real work of Suiliang Chu.

V. Evaluation and Discussions

We evaluate our proposed method as well as other baselines on the database we collected.

1) Effect of the constant loss function and category loss function: Fig. 2 shows the improvement of the model by introducing constant loss. It can be seen from the results that, after constant loss and category loss is introduced, the generated characters are more similar to the target



(a) generated work

(b) real work

Fig. 5. The text generated by the new frame and real work.

characters in details. After further observation, it can be found that there are some polluted pixel points in the image generated by the pix2pix model. After introducing the constant loss function , these polluted pixel points disappear. After introducing category loss function, the model can not only learn a variety of different font styles at the same time, but also improve the generation effect.

2) Effect of the muti-style text style transfer: Multi-style text transfer model is the text style transfer model realized by introducing constant loss function and category loss function on the basis of the pix2pix model and adjusting the structure of the pix2pix model. As shown in Fig. 3, the multi-style transfer model has a good performance in the text style transfer.

3) Effect of the new framework: The new framework is used to solve the problem that the text style transfer models cannot work when the target style sample is insufficient. We propose the GTPD model to analyze the style characteristics by a small number of samples, and then reconstruct the data set for training the multi-style transfer model. It can be seen from Fig. 5, that the characters generated by using the new framework are very close to real characters.

VI. Conclusion and Future Work

In this paper, firstly, we propose a multi-style text transfer model with good performance. The new model can not only generate more realistic text, but also learn multiple different styles of fonts at the same time. In addition, the new model can also generate all traditional Chinese characters by learning the style of simplified Chinese characters. And then, we propose a new framework to solve the problem that the text style cannot work when the target style sample is insufficient. The new framework is composed of GTPD model and multi-style transfer model. The experimental results verify the effect of the new framework.

In fact, everyone's writing has a unique style, all belongs to a kind of font. Experiments in this paper show that each font corresponds to a different probability distribution of font type. So how to identity the handwritings through the probability distribution of font type is one of the future work.

References

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

[2] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, et al., "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018.

[3] Rewrite, "https://github.com/kaonashi-tyc/ rewrite," .

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

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

[6] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, and Zongming Guo, "Controllable artistic text style transfer via shape-matching gan," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[7] Anna Zhu, Xiongbo Lu, Xiang Bai, Seiichi Uchida, and Shengwu Xiong, "Few-shot text style transfer via deep feature similarity," *IEEE Transactions on Image Processing*, vol. PP, no. 99, pp. 1–1, 2020.

[8] Yaniv Taigman, Adam Polyak, and Lior Wolf, "Unsupervised cross-domain image generation," *arXiv preprint arXiv:1611.02200*, 2016.

[9] Augustus Odena, Christopher Olah, and Jonathon Shlens, "Conditional image synthesis with auxiliary classifier gans," in *International conference on machine learning*, 2017, pp. 2642–2651.

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