

Generating Handwritten Chinese Characters using CycleGAN

Bo Chang*

Qiong Zhang*

Shenyi Pan

Lili Meng

University of British Columbia

{bchang, qiong.zhang, shenyi.pan}@stat.ubc.ca, menglili@cs.ubc.ca

Abstract

Handwriting of Chinese has long been an important skill in East Asia. However, automatic generation of handwritten Chinese characters poses a great challenge due to the large number of characters. Various machine learning techniques have been used to recognize Chinese characters, but few works have studied the handwritten Chinese character generation problem, especially with unpaired training data. In this work, we formulate the Chinese handwritten character generation as a problem that learns a mapping from an existing printed font to a personalized handwritten style. We further propose DenseNet CycleGAN to generate Chinese handwritten characters. Our method is applied not only to commonly used Chinese characters but also to calligraphy work with aesthetic values. Furthermore, we propose content accuracy and style discrepancy as the evaluation metrics to assess the quality of the handwritten characters generated. We then use our proposed metrics to evaluate the generated characters from CASIA dataset as well as our newly introduced Lanting calligraphy dataset.

1. Introduction

Chinese characters have been used continually over three millennia by more than a quarter of the world's population [24]. The handwriting of Chinese character has long been one of the most fundamental skills in education, employment, communication, and everyday life in East Asia. For a long time, good handwriting or calligraphy has been considered not only as an artistic expression of language, but also as the supreme visual art as a means of self-expression and cultivation. As an example, Figure 1 shows some of our generated calligraphic characters. These aesthetically pleasing calligraphic works usually needs years of dedication and practice.

In contrast to phonological languages that have very limited number of letters such as English, Chinese has more than 80,000 logographic characters. Therefore, it is more



Figure 1: **Generated Chinese characters in Wang Xizhi's style.** The characters (from top to bottom) in the first column is the source SIMKAI font, the second and the third columns are in the Wang Xizhi's style with each column generated from DenseNet-5 and ResNet-6 CycleGAN. These four characters are not in the Lanting calligraphy dataset. The four characters stand for χ^2 distribution in Chinese.

challenging to design a personalized Chinese font than phonological languages. For example, only 26 letters need to be designed for a personalized English font while for Chinese at least 3,000 most commonly used characters need to be designed. To meet the demands of designing personalized Chinese font, methods that can automatically generate characters with personalized handwritten style based on a relatively small set of training characters are needed.

Although handwritten Chinese character generation is not as widely studied as character recognition, there are

* Authors contributed equally.

still approaches proposed for handwritten Chinese character generation. Most previous works rely on the hierarchical representation of simple strokes [27, 26, 17]. They decompose Chinese characters into strokes and then combine strokes to mimic the personalized writing style. As a result, such methods only focus on local representations of the characters rather than the overall style as a whole, and thus need to adjust the shapes, sizes, and positions of the strokes for every new character. In contrast, zi2zi [25] learns to transform fonts using pix2pix [9] with paired character images as the training data. However, in the task of generating handwritten Chinese characters, it is difficult to obtain a large set of paired training examples since it is infeasible to ask the user to write a large number of characters. The handwriting samples are also often isolated from a user’s writing, without knowing the true labels of the characters. Further, even the same character written by a user varies every time, which makes it more important to learn the overall style instead of mimicking every single character. Therefore, it is more appropriate to use unpaired Chinese characters instead of paired data for the handwritten Chinese generation problem. Figure 2 provides an example of paired and unpaired training data. Paired training data contain images of the same character in both fonts, while unpaired data do not necessarily contain the same set of characters.

In this work, we formulate the Chinese handwritten character generation as a problem that learns a mapping from an existing printed font to a personalized handwritten style. We further propose a method based on unpaired image-to-image translation to solve this problem. Our main contributions are:

- We propose to generate Chinese characters in a personalized handwritten style using DenseNet CycleGAN.
- We propose content accuracy and style discrepancy as the evaluation metrics to assess the quality of the generated characters.
- We demonstrate the efficacy of our proposed method on both CASIA dataset [16] and our newly introduced Lanting calligraphy dataset.

