Applied Human Factors and Ergonomics (AHFE2025), Vol. 199, 2025, 2146–2155
https://doi.org/10.54941/ahfe1007026

A Structure Aware GAN-Based for Ancient Chinese Calligraphy Style Transfer

Meng-Zhe Cai[1], Der-Lor Way[2], and Zen-Chung Shih[1]

[1]Institute of Multimedia Engineering, National Yang Ming Chiao Tung University, Hsinchu, Taiwan
[2]Department of New Media Art, Taipei National University of the Arts, Taipei, Taiwan

ABSTRACT

Chinese calligraphy is an artistic writing style with a diversity of structures and strict rules governing the shape of each stroke. The automatic generation of Chinese calligraphy requires two elements to be accounted for: stroke shape and structural correctness, the latter of which is especially challenging. Many existing approaches to Chinese calligraphy generation require manual intervention. Developers must also ensure that all characters in a font have the same features. This study presents a framework for generating Chinese calligraphic characters based on the works of famous ancient calligraphers. To ensure structural correctness, an inverse mapping correction was developed. To address the adverse effects of nonstandard character forms, a monitoring mechanism was also incorporated into the framework. Several sets of generated characters were used to validate the effectiveness of the proposed framework.

Keywords: Chinese calligraphy, Style transfer, Generated character, Regular script, Deep learning

INTRODUCTION

The development of convolutional neural networks (CNNs) has enabled automatic font generation, and studies have sought to refine CNNs for such a purpose. Studies incorporating deep neural networks have generated whole font sets for alphabetic languages. However, only a few studies have generated fonts in the style of ancient Chinese calligraphy, primarily focusing on developing strokes and radicals. Chinese brush calligraphy is an artistic style in which each character has its own meaning. Calligraphers commonly write some Chinese characters in nonstandard forms due to idiosyncrasies in their handwriting habits.

Calligraphic characters are composed of structures and stroke shapes that are challenging work for neural networks to generate. Therefore, the present study developed a structure-aware framework to provide structural correction and consistency for character generation models. The framework yielded greater robustness to nonstandard character forms by normalizing the loss value of the generated calligraphy. We developed a method for multi-style

© 2025. Published by AHFE Open Access. All rights reserved.

calligraphy transfer based on an ancient Chinese calligraphy copybook. An inverse mapping network was used to automatically monitor the structural correctness of forward-generated Chinese calligraphic characters, including those that the ancient calligraphers referenced had never written. The contributions of our proposed method are as follows: (1) The proposed structure-aware framework can overcome the most challenging aspects of calligraphic font generation, improving the structural correctness of forward-generated characters for widespread application. (2) Our one-to-one transfer model overlooks nonstandard calligraphic character forms during training, normalizing the loss value of generated characters to reduce the influence of nonstandard forms on the overall font.

RELATIVE WORKS

Chinese Character Generation

Xia et al. (2013) proposed a multilevel framework for generating characters in the style of particular calligraphic works, which involved decomposing each character into radical and stroke components. For characters that were not written by calligraphers, the authors synthesized a new calligraphic character by looking up the corresponding radical composition in their index module. If the corresponding radical was not found in any of the selected calligraphic works, they used stroke contours to form the calligraphic radical. A new character was then generated through a multilevel process. However, this process resulted in the generation of some nonstandard characters due to inconsistent radical styles and inappropriate calligraphic strokes.

Lian et al. (2016) developed an automatic method for separately analyzing stroke shape and layout in handwritten fonts. However, their approach relies on the assumption that the reference font style does not require human intervention. Their proposed composition rules are not satisfactory for calligraphic fonts because a single character can be calligraphically rendered in a wide variety of ways. Li et al. (2019) extracted features of characters written by calligraphers to measure the topological similarity between calligraphic and generated characters. Jiang et al. (2019) created SCFont, a structure-guided framework for handwritten font generation, by stacking two neural networks to analyze character structures and stroke shapes separately. Although SCFont preserves appropriate character structures, it requires manually labeling stroke type data for all characters in a font.

