國立成功大學

資訊工程研究所

碩士論文

使用生成對抗網路基於筆畫拆解之中文風格字體轉換

Transformation of Stylized Handwritten Chinese Characters by Generative Adversarial Network Using Stroke Decomposition

研究生：田亦心　　Student: Yi-Hsin Tien

指導教授：鄭憲宗　　Advisor: Sheng-Tzong Cheng

Institute of Computer Science and Information Engineering

National Cheng Kung University,

Tainan, Taiwan, R.O.C

July 2021

中華民國一百一十年七月

國立成功大學

碩士論文

使用生成對抗網路基於筆畫拆解之中文風格字體轉換

Transformation of Stylized Handwritten Chinese Characters by Generative Adversarial Network Using Stroke Decomposition

研究生：田亦心

本論文業經審查及口試合格特此證明

論文考試委員：

詹寶珠　謝孫源

范峻逸　龔旭陽

指導教授：　鄭憲宗

單位主管：

單位主管是否簽章授權由各院、系(所、學位學程)自訂

中 華 民 國 110 年 7 月 20 日

An Attention-Based Extracting Multivariate Features and Time Method for Time Series Forecasting Problem

By

Chia-Hsuan Lin

A thesis submitted to the graduate division.

In partial fulfillment of the requirements for the degree of

Master in Computer Science and Information Engineering,

National Cheng Kung University, Tainan, Taiwan, R.O.C.

July 20, 2021

Approved by：

Sok-Ian Soo　　　　　　Hung-Chang Hsiao

Hsu-Yang Kung

Advisor：

Chairman：Yeimkuan Chang

使用生成對抗網路基於筆畫拆解之中文風格字體轉換

田亦心* 鄭憲宗**
國立成功大學資訊工程研究所

摘要

對於設計師而言，設計中文字型除了需要手寫超過 6000 個字以上的常用字，還必須逐字調整，是件繁雜的工程。隨著人工智慧的發展，以生成對抗網路進行字體生成的技術，能夠迅速的幫助設計師完成大半工作，設計師只需要設計出基本的字型風格後，寫出少量的字，就可以藉由電腦得到其他的字符。

在中文字的構成中，筆畫可以說是最基本的字符構成單位。本論文提出一個模型架構，藉由將筆畫順序加入生成對抗網路模型，將標準字型轉換成手寫特殊字型，完成更好的字型轉換；此外，更改現有的損失函數、加入自注意力機制、雙鑑別器，進一步改進了現有模型。其實驗結果並分別與現有的模型做比較，並達到現今先進模型之效能。為了增加本論文之真實應用面，我們還製作了一個字型轉換模組，可將模型生成的字符圖片，完整轉換成一組可供電腦使用的字體。

本論文主要有以下三項貢獻：提出手寫筆畫字型生成對抗網路模型，將標準中文字體轉換成手寫風格字型，並將筆劃概念加入模型中，增加模型之準確度；將許多最新的方法架構加入至模型中，提升模型之準確度；並將提出之方法實際運用，將生成出的字體製作成字型檔案，並提供介面讓使用者使用。

關鍵字 生成對抗網路、風格轉移

*作者

**指導教授

Transformation of Stylized Handwritten Chinese Characters by Generative Adversarial Network Using Stroke Decomposition

Yi-Hsin Tien* Sheng-Tzong Cheng**
Institute of Computer Science and Information Engineering
National Cheng Kung University

Abstract

Designing Chinese calligraphy has been a tricky task for designers for a long time since there are more than 6000 characters in Chinese. Thanks to the development of artificial intelligence, the techniques of generating new fonts with the deep generative adversarial network now can promptly help designers save their efforts. Nowadays, designers only need to design the basic amount of style fonts then get the rest of the characters by computing.

As we know, strokes are the fundamental components of Chinese characters. In other words, once we understand the features of strokes, we can generate a new character easily. Accordingly, this research proposes a model based on a deep generative adversarial network that can transfer standard Chinese characters into stylized ones for better character transformation. We added a serial of stroke encodings into the generative adversarial network to enhance the quality of generated images. Besides, the model improves the existing model in several aspects, including loss functions, new training methods, dual discriminator, and new architectures. Compared with the previous research, the experiment result demonstrates high-quality stylized Chinese characters images against other state-of-the-art methods. To make the study more applicable, we developed a new dataset of Chinese hand-written characters and the system to transfer standard Chinese font into hand-written stylized one which can be used for typing.

Keywords Generative Adversarial Network; Style Transfer; Chinese Calligraphy

*Author
** Advisor

ACKNOWLEDGMENT

碩班的生活，首先一定要感謝我的指導老師鄭憲宗老師，除了專業知識的指導外，老師看事情的敏銳度、做事情的態度都深深影響了我，讓我看見事情的不同面向；並提供許多實際執行專案的機會，包含中科院等計畫讓我有機會學習成長。也很感謝實驗室的夥伴，學長、學弟與同學們，實驗上有問題時，你們就是我的後盾；在做研究覺得煩悶的時光，轉身就可以與你們閒聊兩句，讓我的碩班生活總有許多閃亮的片刻。還有許多在成大時光遇到的朋友，你們豐富了我的生命，讓我知道世界是這麼廣闊美麗，就算只是一朝一會，每個遇見也都值得我感激。謝謝不離不棄的許多人，從國中就一直支持我的朋友，不管是研究、生活、工作，與你們討論事情總讓我相信不確定的未來也值得期待；謝謝高中認識的女孩們，我們各自努力，但累了就可以互相抱怨生活；謝謝大學交到的好朋友，工作、學業相關的問題總是麻煩你們了。還有最愛我的家人，謝謝你們總是支持與關心我，未來我會努力不造成世界負擔的！

準備踏入職場的現在，希望我能保持對世界的好奇心；努力學習；對事情負責任，努力當最好的自己。

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1. Introduction

Style transfer is an interesting domain in image processing. For designers, they want to give arts different expressions for the same content. With the development of deep learning technology, a generative adversarial network has become a new trend for style transfer since it is so powerful.

For Chinese calligraphy, it is one of the worth discussing problems among style transfer. There are more than 4000 Chinese characters for common communications and nearly 50000 characters in the Chinese dictionary. Otherwise, Chinese characters' structure is unique and complicated, it is difficult for humans to learn the Chinese characters' structures, not to mention a machine. Since Chinese characters' special architecture and huge amounts, designers must check every character and write every stroke by themselves.