2. Related work

2.1. Chinese character handwriting generation

Chinese character generation has been studied since the beginning of the digital age [20]. In the literature, Chinese character generation is mainly formulated as either an artistic calligraphy, typography generation problem [27, 25, 28, 14], or a personal handwriting generation problem [26, 15, 22]. Most previous works rely on the hierarchical representation of simple strokes [27, 26]

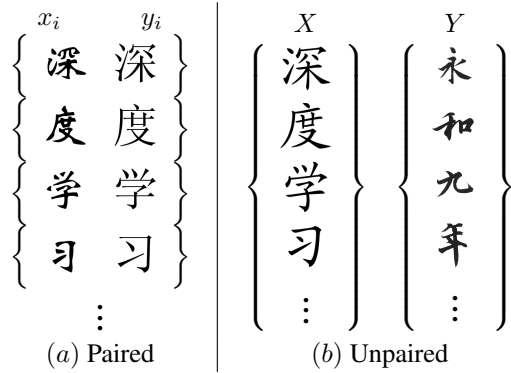


Figure 2: **Paired and unpaired training data.** (a) Paired training data consists of training examples $\{x_i, y_i\}_{i=1}^N$, where there exists correspondence between x_i and y_i . (b) We consider unpaired training data, where a source set X and a target set Y exist, with no matching information for x_i and y_i .

as the basis to represent Chinese characters. For example, StrokeBank [32] decomposes Chinese characters into a tree of components. FlexiFont [19] scans and processes the camera-captured handwritten character images, and then formats these characters as a personalized TrueType font file. Automatic shape morphing [14] first generates the shape template for every character, then decomposes two given Chinese characters into strokes to establish an accurate correspondence between strokes to achieve non-rigid point set registration. More recently, awesome typography [28] explores the problem of generating special-effects for the typography, and exploits the statistics on the high regularity of the spatial distribution for text effects to guide the synthesis process. Zi2zi [25] considers each Chinese character as a whole and learns to transform between fonts with paired training data. However, in the task of generating personalized Chinese handwriting font, it is difficult to obtain a large set of paired training examples.

2.2. Image style transfer

Current image style transfer methods can be divided into two categories, namely descriptive neural methods based on image iteration and generative neural methods based on model iteration [10]. Descriptive neural methods transfer the style by directly computing the gradient with respect to the source image and updating pixels in the image iteratively, while generative neural methods first optimize a generative model and produces the styled image through a single forward pass.

Neural style [2] is one of the most widely used descriptive neural methods for reproducing the content of an image

with the style of another. It formulates style transfer as an optimization problem that combines texture synthesis with content reconstruction. Patch-based loss is added on top of content and style losses in [13, 1].

The drawback of descriptive neural methods is that the iterative updating algorithm only works for a single image, which makes it rather time-consuming if one would like to transfer the styles of many images. In contrast, generative neural methods are faster but usually generates poorer style transfer results.

Nonetheless, our problem of generating handwritten Chinese characters does not exactly fall into the neural style transfer domain. In particular, it is difficult to define the content loss between characters in different styles since the strokes can be very different in positions and angles. For example, although x_i and y_i in Figure 2(a) represent the same characters, the corresponding images differ drastically.

2.3. Generative adversarial networks

GANs. General adversarial networks (GANs) [4] are powerful generative models which have achieved impressive results in many computer vision tasks such as image inpainting [7] and image-to-image translation [31], as well as natural language processing tasks such as speech synthesis [18] and cross-language learning [11]. GANs formulate generative modeling as a game between two competing networks: a generator network produces synthetic data given some input noise and a discriminator network distinguishes between the generator’s output and true data. Formally, the game between the generator G and the discriminator D has the minimax objective:

$$\min_G \max_D \mathbb{E}_{x \sim \mathbb{P}_r} [\log D(x)] + \mathbb{E}_{z \sim \mathbb{P}_g} [\log(1 - D(G(z)))] \tag{1}$$

where $x \sim \mathbb{P}_r$ are samples from the input data and $z \sim \mathbb{P}_g$ are the random noise samples, $G(z)$ are the generated images using the neural network generator G , and $D(\cdot)$ gives the probability of an input being real.

cGANs and pix2pix. Unlike GANs which learn a mapping from a random noise vector to an output image, conditional GANs (cGANs) learn a mapping from a random noise vector to an output image conditioning on additional information. cGANs are capable of image-to-image translation since they can condition on an input image and generate a corresponding output image. Pix2pix[9] is a generic image-to-image translation algorithm using cGANs. It can produce reasonable results on a wide variety of problems. Given a training set which contains pairs of related images, pix2pix learns how to convert an image of one type into an image of another type, or vice versa.

Zi2zi. Zi2zi [25] uses GAN to transform Chinese characters between fonts in an end-to-end fashion, assuming no stroke label or any other auxiliary information which is usu-

ally difficult to obtain. The network structure of zi2zi is based on pix2pix with the addition of category embedding for multiple fonts. This enables zi2zi to transform characters into several different fonts with one trained model. Zi2zi uses paired Chinese characters of the source font and the target font as the training data. However, since it is impractical to obtain a large set of paired training examples for personalized handwritten Chinese character generation, zi2zi is not applicable to our problem.