MX-Font uses input from multiple experts and applies bipartite matching to do away with the dependence on component annotations (Park et al., 2021). XMP-Font includes a cross-modality encoder that functions for both stroke labels and character images (Liu et al., 2022). In addition, a CG-generative adversarial network (GAN) font generator can decouple the content of a character from its style at the component level using a component-aware module (Kong et al., 2022). These three methods require component labeling and spatial correspondence between style and content images for models to learn calligraphic styles (Tang et al., 2022).

Additionally, reference characters must be carefully selected to ensure high-quality results. By contrast, DG-Font incorporates a skip connection module for deformed characters without any labels (Xie et al., 2021). Nonetheless, the aforementioned generators encounter difficulties creating new fonts when faced with diverse source and target domains.

Guo et al. (2024) proposed an approach to Chinese calligraphy style transfer that simulates brush strokes and spacing with GAN, resulting in a visually and stylistically coherent font. Feng (2023) used CycleGAN for the digital conversion of stone and ink writing forms. To detect calligraphic characters in paintings, Kang et al. (2023) used a high-resolution net to extract the features of calligraphic characters, allowing machine learning based on high-resolution images.

Image-to-Image Transformation

Style transfer, where a particular artistic style is transferred onto a given source image, has become the focus of increasing scholarly attention. Several approaches to image-to-image transformation have been developed. Isola et al. (2023) leveraged adversarial and cycle consistency loss to train a network capable of transferring between source and target domains without the need for paired training data. The Pixel2style2pixel framework (Richardson et al., 2023) is based on an encoder network that creates a series of style vectors for a pretrained StyleGAN generator to produce high-quality results.

Gao et al. (2020) developed a forward-mapping framework to fine-tune GAN-based architecture for synthesizing calligraphy. Wu et al. (2020) created CalliGAN, a GAN-based approach to generating Chinese calligraphic characters, by decomposing characters into component sequences to obtain radical-level information that was then used to guide the generation process. Following this method, CalliGAN can achieve stable, high-quality calligraphy synthesis, treating component sequences as conditional input for each character. Wang et al. proposed a content fusion module where each character is characterized by a unique set of one-dimensional probability distributions for a given set of features (Wang et al., 2023). They also compared distribution distance with reconstruction loss to adjust character shapes.

PROPOSED METHOD

Our proposed framework generates calligraphic characters in the style of famous Chinese calligraphers, as indicated in Figure 1. Given a dataset X as the source domain and a dataset Y as the target domain, the framework can identify a function G that maps domain X to domain Y. A forward generator then transfers an input character into a particular calligraphic style S_R, where R represents the actual calligraphic output of a particular calligrapher. Three neural networks, including a forward discriminator, local discriminator, and inverse mapping generator, train on the results obtained through forward generation. The forward and local discriminators classify characters as real or generated (fake), allowing the

forward generator to produce characters with real calligraphic style S_R. The inverse mapping generator then transfers the characters in calligraphic style S_R back to the input source image, and corrects their structure. A reconstruction mechanism measures the difference between real and generated characters, guiding forward generation.

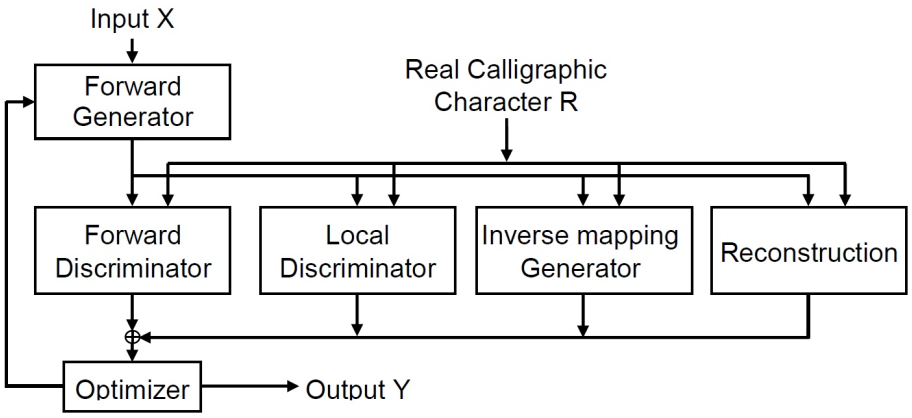

Figure 1: Architecture of the proposed framework.