Things may change because of artificial intelligence. In 2017, the first Chinese calligraphy transfer model is proposed. zi2zi[1] is a pix2pix[2] based generative adversarial network. It can transfer one style to multiple styles with paired images. Some models follow its footsteps. CycleGAN[3] has been a classical model for transfer learning, and it also has a good result. Moreover, it only acquired impaired images. Although previous models have provided great performance. Since Chinese characters' glyph is complex, some of generating characters are blurred or inscrutable. After that, more models are proposed. FontGAN[4] divides features into style representation and content representation, and it consists of two independent encoders to learn character glyph. CalliGAN [5] decomposes characters into components, with its low-level structure representation, it can generate Chinese characters with a low error rate.

Except for giving designers tools to make a new font of Chinese calligraphy, our method focuses on hand-written Chinese characters. Nowadays, people pursue idealization and specialty in our generation, so many demands for customized goods are increasing. Not only does our proposed model uses font-rendered character images as datasets like other existing methods, but we also use hand-written Chinese character images written by several subjects. Since everyone has his or her style of Chinese calligraphy, learning hand-written Chinese characters is a hard task. This model attempts to deal with this kind of problem.

Although some of the methods for Chinese calligraphy transfer can do the one-to-many task, we think it is not useful for applications because we need more images in one font for one-to-many transfer tasks than one-to-one tasks. Our proposed method made a trade-off between amounts of characters' images and exquisite results which can be made as a TrueType font file with high quality.

In this paper, we proposed an image-to-image translation model. We decompose Chinese characters into strokes, which is the lowest level structure for Chinese characters. Different from StrokeGAN [6] using a one-hot vector as stroke embeddings, we use a database-based model and encoder to decompose Chinese characters as a serial of strokes. A serial of strokes has different priorities in Chinese architecture. The main contributions can be summarized as follows:

(1) We propose a novel model for the generation of Chinese calligraphy, which contains stroke embeddings. Compared with existing papers using one-hot vectors, our embeddings are encoded with the order. By stroke embeddings, a generator can handle font transfer tasks well.

(2) Our proposed model is improved by the state-of-art architecture and loss functions. While using contextual loss and a re-senet network, the performance of the

model greatly improves. We also add the dual discriminator to focus on specific regions for helping a generator generating high-quality images.

(3) We provide new Chinese handwritten datasets for researchers, all the datasets can be download on github page. In addition, we do a module to transfer generated images to a TrueType font file which is convenient for users to make their font files, which means the users can make their own Chinese hand-written characters datasets by themselves easily.

Chapter 2. Related Work

2.1 Image-to-image Translation

An image-to-image translation is one type of vision and graphics problem, and its goal is to learn the mapping between an input image and a target image. Since 2015, image-to-image translation becomes a popular issue in artificial intelligence, which contains style transfer. A Neural Algorithm of Artistic Style [7] uses CNN as a feature extractor. This method uses two images as CNN input, one is for content, and the other is for style. Each feature map can be seen as content features and style features. Using these content representations and style representations, we can then compare output images and input images to get stylized images.

2.2 Generative Adversarial Network

At the same time, generative adversarial network [8] succeeds in a variety of tasks. Figure 1 shows the basic model of a generative adversarial network, and it is the foundation of many following models. Pix2pix is a cGAN[9] based model, and it seen images as a conditional input to generate high-quality images. While GAN is so powerful, the mode collapse problem is extremely troublesome. Wasserstein Generative Adversarial Network [10] solved this problem by proposing a clipping to restrict weight value. Dual discriminator generative adversarial net [1] also provides an idea to solve the mode collapse problem by combining Kullback-Leibler (KL) and reverse KL divergences into an objective function. In 2017, CycleGAN[3] is a type of generative adversarial network for unpaired images. It is famous for its artistic style transfer and then used in lots of domain transfer problems.