CycleGAN. Cycle-consistent GANs (CycleGANs) learn the image translation without paired examples [31]. Instead, it trains two generative models cycle-wise between the input and output images. In addition to the adversarial losses, cycle consistency loss is used to prevent the two generative models from contradicting each other. The default generator architecture of CycleGAN is ResNet [5], while the default discriminator architecture is a PatchGAN classifier [9].

3. Our method

Our task is to generate handwritten Chinese characters using unpaired source and target fonts. It can be formulated as learning a mapping G from the source font X to the target font Y given training samples $\{x_i\}_{i=1}^N$ where $x_i \in X$ and $\{y_j\}_{j=1}^M$ where $y_j \in Y$. An example of the unpaired training examples can be found in Figure 2(b). Our objective is consistent with that of CycleGAN [31], which is to learn image/font transformations across domains without paired training examples.

The mapping $G : X \rightarrow Y$ can be seen as the generators in the GANs except that the input of the generator is an image from the source domain rather than a random noise. An example of the architecture of generator G is shown in Figure 3. The mapping for learning the styles works as follows: the information in the image from domain X is first encoded to a lower dimension via the encoder given in Table 1. Then the extracted features from the source domain go through a transfer module. The outputs of the transfer module can be considered as the extracted features from the target domain. Finally, the output of the transfer module is decoded via the decoder as shown in Table 1.

An adversarial discriminator D_G , which is a 70×70 PatchGAN [9], is used to assess the quality of the generated images in the target domain. The second mapping $F : Y \rightarrow X$ and the corresponding discriminator D_F can be defined similarly. The objective of the discriminator is to distinguish between true images from the target domain and fake images produced by the generator based on images from the source domain.

The loss function of CycleGAN contains two parts: adversarial losses and cycle consistency losses. On one hand, the adversarial losses aim to match the distribution of generated images to the data distribution in the target domain.

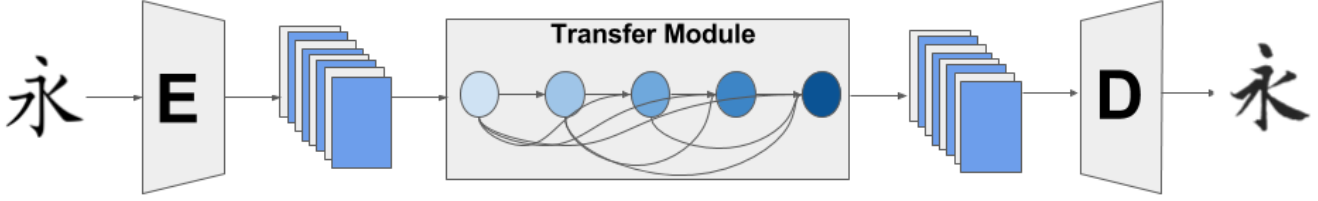


Figure 3: **The architecture of the generator in DenseNet-5 CycleGAN.** The information in the image in source style is first encoded to a lower dimensional space via the encoder E . Then the extracted features go through a transfer module, the outputs of which can be considered as the extracted features in the target style. Finally, the output of the transfer module is decoded via the decoder D .

Module	Specifications
Encoder	7×7 Conv-Norm-ReLU, 64 filters, stride 1
	3×3 Conv-Norm-ReLU, 128 filters, stride 2
	3×3 Conv-Norm-ReLU, 256 filters, stride 2
Transfer	Dense block, growth rate 256
	Dense block, growth rate 256
	Dense block, growth rate 256
	Dense block, growth rate 256
	Dense block, growth rate 256
Decoder	3×3 Deconv-Norm-ReLU, 128 filters, stride $1/2$
	3×3 Deconv-Norm-ReLU, 64 filters, stride $1/2$
	7×7 Deconv-Norm-ReLU, 3 filters, stride 1

Table 1: **The architecture and layer specifications of the encoder, transfer, and decoder modules of the generator in DenseNet CycleGAN.** Conv-Norm-ReLU represents a Convolution-InstanceNorm-ReLU layer. Deconv-Norm-ReLU represents a Fractional-strided-convolution-InstanceNorm-ReLU layer.