Forward Generator Network

Forward generator G consists of several convolution layers (Krizhevsky et al., 2023) and several up-convolution layers (Dumoulin et al., 2016). Each convolution/up-convolution layer consisted of several filters. An image in the convolution/up-convolution layer is also known as feature transfer, except for the original input image. The output shape of a convolutional layer is influenced by input shape, kernel filter, zero padding, and strides. Feature maps in an up-convolution layer are stacked with feature transfers in the corresponding convolution layer. A skip connection stacks each corresponding feature maps channel-wise (Ronneberger et al., 2016). The connection is to stack the feature maps in channel-wise. The forward generator G contains an Instance Normalization layer (IN) (Ulyanov et al., 2016) and an activation function ReLU (Nair et al., 2016) after each convolution/up-convolution layer except for the last up-convolution layer. An IN normalizes each individual channel to zero mean and unit standard-deviation. Non-linear activation function ReLUs allow the deep neural network to create complex transformation between the input and output. Both IN and ReLU operations are defined as:

$$\mathrm{IN}(I) = \alpha\left(I - \mu(I) \cdot {}^{1}/_{\sigma(I)}\right) + \beta, \quad \mathrm{ReLU}\left(I_{i,j}\right) = \max(0, I_{i,j}) \qquad (1)$$

where I represents a feature map in the convolution and up-convolution layer. $I_{i,j}$ is the feature at the position (i, j) of I. the ReLU is applied to every feature of I. A real number $\sigma(I)$ is a standard-deviation of I. A real number $\mu(I)$ is the mean of I and the operation " $\cdot$ " is the scalar multiplication. The $\mathbf{1}$ is an all-ones matrix which has the same resolution as I. The learnable parameter

α is a real number used to adjust the standard-deviation of I. The learnable parameter β is a matrix where each element is equal to a real number β. It is used to adjust the mean of I.

Loss Function

A loss function is used to train the developed model, $G(I_a^X) = I_a^S$ as a transfer function from an input source image I_a^X corresponding to a character a from the font X to the output image I_a^S, where S denotes a calligraphic style. G represents the forward generator. The major goal is to produce I_a^S as close as possible to the same a written by a calligrapher. Both the content of I_a^X and I_a^S are the same character a. For each generated character of the forward generator, four loss terms were calculated, such as adversarial loss, local refinement loss, reconstruction loss, and inverse mapping loss. Therefore, the total loss function of the forward generator G is expressed as follows:

$$L(G) = \lambda_{adv} L_{adv} + \lambda_{local} L_{local} + \lambda_{inverse} L_{inverse} + \lambda_{recon} L_{recon} \quad (2)$$

where λ_{adv}, λ_{local}, $\lambda_{inverse}$, and λ_{recon} are tradeoff parameters between the loss functions. L_{adv} is the adversarial loss, which is calculated by the discrimination network; L_{local} is the local discriminator by AGIS-Net (Tang et al., 2022) to improve the quality of the generated character. $L_{inverse}$ is the proposed inverse mapping loss to penalize incorrect structures of the generated character. L_{recon} measures the difference between the generated character and the real calligraphic character.

Forward Discriminator

The architecture of forward discriminator is modified from PatchGAN (Isola et al., 2023). It consists of several convolution layers, Instance Normalization layers (IN), and activation functions ReLU. Zero-padding and non-unit strides are applied in convolution layers. A discriminator of PatchGAN delivers a set of scores instead of a single score. Each score represents the grade on a part of the input image. A receptive field is defined as the resolution of the region in the original input image that produces the feature. The receptive field of a feature is related to the kernel size and strides used in previous layers. Each feature in the final layer is a score which represents the grade on 47×47 pixels of the input image. The output of forward discriminator includes 256 (16×16) scores which are real numbers between 0 and 1. The receptive field of a score partially overlaps with receptive fields of neighboring scores.