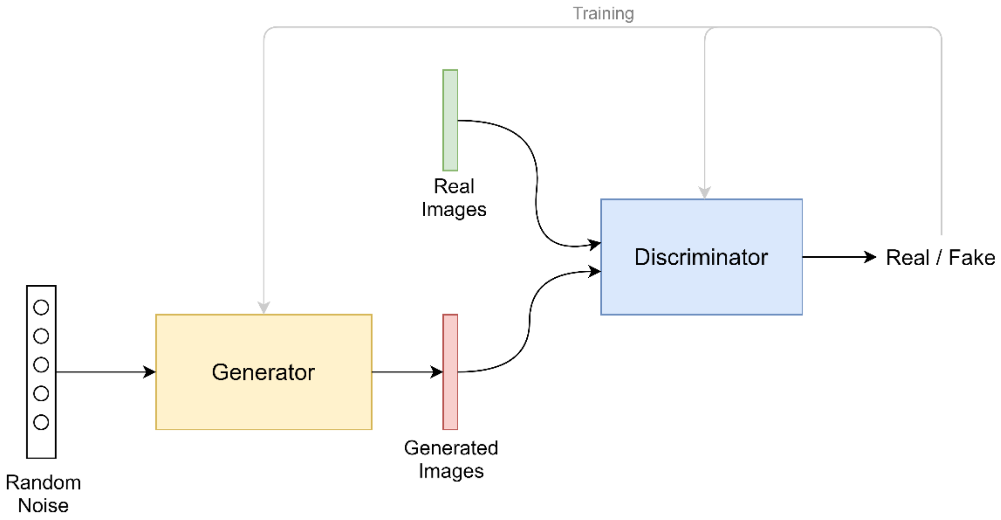


Figure 1 Generative Adversarial Network

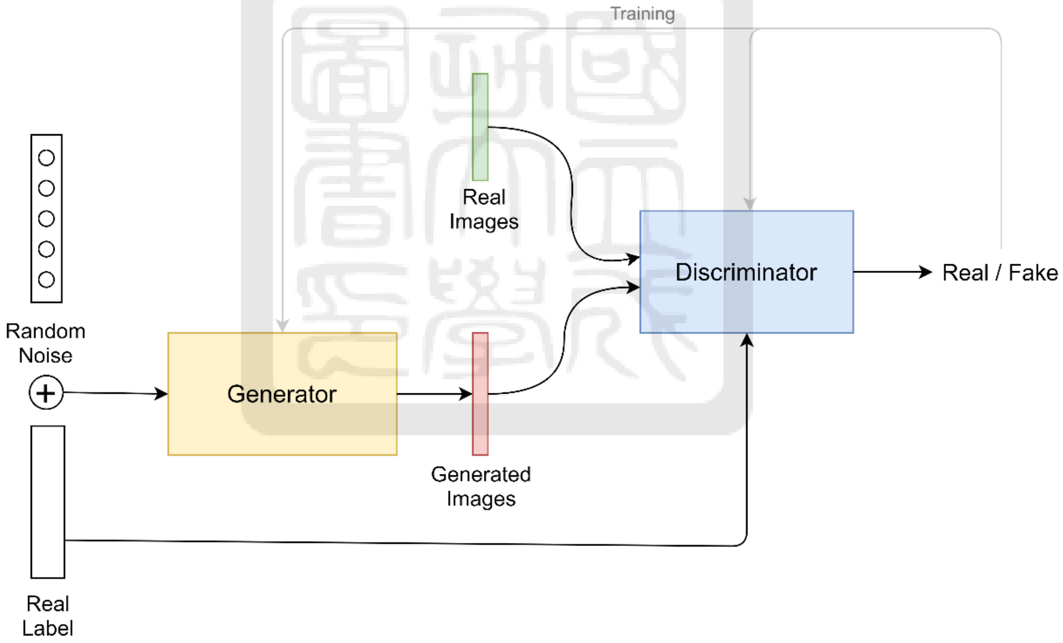


Figure 2 Conditional Generative Adversarial Network

2.3 Font Generation

Font generation contains many different tasks. MC-GAN [12] and AGIS-Net [13] concentrate on the artistic style of glyph images. The former is an end-to-end stacked conditional GAN model, and the latter transfers shape and texture styles in a one-stage model. Attr2Font [14] proposed a model which can create new fonts by synthesizing different glyph images according to user-defined attributes and their corresponding values. FTransGAN[15] is a network that can learn style from different language glyph images, which means its generator has a good ability to learn style. It uses two discriminators to learn content and style, respectively.

Our model is different from the above models, and it is especially for Chinese hand-written characters. Rewrite [16] is a style transfer neural network for Chinese font. It uses CNN as a feature extractor. At that time, result images of Rewrite model are sometimes broken and shattered, leaving it a problem waiting to be solved. Zi2zi [1] is the first generative adversarial network model specialized for Chinese character style transfer. It is an extension of pix2pix. Thanks to category embeddings and category loss, it can handle one-to-many style transfer tasks. FontGAN [4] stylizes and de-stylizes Chinese characters as content representation and style representation. It provides a model network that can both stylized characters and destylized characters at the same time. Another method proposed a collaborative stroke refinement [17] module to solve the thin strokes in hand-written characters. LF-Font [18] combines strong representation and a compact factorization strategy to generate new-style characters. CalliGAN [5] is the first research dealing style at a fine-grained level. It used component code as the input of the recurrent neural network encoder. ChiroGAN[19] is a three-stage network for one-to-many Chinese character translation, including an Enet to extract the skeleton, TNet to do the skeleton transformation, and RNet to transfer the skeleton structure and stroke style for the input image. RD-GAN [20] uses a multi-level discriminator, and its generator renders Chinese characters into radicals. StrokeGAN [6] uses strokes to improve the quality of generated images. Different from our methods using a serial of storks directly to a generator, it designs a stroke encoding method that provides stroke embeddings as input to a discriminator to avoid mode collapse problems.

Chapter 3. Proposed Methodology

3.1 Overview

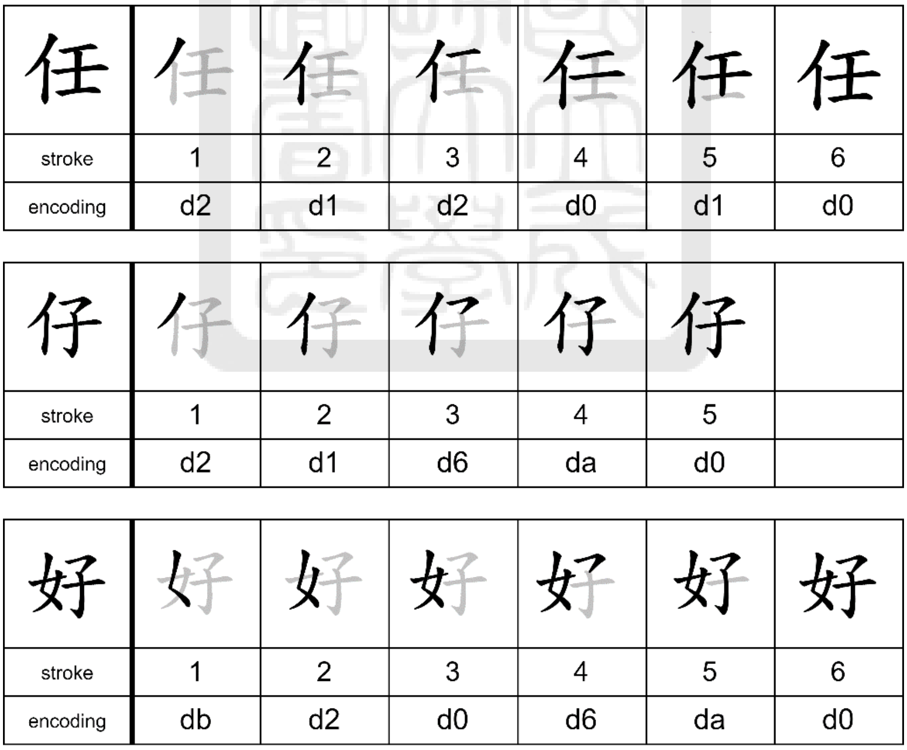
In this section, we describe our proposed model used for Chinese font transfer. Since Chinese characters are highly structured, and they can be decomposed as components or strokes. We take of these features and try to improve our model. The main idea of our model is to add order strokes in the generator to improve the model performance of Chinese calligraphy transfer.