For G and D_G , the objective is

$$\mathcal{L}_{\text{GAN},G}(G, D_G) = \mathbb{E}_{x \sim p(x)} [\log(1 - D_G(G(x)))] + \mathbb{E}_{y \sim p(y)} [\log D_G(y)]. \quad (2)$$

The adversarial loss for F and D_F can be defined similarly. On the other hand, the cycle consistency losses ensure that the cyclic transformation is able to bring the image back to the original state. This is defined by

$$\mathcal{L}_{\text{cycle}}(G, F) = \mathbb{E}_{x \sim p(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p(y)} [\|G(F(y)) - y\|_1]. \quad (3)$$

The cycle consistency loss can be seen as a regularization and the strength of the regularization is controlled by λ . The full objective of CycleGAN is the summation of the adversarial losses for both mappings and the cycle consistency

losses:

$$\begin{aligned} \mathcal{L}_{\text{total}}(G, D_G, F, D_F) = & \mathcal{L}_{\text{GAN},G}(G, D_G) \\ & + \mathcal{L}_{\text{GAN},F}(F, D_F) \\ & + \lambda \mathcal{L}_{\text{cycle}}(G, F). \end{aligned} \quad (4)$$

The choice of the architecture of the transfer module is flexible. In the original CycleGAN [31], the transfer module contains several Residual Network (ResNet) blocks [5]:

$$\mathbf{x}_\ell = H_\ell(\mathbf{x}_{\ell-1}) + \mathbf{x}_{\ell-1}, \quad (5)$$

where $\mathbf{x}_{\ell-1}$ and \mathbf{x}_ℓ are the input and output of the ℓ -th ResNet block, and H_ℓ represents a composite function of operations such as batch normalization (BN) [8], rectified linear units (ReLU) [3] and convolution. The purpose of using an identity skip-connection to bypass the non-linear transformations is to facilitate gradient back-propagation.

Apart from the ResNet, the Dense Convolutional Network (DenseNet) [6] is among the most recent developments in convolutional neural networks. It further improves the information flow across blocks by connecting all the blocks directly with each other. The ℓ -th block receives the feature maps of all preceding blocks, $\mathbf{x}_0, \dots, \mathbf{x}_{\ell-1}$, as input,

$$\mathbf{x}_\ell = H_\ell([\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{\ell-1}]), \quad (6)$$

where $[\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_{\ell-1}]$ refers to the concatenation of the feature maps produced in blocks $0, \dots, \ell - 1$. DenseNets achieve state-of-the-art classification accuracy across several highly competitive datasets, using fewer parameters and less computation than ResNets.

Inspired by the design of DenseNets, we incorporate DenseNet blocks in the transfer module and propose a CycleGAN with DenseNet generator architecture (DenseNet CycleGAN) to generate handwritten Chinese characters. As shown in Table 1, the transfer module in our DenseNet CycleGAN consists of DenseNet blocks instead of ResNet blocks.

4. Experiments

In this section, we evaluate our proposed method on two publicly available datasets. Furthermore, we propose content accuracy and style discrepancy as complementary evaluation metrics in addition to visual appearance. Main results are shown in this section.

4.1. Datasets

CASIA-HWDB dataset. The Chinese handwriting database, CASIA-HWDB [16] is a widely used database for Chinese handwritten character recognition [30, 29]. This database contains samples of isolated characters and handwritten texts that were produced by 1020 writers using Anoto pen on papers.

In this study, we use the HWDB1.1 dataset from the CASIA-HWDB. It contains 300 files (240 in HWDB1.1 training set and 60 in HWDB1.1 test set). Each file contains about 3000 isolated gray-scale Chinese character images written by one writer, as well as their corresponding labels. The isolated character images are resized to 128×128 pixels. Other than resizing, no other data preprocessing method is performed.

For the task of generating handwritten characters, we use the file HW252 (1252-c.gnt) in the HWDB1.1 dataset as the target style, and SIMHEI font as the source style. SIMHEI is a commonly used Chinese font. Figure 4 shows the Chinese character “yong” in 5 different styles. The first two are printed fonts, and the last three are handwritten.



(a) SIMHEI (b) SIMKAI (c) Lanting (d) HW252 (e) HW292

Figure 4: **The character “yong” in 5 different fonts.** (a) SIMHEI; (b) SIMKAI; (c) character in Lanting calligraphy dataset; (d) handwritten character from HW252 (1252-c.gnt) in HWDB1.1; (e) handwritten character from HW292 (1292-c.gnt) in HWDB1.1.

Lanting calligraphy dataset. Chinese calligraphy is a form of aesthetically pleasing writing, which has been widely practiced in China and has been generally highly esteemed in the Chinese cultural sphere. In this work, we use the calligraphic work by Wang Xizhi as an example, who is generally regarded as the greatest Chinese calligrapher in history. Wang’s most famous work is the *Lantingji Xu*, which consists of 324 characters written in the *semi-cursive style*.