In adversarial training (Goodfellow et al., 2014), the forward discriminator classifies that an input character is a real calligraphic character or a generated character. The scores of forward discriminator are high if the character is considered a real calligraphic character. Therefore, the scores of forward generator G is as high as possible from the forward discriminator D. A forward discriminator D delivers scores S_s for generated character. In the adversarial loss, mean-square error (MSE) is used to measure the distance between scores S_s and ideal scores. Thus, the adversarial loss value of the forward generator is $L_{adv} = \text{MSE}(\mathbf{1}, D(G(I_a^A)))$, where $D(G(I_a^A))$ is

equal to scores S_s and $G(I_a^A)$ represents the generated character. Ideal scores **1** is an all-one matrix and in 16×16 resolution. The MSE is defined as $\mathrm{MSE}(I_1, I_2) = \frac{1}{H}\frac{1}{W}\sum_{h=1}^{H}\sum_{w=1}^{W}[(I_1 - I_2)_{h,w}]^2$,where I_1 and I_2 are any two images. The H an W represent the height and width of the images, respectively. The $(I_1 - I_2)_{h,w}$ represents the pixel value at position (h, w) of $(I_1 - I_2)$.

A forward discriminator classifies that the input character is a real calligraphic character or a generated character. The discriminator D delivers high scores for the real calligraphic character and lower scores for the generated character. Thus, the discriminator D minimizes the following equation:

$$L(D) = \mathrm{MSE}(\mathbf{1}, D(I_a^R)) + \mathrm{MSE}(\mathbf{0}, D(G(I_a^A))) \tag{3}$$

where **0** is an all-zero matrix. The calligraphic character I_a^R is sampled from the real calligraphy R. Since the characters in the font X are usually more than characters in the real calligraphy R, another calligraphic character was randomly sampled if the calligraphic character I_a^R has not been written.

Local Discriminator

A local discriminator in AGIS-Net (Gao et al., 2019) is used to improve the quality of the generated character. The AGIS-Net randomly cut k patches from both generated character and calligraphic character. In order to produce clear contours of the generated character in cutting module. First, the Gaussian blur is applied to patches of real calligraphic character to get fuzzy data. Then, both fuzzy data and patches of the generated character are negative samples for local discriminator. Contrast, patches of real calligraphic character are positive samples for local discriminator. The scores S_r of local discriminator are high for the patches which are cut from real calligraphic character; S_{blur} and S_s are low for the fuzzy data and the patches, respectively. Thus, the local discriminator D_{local} minimizes the following equation:

$$\begin{aligned} L(D_{local}) = \frac{1}{k}\sum_{1}^{k}\mathrm{MSE}(\mathbf{1}, D_{local}(p_{real}^k)) + \frac{1}{k}\sum_{1}^{k}\mathrm{MSE}\left(\mathbf{0}, D_{local}\left(p_{blur}^k\right)\right) \\ + \frac{1}{k}\sum_{1}^{k}\mathrm{MSE}(\mathbf{0}, D_{local}(p_s^k)) \end{aligned} \tag{4}$$

where patches p_{real}^k and p_s^k are random cuts patches from the real calligraphic character and the generated character, respectively. A patch p_{blur}^k is a random cut from the fuzzy data. The **1** is an all-one matrix and **0** is an all-zero matrix.

A random cut k patches is used to get scores S_s in the local discriminator. The forward generator minimizes the distance between scores S_s and ideal scores **1**. Thus, the local refinement loss of the forward generator is expressed as follows:

$$L_{local} = \frac{1}{k}\sum_{1}^{k} \mathrm{MSE}(\mathbf{1},\ D_{local}\ (p_s^k)) \tag{5}$$

where the patch p_s^k is a random cut from the generated character.

EXPERIMENTAL RESULTS

The China Academic Digital Associative Library calligraphic system contains scans of many famous ancient calligraphic works, including 4275, 6727, 1354, and 2358 calligraphic characters written by Liu Gongquan, Yan Zhenqing, Zhao Mengfu, and Ouyang Xun, respectively. The present study sampled input source images from DFKai-SB font in the Windows font library. Each image was rendered in greyscale to represent a single Chinese character. The neural network was trained using an Adam optimizer with a learning rate of 10^{-3} for 300 training epochs with a batch size of 16. Loss function parameters for the forward generator were set as $\lambda_{adv} = 1$, $\lambda_{local} = 1$, $\lambda_{recon} = 50$, and $\lambda_{inverse} = 25$. An inverse mapping generator was used to monitor the forward generator, with inverse mapping loss disabled until convergence with $G_{inverse}$. Thus, $\lambda_{inverse}$ was set to zero for the first 150 training epochs. Four patches cut from each character in an epoch were fed into a local discriminator. For each image, one patch was cut from one of the four corners and the other three were cut randomly, meaning that the local discriminator did not need to account for all four image corners.