Table 1 basic strokes of Chinese characters and their encodings

1	點	㇔	d4	主	16	豎撇	㇓	d3	風
2	長頓點	㇔	d4	小	17	豎折	㇗	d7	匠
3	橫	㇐	d0	木	18	豎橫折	㇞	de	吳
4	橫鉤	㇖	d6	家	19	豎曲	㇄	d3	匹
5	橫折	㇕	d5	已	20	豎橫折鉤	㇉	c9	弓
6	橫折橫	㇍	c5	凹	21	豎曲鉤	㇟	df	乳
7	橫折鉤	㇆	c6	同	22	豎鉤	㇚	da	小
8	橫折橫折	㇅	ce	凸	23	斜鉤	㇂	c2	武
9	橫撇	㇇	c7	發	24	彎鉤	㇁	c1	冡
10	橫曲鉤	㇠	e0	乙	25	臥鉤	㇃	c3	心
11	橫撇橫折鉤	㇌	e1	乃	26	撇	㇒	d2	戈
12	橫斜鉤	㇈	c8	風	27	撇挑	㇜	dc	私
13	挑	㇀	c0	冰	28	撇頓點	㇛	db	女
14	豎	㇑	d1	十	29	撇橫	㇜	d7	母
15	豎挑	㇙	d9	民	30	捺	㇏	cf	木

During generating process, it may be useful if we get the low-level structure of Chinese characters. Therefore, we decompose Chinese characters as strokes which are the smallest components in Chinese characters. Since strokes provide the smallest basic structural information but not regional information, we think that learning regional information will be helpful. Thus, we use a local discriminator to learn regional information and improve overall model performance.

By using cjklib package, we can easily get stroke orders. Cjklib is a database providing language information related to Han characters. It almost contains all of the characters in Chinese. Table 1 shows the basic strokes of Chinese characters, and we transfer the encoding of cjklib to the 32×32 vector which can then concatenate to the middle vector from the generator. There are some characters of stroke decomposition in Figure 3.



任						
stroke	1	2	3	4	5	6
encoding	d2	d1	d2	d0	d1	d0

仔						
stroke	1	2	3	4	5	
encoding	d2	d1	d6	da	d0	

好						
stroke	1	2	3	4	5	6
encoding	db	d2	d0	d6	da	d0

Figure 3 Stroke Decomposition of Chinese Characters

In general, there is one discriminator in a generative adversarial network. This discriminator watches the whole image and teaches the generator to generate high-quality images. However, sometimes generators will miss details of images. We propose a local discriminator in our work. By contour-based segmentation [24], we decompose characters into components. The concept of character components can be seen in figure 5. Then, we choose the biggest component as characters' concerning regions. Followed the whole images, the concerning region is sent into the generator and the dual discriminator will look after them to help the generator focus on details of the images.

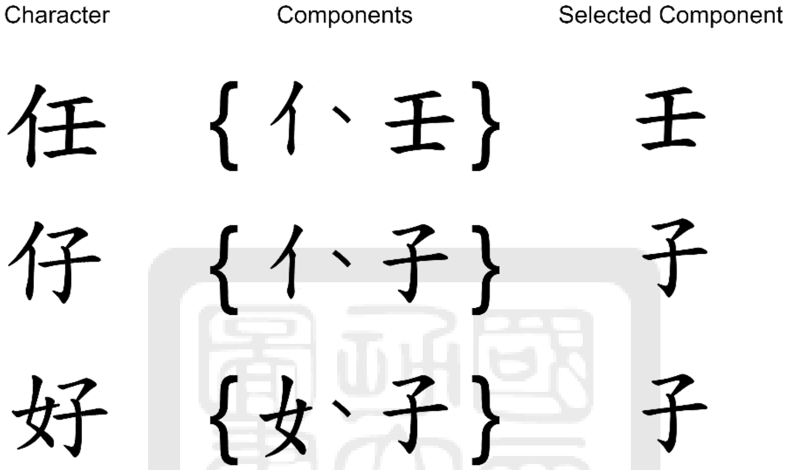


Figure 4 Component Decomposition of Chinese Characters

After we get information about stroke order and concerning region, we can introduce our model.

3.2 Network Architecture

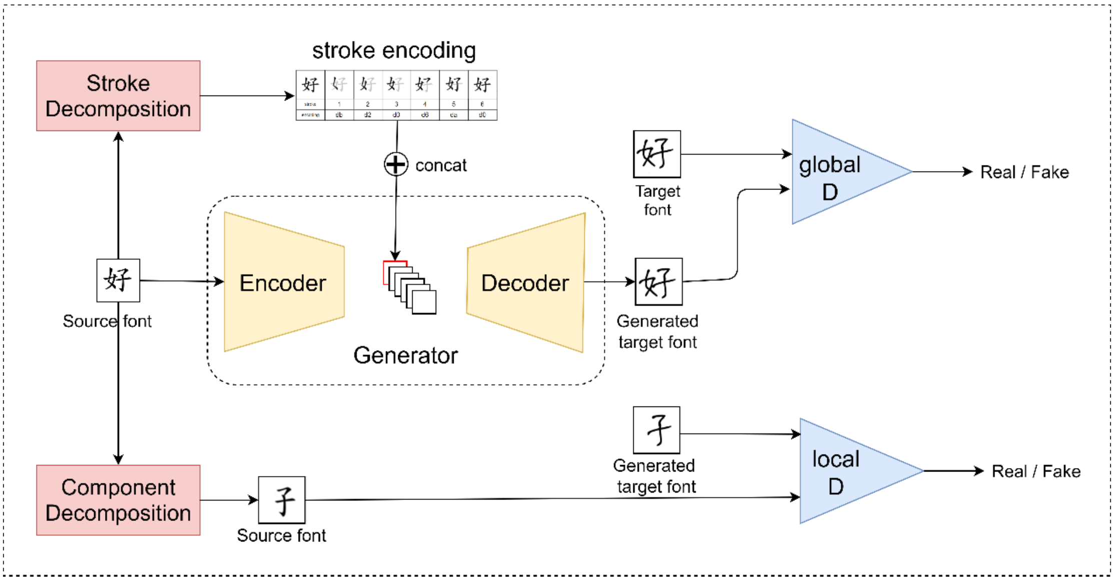


Figure 5 The architecture of our proposed model. Our proposed model contains one generator and two discriminators. In the training process, we can divide it into four steps: (a) We fix the global discriminator and local discriminator. The source font images go through the encoder. After convolution, the vector concatenates with the stroke encodings. This vector then decodes back to the styled hand-written characters. (b) We fix the generator and the local discriminator, and the global discriminator judges if the generated images look similar to the target font images or not. (c) We fix the global discriminator and local discriminator again. We send the image of the Chinese characters’ components to the generator. The generator attempts to produce the component images of the target font, and update its parameters again. (d) At this time, we fix the generator and global discriminator. The local discriminator makes a similarity judgment of the generated component images and target component images. Above are the overall training process of this model.

Our proposed model aims to transfer source font Chinese character images into hand-written glyphs. Let x , x_{local}, y' denote a source font character image, source font concerning region image and generated target font image, while y, y_{local} represent target font character image and target font concerning region image. We use pair (x, y) and pair (x_{local}, y_{local}) as training materials. Furthermore, stroke encoding is written as notation s.