Each character in the *Lantingji Xu* is first scanned and isolated from the manuscript. They are further binarized

and denoised using median filtering. Finally, the characters are padded to square and resized to 128×128 pixels. The resulting dataset is referred to as the **Lanting calligraphy dataset**. The characters in the first column in Figure 7 are examples from the Lanting calligraphy dataset. The source font used for this task is the SIMKAI font, which is in the *regular style*. The dataset can be found here: <https://github.com/changebo/HCCG-CycleGAN/blob/master/lanting.zip>.

4.2. Performance metrics

Generative models usually lack objective evaluation criteria, which makes it difficult to quantitatively assess the quality of the images generated. To measure the performance of our handwritten character generation method on the CASIA-HWDB dataset, we propose two complementary evaluation metrics: the content accuracy and the style discrepancy. Both evaluations are based on a pre-trained network: the HCCR-GoogLeNet [30], which is a handwritten Chinese character classification model based on GoogLeNet [23].

Content accuracy. The HCCR-GoogLeNet model is trained using the CASIA-HWDB handwritten character database with 1,020 writers in total, including HW252. It achieved the state-of-the-art accuracy of Chinese character classification. Inspired by the idea of the Inception score [21], the pre-trained HCCR-GoogLeNet model can be used to evaluate the quality of the generated handwritten characters. The intuition is that if the generated characters are realistic, the pre-trained HCCR-GoogLeNet will also be able to classify the generated characters correctly. In our case, characters in the target style are generated from available images in the source style. Therefore, the true labels of the generated characters are known. If the characters generated can be accurately classified by pre-trained character recognition models, to some extent it indicates that the generative model is of high quality.

Table 2 shows the test accuracy of HCCR-GoogLeNet on the HW252 handwritten characters and the SIMHEI font characters as a baseline. The high recognition accuracy indicates that the HCCR-GoogLeNet accuracy is a reliable metric for quality measurement. However, the equally good performance on HW252 and SIMHEI implies that this metric measures the generation quality from a single perspective: it only assesses the content quality, not the style quality. Therefore, the recognition accuracy of HCCR-GoogLeNet on the generated handwritten characters is referred to as the content accuracy.

Style discrepancy. To measure the discrepancy in style between the true characters in the target domain and the generated characters, we borrow the style loss in neural style transfer algorithm [2]. The idea is to use the correlations between different filter activations at one layer as a

style representation. The feature correlations are given by the Gram matrix $G^\ell \in \mathbb{R}^{N_\ell \times N_\ell}$, where N^ℓ is the number of filters in the ℓ -th layer, and G_{ij}^ℓ is the inner product between the vectorized feature map i and j in layer ℓ :

$$G_{ij}^\ell = \sum_k F_{ik}^\ell F_{jk}^\ell. \quad (7)$$

The style discrepancy is thus defined as the root-mean-square difference between the style representations of the target characters and the generated characters. Lower discrepancy corresponds to better style quality. In our experiments, the input of Inception module 3 in HCCR-GoogLeNet is used as layer ℓ to calculate the style discrepancy.

We run two baseline experiments to get an approximate of the range of the style discrepancy. **(a)** The style discrepancy between two randomly and equally split subsets of the HW252 handwritten dataset. Since these two subsets are written by the same person and have the same style, the result represents the lower bound of the style discrepancy. *The style discrepancy lower bound is 503.77.* **(b)** The style discrepancy between HW252 and SIMHEI. This is the style difference between the source style and the target style. It thus represents the most possible disagreement in style, and it measures the style quality of a trivial identity style transfer model. Therefore it can be regarded as an upper bound of the style discrepancy. *The style discrepancy upper bound is 3006.03.*

4.3. Implementation details

In the experiments, we consider two types of transfer modules: ResNet with 6 blocks (ResNet-6) and DenseNet with 5 blocks (DenseNet-5). The DenseNet-5 transfer module has roughly the same number of parameters as the ResNet-6 transfer module.

The only preprocessing procedure we used is to resize the training images to 128×128 pixels; no other preprocessing methods (e.g. crop and flip) are used. For all the experiments, the regularization strength is set to $\lambda = 10$, and the Adam optimizer [12] with a batch size of 1 is used. The learning rate is set to 0.0002 for the first 100 epochs

	Top-1 accuracy	Top-5 accuracy
HW252	98.03%	99.84%
SIMHEI	94.22%	99.60%

Table 2: The top-1 and top-5 test accuracy of HCCR-GoogLeNet on the HW252 handwritten characters and the SIMHEI font characters. The equally good performance on HW252 and SIMHEI implies that this metric only measures the content quality.

and then linearly decays to 0 over the next 100 epochs. The number of iterations in each epoch in the experiments is the larger number of the training examples in the two styles.