We first compared our character synthesis approach with that of Xia et al. (2013), who generated characters in the style of Liu Gongquan by predicting radical positions and stroke shapes. Several of the characters generated by Xia et al. exhibited inconsistent radical styles, as illustrated in Figure 2(a)(b), due to the selection of radicals from different works of calligraphy for character generation. In some instances, the distance between each radical was too far, as displayed in Figure 2(c). Moreover, some radicals consisted of unsuitable calligraphic strokes, as indicated in Figure 2(d)(e).

Figure 2: Comparison between our results and those of Xia et al. (2013).

We also compared our GAN-based approach to calligraphic character generation with that of Wu et al. (2020) using their training dataset.

We compared results for characters generated in style of Liu Gongquan. Wu et al.'s use of radical-level information to guide the generator in their GAN model resulted in improved quality and synthesis performance. However, some stroke details exhibited flaws, as demonstrated in Figure 3. Our comparison can enhance character generation stability and availability.

Figure 3: Comparison between our results and those of Wu et al. (2020).

Finally, we demonstrated the utility of our font by generating the famous poem "Yearning" in the style of Liu Gongquan, Yan Zhenqing, Zhao Mengfu, and Ouyang Xun, as displayed in Figure 4.

Figure 4: Enerated results for the poem "Yearning." (a) Liu Gongquan. (b) Yan Zhenqing. (c) Zhao Mengfu. (d) Ouyang Xun.

CONCLUSION AND FUTURE WORK

Previous efforts to automatically generate Chinese calligraphy fonts have provided promising results. However, existing approaches usually rely on labor-intensive manual intervention methods, such as radical decomposition. Some previous models have generated characters with flawed stroke shapes or incorrect structures. To address these problems, we designed an inverse mapping architecture that penalizes incorrect character structures, broadening the applicability of our framework. Our framework also incorporated a monitoring mechanism to decrease the adverse effects of nonstandard character forms. The monitoring mechanism normalized the L_1 loss of generated characters by multiplying an overlooked weight, allowing preservation of various Chinese calligraphy styles.

Our model was trained on the style of each calligrapher referenced separately, paying full attention to individual details. However, common

features between calligraphers were neglected. Future studies should design a framework that can learn the styles of all selected calligraphers simultaneously and take advantage of common features to supplement the lack of style information available in some datasets. Additionally, future studies could focus on generating other styles of Chinese calligraphy that remain challenging to model due to unique character structures, such as semi-cursive and cursive scripts.

ACKNOWLEDGMENT

Authors would like to thank the National Science and Technology Council of the Republic of China, Taiwan, for financially supporting this research under Contract No. NSTC 112-2410-H-119-008-MY2.

REFERENCES

Booher, H. R., Minninger, J. (2003) "Human systems integration in army systems acquisition", in: Handbook of human systems integration, Booher, Harold (Ed.). pp. 663–698.

Booher, Harold, ed. (2003). Handbook of human systems integration. New Jersey: Wiley.

Dumoulin V and Francesco V 2016 A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv*:1603.07285.

Gao Y and Wu J 2020 GAN-Based Unpaired Chinese Character Image Translation via Skeleton Transformation and Stroke Rendering. *Proc. of the AAAI Conference on Artificial Intelligence*. 34(01) track 1.

Gao Y, Gao Y, Lian Z Tang Y and Xiao J 2019 Artistic glyph image synthesis via one-stage few-shot learning. *ACM Transactions on Graphics* 38(6), 1–12.

Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio Y 2014 Generative adversarial nets. *arXiv*:1406.2661.