As shown in Figure 5, our proposed model is based on an encoder-decoder generator G and two discriminators, global discriminator D_{global} and local discriminator D_{local}. After decomposing process, we mentioned in previous sections, we got x, x_{local} and s as our training materials.

We encode x through encoder En, and x become an image feature representation v_i of the Chinese character image. Simultaneously, the strokes encoding s are obtained from Cjklib package. We concatenate v_i and s as vector input for Decoder De. Then, we can get the generated target font image x' after decoding.

After Generation, the discriminator tries to teach the generator to generate high-quality images. By sending real image y and generated imagey' images as inputs, the global discriminator can learn how to classify these pairs of images and give feedback to the generator. Although global discriminator is useful for the overall images, Chinese characters are complicated and hard to recognize. Local discriminator reads pair images (y'_{local}, y_{loacl}) for minutiae of the Chinese characters.

3.2.1 Generator and Discriminator

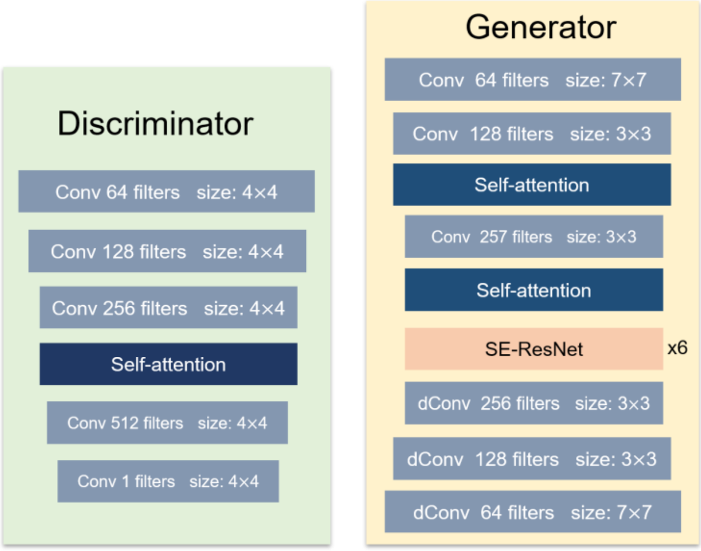

Discriminator
Conv 64 filters size: 4×4
Conv 128 filters size: 4×4
Conv 256 filters size: 4×4
Self-attention
Conv 512 filters size: 4×4
Conv 1 filters size: 4×4
Generator
Conv 64 filters size: 7×7
Conv 128 filters size: 3×3
Self-attention
Conv 257 filters size: 3×3
Self-attention
SE-ResNet
x6
dConv 256 filters size: 3×3
dConv 128 filters size: 3×3
dConv 64 filters size: 7×7

Figure 6 Network structure of our proposed model.

The network structure is shown in Figure 6. Our Generator is an encoder-decoder network that is modified by CycleGAN. The encoder contains three down-sampling modules. Among them are two self-attention [22] layers. Attention layers are widely used in a recent study, and their architecture is in Figure 8. It draws dependencies without regard to the distance, so it can pay attention to whole pictures and get global features. After convolution layers and self-attention layers, stroke encoding s is concatenated to v_i. Process of concatenating strokes is shown in Figure 7.

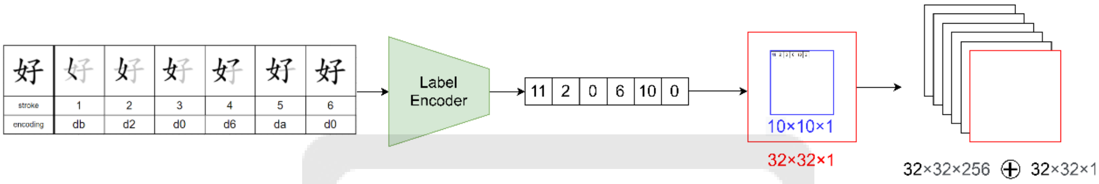


Figure 7 Process of concatenating strokes. We got basic encodings from cjklib packages. Because the encodings are mixed with characters, we use a label encoder to encode stroke encodings with a value between 1 and n classes. Furthermore, v_i has $[32 \times 32 \times 256]$ shape, so we transfer stroke encoding to $[32 \times 32 \times 1]$ the shape which can easily concatenate with v_i.

The concatenating result of stroke encoding s and v_i is an input of the following six SE-ResNet[23]. SE-ResNet combines residual network and squeeze-and-excitation network. In general, while the network is deeper, the network might lose some important features. A residual network can skip the training of few layers, allowing the network to learn simple and complex information simultaneously. SE-net is developed for a weighted representation. When the feature is important, the SE-net will make the feature weight heavier. In the contrast, if the feature is trivial, its weight will be lighter. By using these block modules, our proposed network can get better performance.

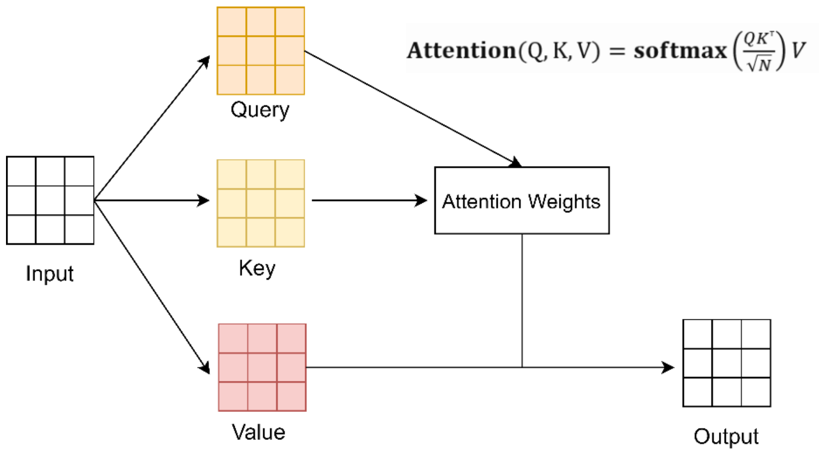

Attention(Q, K, V) = softmax$\left(\frac{QK^T}{\sqrt{N}}\right)V$
Query
Input
Key
Attention Weights
Value
Output

Figure 8 Structure of self-attention

Then, there are three up-sampling modules in Decoder De. It decodes the map vector to the final image. The final generated image y' and target image y will be the input for the global discriminator. Besides, the locally generated image y'_{local} also experiences the same process. The locally generated image y'_{local} and local target image y_{local} will feed the local discriminator.