4.4. Handwritten characters results

We use SIMHEI font as the source style and handwritten characters in HW252 as the target style. SIMHEI and HW252 are both split to unpaired training and validation sets. In real applications, we would like the users to only write a few Chinese characters, based on which the remaining handwritten characters can be generated with his/her personal style using our proposed method. Therefore, the goal of this experiment is to use small training sets to train a CycleGAN model, and perform style transfer on the validation sets. Let r_A be the split ratio of HW252, that is, r_A of the characters in HW252 are randomly assigned to the training set, and the remaining characters are in the validation set. Similarly, r_B denotes the split ratio of SIMHEI. Both r_A and r_B take values in $\{5\%, 10\%, 15\%, 30\%\}$, which gives 16 combinations. Table 3 and 4 show the top-5 content accuracy and style discrepancy for ResNet-6 and DenseNet-5, respectively. The results indicate that both content and style quality improves with the number of increasing training data. In particular, when r_A and r_B are greater than 10%, the content accuracy is always greater than 80%. The performance of ResNet-6 and DenseNet-5 are comparable. Furthermore, since the style discrepancy ranges between 503.77 and 3006.03, the style discrepancies are on the low end of the spectrum.

Figure 5 shows the generated handwritten characters using DenseNet-5 and ResNet-6, as well as the source and target styles. The background of the generated characters is clear and the contents are perfectly recognizable. The style of the strokes is noticeably different from that of the source font. The composition of the radicals and the blurry boundaries of the strokes highly resemble the writer’s handwriting style. Figure 6 shows a famous Chinese poem “On the Stork Tower” composed of characters generated by ResNet-6 CycleGAN. All the characters are clearly recognizable with personalized style.

4.5. Calligraphy results

We use SIMKAI as the source font for the calligraphy generation in Wang Xizhi’s style. Overall, DenseNet marginally outperforms ResNet in style. Figure 7 shows the ground truth as well as the generated calligraphy characters of the first four characters in the Lanting calligraphy dataset given unpaired training data. It can be seen that CycleGAN with both DenseNet and ResNet generator captures the overall writing style of Wang Xizhi and generates reasonable outputs. Compared with ResNet, DenseNet tends to generate fewer cases of missing strokes (e.g., the dot in the first character) or extra strokes (e.g., the extra throw-away

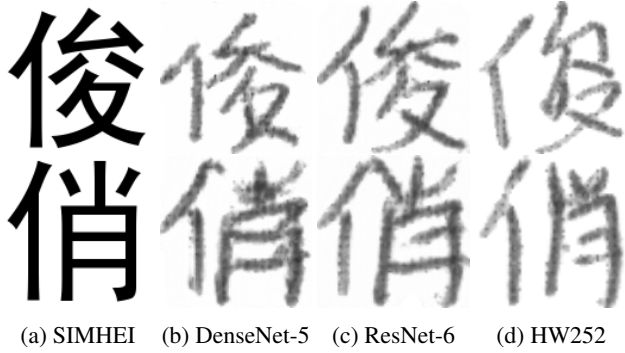


Figure 5: **The handwritten Chinese characters.** (a) The source characters in SIMHEI font; (b) the generated characters in HW252’s style using ResNet-6; (c) the generated characters in HW252’s style using DenseNet-5; (d) the ground truth characters in HW252’s style.

in the second character). Nevertheless, both DenseNet and ResNet generators fail to learn certain features of Wang’s semi-cursive style. For example, for the character “he” in the second row, the strokes “throw-away” (falling leftwards) and “press-down” (falling rightwards) are usually simplified to a single “break” (a change in direction) stroke in semi-cursive script. This feature is not learned by the CycleGAN model: both DenseNet and ResNet generate separate “throw-away” and “press-down” strokes.

The last two columns in Figure 1 are also examples of the generated Chinese calligraphy in Wang Xizhi’s style. These characters are not in the original Lanting calligraphy dataset and the generated characters show satisfactory performance.