Guo J, Li J, Linghu K, B. Gao and Xia Z 2024 CCST-GAN: Generative Adversarial Networks for Chinese Calligraphy Style Transfer, *3rd International Conf. on Image Processing and Media Computing*, 62–69.

Feng R 2023 CycleGAN with auxiliary classifier for Chinese calligraphy style transfer and font classification *Theoretical and Natural Science* 18(1):167–173.

Isola P, Zhu J, Zhou T and Efros A 2017 Image-to-image translation with conditional adversarial networks, *In IEEE Conf. Comput. Vis. Pattern Recog*. 5967–5976.

Jiang Y, Lian Z, Tang Y and Xiao J 2019 SCont: Structure-guided chinese font generation via deep stacked networks, *Proc. of the AAAI Conference on Artificial Intelligence*. 33(01), 4015–4022.

Kang J, Wu Y and Xia Z 2022 Application of Deep Convolution Neural Network Algorithm in Detecting Traditional Calligraphy Characters, *International Conf. on Image Processing and Media Computing*, 12–16.

Kong Y, Luo C, Ma W, Zhu Q, Zhu S, Yuan N, and Jin L 2022 One-shot font generation via component based discriminator. *In IEEE Conf. Comput. Vis. Pattern Recog*. 13482–13491.

Krizhevsky A, Ilya S and Geoffrey E 2012 Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25, 1097–1105.

Li W, Song Y, and Zhou C 2014 Computationally evaluating and synthesizing Chinese calligraphy, *Neurocomputing* 135, 299–305.

Lian Z, Zhao B, and Xiao J 2016 Automatic generation of large-scale handwriting fonts via style learning, SIGGRAPH *Asia 2016 Technical Briefs*. 12, 1–4.

Lian Z, Zhao B, Chen X and Xiao J 2016 EasyFont: A style learning-based system to easily build your large-scale handwriting fonts. *ACM Transactions on Graphics*. 38(1), 1–18.

Liu W, Liu F, Ding F, He Q, and Yi Z 2022 XMPfont: Self-supervised cross-modality pre-training for few shot font generations. *In IEEE Conf. Comput. Vis. Pattern Recog*. 7905–7914, 2022.

Nair V and Geoffrey E. 2010 "Rectified linear units improve restricted Boltzmann machines." *Proc. of International Conf. on Machine Learning*, 807–814.

Park S, Chun S, Cha J, Lee B, and Shim H 2021 Few shot font generation with multiple localized experts. *In IEEE Conf. Comput. Vis. Pattern Recog*. 13880–13889.

Richardson E, Alaluf Y, Patashnik O, Nitzan Y, Azar Y, Shapiro S and Cohen-Or D 2021 "A styleGAN encoder for image-to-image translation," *In IEEE Conf. Comput. Vis. Pattern Recog*. 2287–2296.

Ronneberger O, Fischer P and Brox T 2015 U-net: Convolutional networks for biomedical image segmentation." *International Conf. on Medical image computing and computer-assisted intervention*. 9315.

Tang L, Cai Y, Liu J, Hong Z, Gong M, Fan M, Han J, Liu J, Ding E, and Wang J 2022 Few-shot font generation by learning fine-grained local styles. *In IEEE Conf. Comput. Vis. Pattern Recog*. 7895–7904.

Ulyanov D, Vedaldi A and Lempitsky V 2016 Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv*:1607.08022.

Wang C, Zhou M, Ge T, Jiang Y, Bao H and Xu W 2023 CF-Font: Content Fusion for Few-Shot Font Generation, *In IEEE Conf. Comput. Vis. Pattern Recog*. 1858–1867.

Wu S, Yang C and Hsu Y 2020 CalliGAN: Style and structure-aware Chinese calligraphy character generator. *arXiv preprint arXiv:* 2005.12500.

Xia Y, Wu J, Gao P, Lin Y and Mao T 2013 Ontology-based model for Chinese calligraphy synthesis, *Computer Graphics Forum*. 32(7), 11–20.

Xie Y, Chen X, Sun L and Yue Lu 2021 DG-font: Deformable generative networks for unsupervised font generation. *In IEEE Conf. Comput. Vis. Pattern Recog*. 5130–5140.