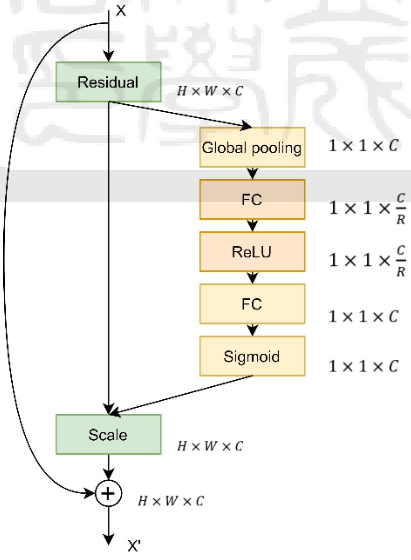


Figure 9 SE-Resnet network

Both D_{local} and D_{global} have the same network structure. It contains five layers of convolutional layers and one self-attention layer. As just mentioned, a self-attention layer can track features from all locations, which helps the discriminator classify generated images well.

3.2.2 Loss Function

To generate realistic images, we must have some loss function to train our model. We define four losses in our model: L_{adv}loss, L_{pixel} loss, L_{const} loss and L_{cx} loss.

The first loss is an adversarial loss of GAN. This formula is a cross-entropy between real and generated distributions. As the generator tries hard to generate realistic images, the discriminator is eager to distinguish the real images and fake images. Thus, while the generator tries to minimize this loss function, the discriminators try to maximize it.

$$L_{adv} = Ex[\log D(x, y)] + Ex[\log (1 - D(x, y'))] \quad (1)$$

Since the generator must minimize the difference between real images and generated images, we define the second loss: a pix-wise l1 loss. Since component-level images are not as similar as global images, we do not use these loss functions in component-level images.

$$L_{pixel} = \|y - y'\| \quad (2)$$

The generated image y' and input image y might look alike, so they represent similar representations. Derived from this concept, we have constancy loss [1] [5] as below.

$$L_{const} = \|Ei(y) - Ei(y')\|^2 \quad (3)$$

Contextual loss [21] is first used for non-align data because it can focus on high-level features instead of pixelwise features. Now it is widely used on many kinds of problems in image processing. Since it uses a pre-trained model VGG19, it greatly improves the speed of convergence. It can be written as a function below. Equation 4 is the loss function, and the equation 5 to 8 are the similarity of cosine to calculate this loss.

$$L_{cx} = CX(Y, Y') = \frac{1}{N} \sum_j \max_i CX_{ij} \quad (4)$$

$$d_{ij} = \left(1 - \frac{(x_i - u_y)(y_j - u_y)}{\|x_i - u_y\|_2 \|y_j - u_y\|_2}\right) \ (5)$$

$$\widetilde{d_{ij}} = \frac{d_{jj}}{min_k \, d_{ik} + \epsilon} \ (6)$$

$$w_{ij} = exp\left(\frac{1 - \widetilde{d_{ij}}}{h}\right) \ (7)$$

$$CX_{ij} = \frac{w_{ij}}{\sum_k w_{il}} \quad (8)$$

Therefore, the total objective function is like following one, where λ1, λ2, λ3, and λ4 are control parameters of the objective function.

$$L = \lambda_1 \times L_{adv} + \lambda_2 \times L_{pixel} + \lambda_3 \times L_{const} + \lambda_4 \times L_{cx} \quad (9)$$

Chapter 4. Implementation and Experiments

4.1 Experiment Setup

The software and hardware specification of this research is shown in detail in Table 2.

Table 2 Experiment environment.

	Setting
OS	Ubuntu 16.04
Platform	Pytorch 1.6.0
Program language	Python 3.6
CPU	Intel(R) Core(TM) i7-6700 CPU@ 3.40GHz
GPU	Nvidia GTX1060 6G
memory	16G

4.1.1 Dataset

Datasets used in this paper divide into three categories: images rendering from TrueType font files, pseudo handwritten images, and handwritten images. For style transfer, we have to decide on a standard font, which is SimSun font. Except for StrokeGAN, zi2zi and CycleGAN have the same datasets. The first type of dataset contains jf-openhuninn font with 100 paired images. The second type includes Jason-Handwriting font and SentyPea font, and both are written by designers and render by .ttf file. Jason-Handwriting font contains 300 paired images, while SentyPea font has 400 paired images. The last part of the datasets is collected by us. We find some volunteers to write these Chinese characters for us, and they all contain 450 paired images. Since hand-written glyphs are hard for us to evaluate, we will show the image result in the last part of this paper. Moreover, StrokeGAN needs some preprocessing process to decompose characters, we compare it by the datasets provided on their github page. Their training datasets are adjusted by us to make them easier to compare. Jf-openhuninn font contains 339 paired characters; Jason-Handwriting font includes 486 paired images, while SentyPea font has 484 paired images. For testing sets, we use 400 unseen characters of the sample, respectively. The details of our datasets are summarized as Table 3.

Table 3 Datasets of our experiments. The first two rows of training and testing number of images of comparison experiment between zi2zi, CycleGAN, and our proposed model. The last is the number of images for the experiment comparison between StrokeGAN and our proposed model.

Datasets	**Simsun to jf-openhunin (1)**	**Simsun to Jason-Handwriting (2)**	**Simsun to Sentypea (3)**	**Simsun to Sentytea (4)**
Training	100	300	400	400
Testing	400	400	400	400
Training	340	486	483	482
Testing	100	100	100	100
Style	中易宋體轉粉圓體	中易宋體轉清松手寫體	中易宋體轉新蒂綠豆體	中易宋體轉新蒂下午茶體

4.1.2 Baseline Model

There are three baseline models, including zi2zi, CycleGAN, and StrokeGAN. All of them can be seen as a state-of-art method in Chinese calligraphy transfer. When we do the comparison, we use the source code they provided on the internet. We do not change the parameters for training except we notice the training epochs are not enough to converge. Zi2zi is the first GAN model specifically for Chinese font transfer, and it is based on pix2pix. CycleGAN is commonly used in domain transfer. It also works well in Chinese-style transfer. StrokeGAN is proposed in 2021, which contains stroke encodings. Although StrokeGAN and our proposed method both include stroke encodings, we used different methods. Since strokes correlate with the order, it might be improper to get rid of this feature. StrokeGAN uses a one-hot vector while our method contains serial information of this feature. In addition, our work also contains some special layers which improve the results.