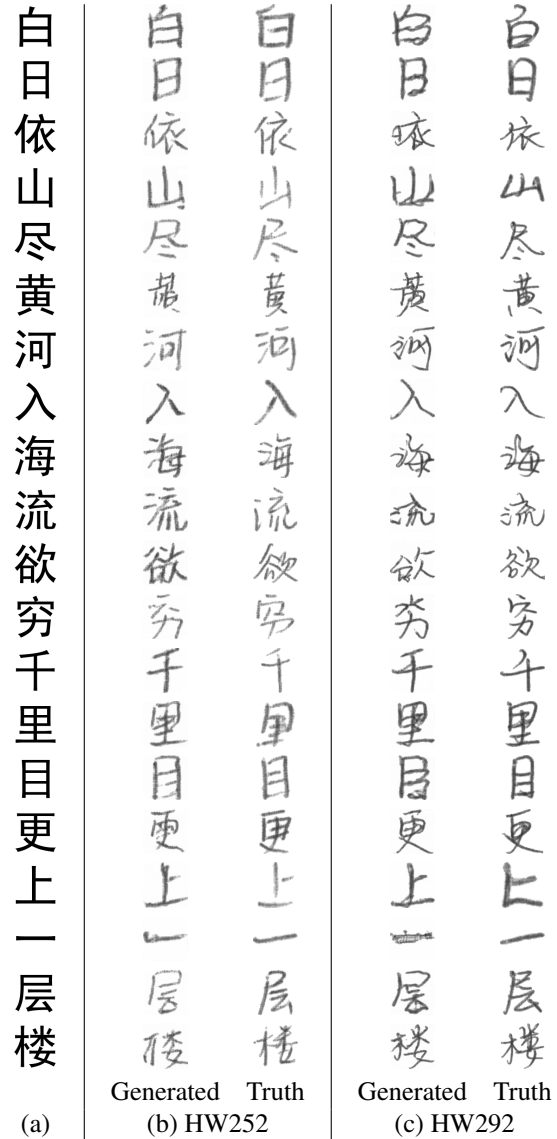


Figure 6: **A famous Chinese poem entitled “On the Stork Tower” generated by our proposed method.** (a) SIMHEI is the source style; (b) and (c) are two handwritten styles. The generated characters are clearly recognizable with personalized style.

4.6. Comparison with neural style transfer

We compare our method with neural style transfer [2] by using VGG-19 as the pre-trained network. We use layer `relu4_2` of VGG-19 for content loss, and layer `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1`, and `relu5_1` for style loss. Two characters in SIMHEI font are used as the content images, and 30% of the characters are randomly selected from the HW252 dataset as style images. This is the same

ResNet-6		r_A			
		5%	10%	15%	30%
r_B	5%	83.85% 742.93	72.83% 786.83	56.32% 970.98	49.31% 1225.86
	10%	77.92% 836.46	87.30% 758.73	86.98% 697.79	80.88% 702.45
	15%	78.31% 811.75	81.17% 700.12	89.41% 709.78	86.59% 530.31
	30%	82.08% 885.91	86.34% 752.86	89.19% 761.41	86.49% 613.44

Table 3: **ResNet-6 results.** Top-5 content accuracy and style discrepancy on HW252 and SIMHEI font. r_A and r_B represent the split ratio of HW252 and SIMHEI respectively.

DenseNet-5		r_A			
		5%	10%	15%	30%
r_B	5%	76.67%	58.90%	60.75%	15.78%
		839.22	1075.58	1326.33	2016.11
	10%	84.02%	83.01%	82.39%	84.46%
		840.00	761.60	844.01	576.14
	15%	84.21%	82.33%	90.47%	82.14%
		792.23	725.15	723.82	700.19
	30%	81.43%	90.03%	89.04%	88.55%
		792.89	703.96	759.20	594.876

Table 4: **DenseNet-5 results.** Top-5 content accuracy and style discrepancy on HW252 and SIMHEI font. r_A and r_B represent the split ratio of HW252 and SIMHEI respectively.



Figure 7: **The Chinese calligraphy characters in Lanting calligraphy dataset.** (a) The ground truth characters; (b) the generated characters in Wang Xizhi’s style using DenseNet-5; (c) the generated characters in Wang Xizhi’s style using ResNet-6. Note that for the character “he” in the second row, the strokes “throw-away” and “press-down” in the red circle are simplified to a single “break” stroke in the ground truth characters, but they are generated as separate strokes by DenseNet and ResNet CycleGAN.

as the setting of $r_A = 30\%$. The generated results by neural style transfer are shown in Figure 8. It can be seen the style

of the generated images is almost identical to the source font, implying that the learned transform is trivial. Furthermore, the background of the generated images is blurry. The overall visual quality is worse than those generated by our method.

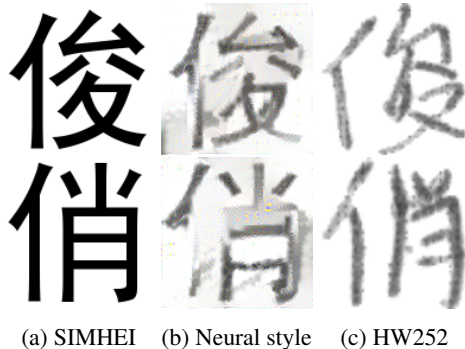


Figure 8: **The results of neural style transfer.** (a) The source characters in SIMHEI font; (b) characters generated by neural style transfer; (c) the ground truth characters in HW252’s style. The style of the generated images is almost identical to the source font and the background is blurry.