4.2 Experiment Results

Next, we demonstrate the ability of our proposed model. First, we compare our proposed model with other state-of-art models. Then, we analyze the effects of some components in our proposed model.

4.2.1 Comparison with the State of the Art

Font glyphs are hard to evaluate because judging the arts is not the traditional task for computers. It is hard to give a standard to determine which model outperforms the other models. Nevertheless, we still evaluate our generated images and target images with some image quality assessment, including root-mean-square error (RMSE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM). They are the classical way to decide the quality of images, and they can be good criteria. The definitions of these image quality assessments can be seen as the following functions.

$$RMS = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - y_i)^2} \quad (10)$$

$$PSNR = 10\log_{10}\left(\frac{MAX_i^2}{MSE}\right) \quad (11)$$

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x{}^2 + \mu_y{}^2 + c_1)(\sigma_x{}^2 + \sigma_y{}^2 + c_2)} \quad (12)$$

Table 4 shows the qualitative comparison results of zi2zi2, CycleGAN, and our proposed method. Figure 10 shows the quantitative comparisons between our proposed model and other models. The first form in Figure 10 is the first type dataset rendering from jf-openhuninn font. This font is highly structured, and the generated images from three models are like target font. Although SSIM of our proposed model is a little lower than zi2zi, we can see results on the generated images. Our model performs as well as zi2zi, and even better than zi2zi. The second form in Figure 10 and The last form in

Figure 10 are the second type dataset, but we can see that Jason-handwriting is also a clear font, so its result resembles jf-openhuninn font. On the contrary, Sentypea font is a complicated and stylized hand-written font. Our proposed model outperforms other previous methods qualitatively and quantitatively.

Table 4 Comparison among previous researches and our proposed method.

Dataset	**Model**	**RMSE(↓)**	**SSIM(↑)**	**PSNR(↑)**	**epoch(↓)**
(1)	zi2zi	5.1563	0.58599	33.9211	99
	Cycle-gan	5.1304	0.49010	33.9588	399
	Proposed	4.7771	0.55844	34.5722	59
(2)	zi2zi	5.1652	0.56413	33.9146	149
	CycleGAN	5.1087	0.48344	33.9987	429
	Proposed	4.6572	0.52225	34.8121	19
(3)	zi2zi	6.1199	0.49716	37.6344	499
	CycleGAN	6.0765	0.46553	37.6960	349
	Proposed	5.8876	0.52576	38.0267	199
(4)	zi2zi	5.0822	0.62107	39.4583	1200
	CycleGAN	5.0370	0.54424	39.5228	199
	Proposed	4.6796	0.62361	40.2790	59

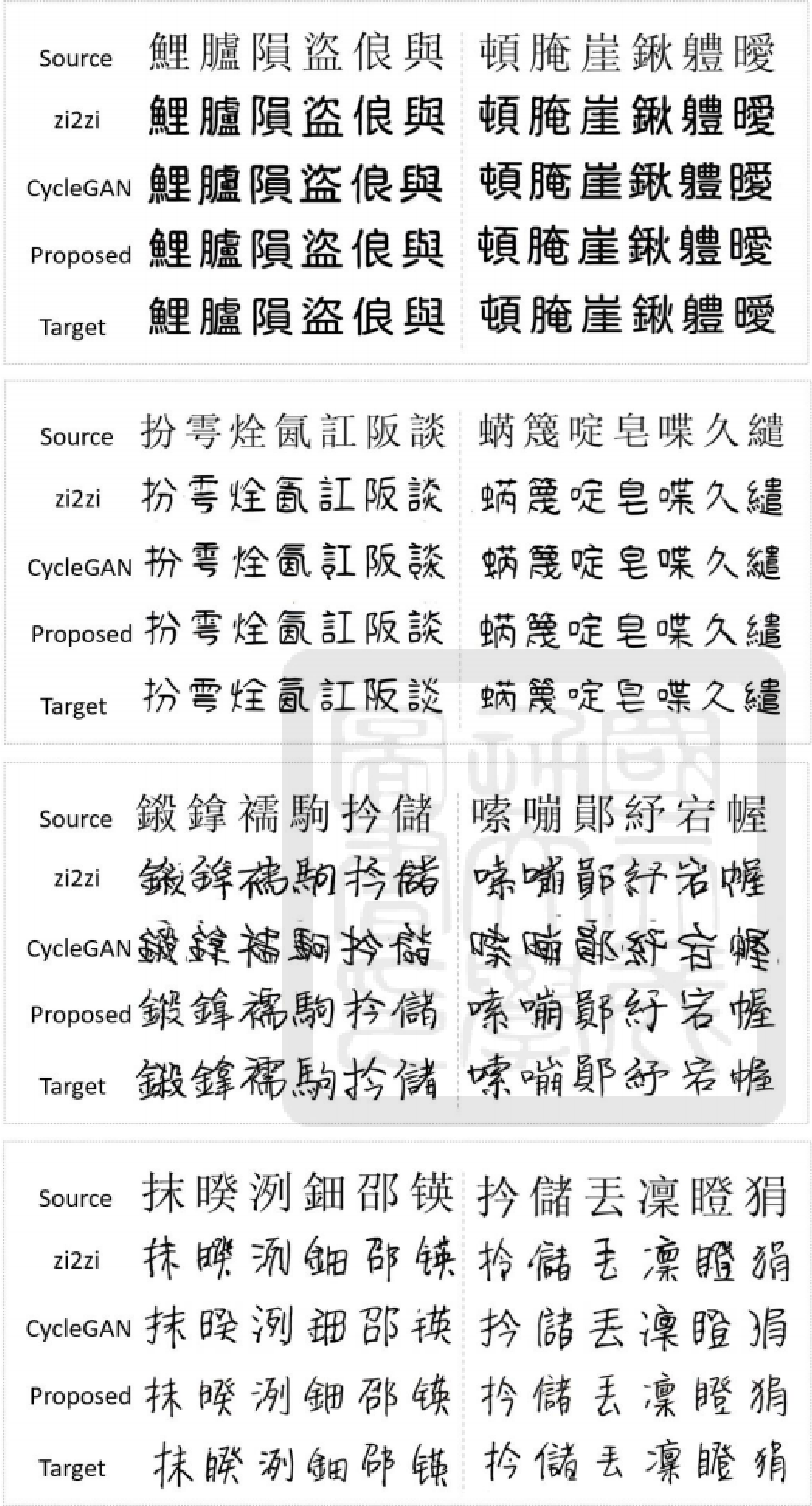

Source
zi2zi
CycleGAN
Proposed
Target
鯉臚隕盜俍與 頓腌崖鍬體曖
扮雩烇氤訌阪談 蛃篋啶皂喋久繾
鑀鍏襦駒拎儲 嗦�鄖紓宕幄
抹暌洌鈿邵铗 拎儲丟凜瞪狷

Figure 10 Comparison of our proposed model and other state-of-art models. The first one is training with 100 images from Simsun font to jf-openhuninn font. The second one is the result of training with 300 images from Simsun font to Jason-Handwriting font. The last one is the outcome of training with 400 images from Simsun font to Sentypea font.