5. Conclusion

In this work, we formulate the Chinese handwritten character generation problem as learning a mapping from an existing printed font to a personalized handwritten style. We present DenseNet CycleGAN to solve this problem, and our method uses DenseNet as part of the CycleGAN generator to improve the generation quality. The proposed method is compared with the original CycleGAN with ResNet blocks and Neural style transfer. We evaluate the proposed method on the CASIA dataset and the newly introduced Lanting calligraphy dataset. Furthermore, we propose two novel Chinese character generation performance evaluation metrics content accuracy and style discrepancy for quantitatively assessing the quality of generated Chinese handwritten character. Extensive experimental results demonstrate the efficacy of our method, showing superior or on-par performance.

Acknowledgements. The authors would like to thank NVIDIA for the hardware donation from GPU Grant Program.

References

- [1] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016.
- [2] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

- [3] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *AISTATS*, 2011.
- [4] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [6] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *CVPR*, 2017.
- [7] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (TOG)*, 2017.
- [8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- [9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2016.
- [10] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song. Neural style transfer: A review. *arXiv preprint arXiv:1705.04058*, 2017.
- [11] S. Joty, P. Nakov, L. Màrquez, and I. Jaradat. Cross-language learning with adversarial neural networks. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 226–237, 2017.
- [12] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [13] C. Li and M. Wand. Combining markov random fields and convolutional neural networks for image synthesis. In *CVPR*, 2016.
- [14] Z. Lian and J. Xiao. Automatic shape morphing for chinese characters. In *SIGGRAPH Asia 2012 Technical Briefs*, page 2. ACM, 2012.
- [15] J.-W. Lin, C.-Y. Hong, R.-I. Chang, Y.-C. Wang, S.-Y. Lin, and J.-M. Ho. Complete font generation of chinese characters in personal handwriting style. In *IPCCC*. IEEE, 2015.
- [16] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. Casia online and offline chinese handwriting databases. In *ICDAR*. IEEE, 2011.
- [17] P. Liu, S. Xu, and S. Lin. Automatic generation of personalized chinese handwriting characters. In *Digital Home (ICDH), 2012 Fourth International Conference on*, pages 109–116. IEEE, 2012.
- [18] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.
- [19] W. Pan, Z. Lian, R. Sun, Y. Tang, and J. Xiao. Flexifont: a flexible system to generate personal font libraries. In *Proceedings of the 2014 ACM symposium on Document engineering*, pages 17–20. ACM, 2014.
- [20] R. Plamondon, W. Guerfali, and X. Li. The generation of oriental characters: new perspectives for automatic handwriting processing. *International journal of pattern recognition and artificial intelligence*, 1998.
- [21] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *NIPS*, 2016.
- [22] D. Sun, T. Ren, C. Li, J. Zhu, and H. Su. Learning to write stylized chinese characters by reading a handful of examples. *arXiv preprint arXiv:1712.06424*, 2017.
- [23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.
- [24] I. Taylor and M. M. Taylor. *Writing and Literacy in Chinese, Korean and Japanese: Revised Edition*, volume 14. John Benjamins Publishing Company, 2014.
- [25] Y. Tian. zi2zi: Master chinese calligraphy with conditional adversarial networks. <https://github.com/kaonashi-tyc/zi2zi>, 2017.
- [26] S. Xu, T. Jin, H. Jiang, and F. C. Lau. Automatic generation of personal chinese handwriting by capturing the characteristics of personal handwriting. In *IAAI*, 2009.
- [27] S. Xu, F. C. Lau, W. K. Cheung, and Y. Pan. Automatic generation of artistic chinese calligraphy. *IEEE Intelligent Systems*, 2005.
- [28] S. Yang, J. Liu, Z. Lian, and Z. Guo. Awesome typography: Statistics-based text effects transfer. *CVPR*, 2017.
- [29] X.-Y. Zhang, Y. Bengio, and C.-L. Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, 2017.
- [30] Z. Zhong, L. Jin, and Z. Xie. High performance offline handwritten chinese character recognition using googlenet and directional feature maps. In *ICDAR*. IEEE, 2015.
- [31] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *ICCV*, 2017.
- [32] A. Zong and Y. Zhu. Strokebank: Automating personalized chinese handwriting generation. In *AAAI*, 2014.