The above models use the same datasets to train. We can observe that our methods perform well among them. Then, we compare our model with StrokeGAN. Since StrokeGAN is like our model which needs stroke decomposition, and the author did not provide traditional Chinese characters’ decomposition datasets, we use simplified Chinese datasets on their github page to train our model and do the comparison. We reduce the number of characters and change their original font to the previously mentioned fonts. The results are shown in Figure 11 and Table 5. We use 340, 486, and 483 characters as character images in Simsun to jf-openhuninn, Simsun to Jason-Handwriting, and Simsun to Sentypea respectively. Since we use fewer images for training compared to the original StrokeGAN paper, the outcome of the experiment might not be as good as they show on their paper.

Table 5 Comparison results of our proposed model and StrokeGAN.

Dataset	**Model**	**RMSE(↓)**	**SSIM(↑)**	**PSNR(↑)**	**epoch(↓)**
(1)	StrokeGAN	3.4223	0.52057	37.4762	550
	Proposed	4.1747	0.58703	35.7312	59
(2)	StrokeGAN	3.5484	0.45465	37.1443	600
	Proposed	3.6703	0.46127	36.8717	19
(3)	StrokeGAN	3.5415	0.52689	37.2353	600
	Proposed	4.0585	0.56362	36.1194	199
(4)	StrokeGAN	3.5703	0.53142	37.1162	600
	Proposed	3.8305	0.58414	36.5190	59

Although StrokeGAN does not outperform our proposed model in Figure 11, its qualitative analysis is not worse than ours in Table 5. It explains that the current image standard for Chinese character style transfer is not suitable. The idea of similarity is different between machines and humans, so it may be a future work for researchers to find a proper image standard for character style transfer.

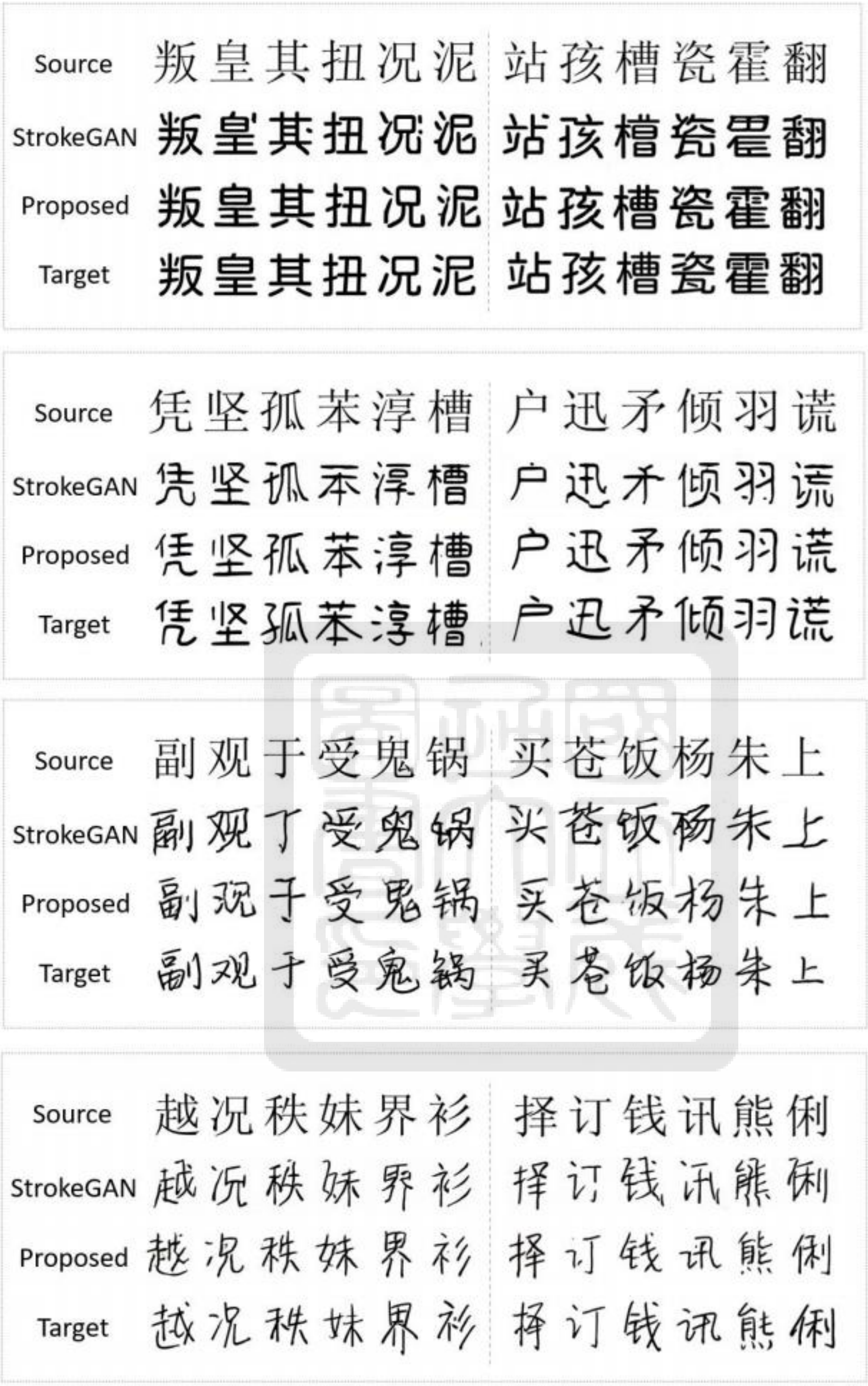

Source
StrokeGAN
Proposed
Target
叛皇其扭况泥 站孩槽瓷霍翻
凭坚孤苯淳槽 户迅矛倾羽谎
副观于受鬼锅 买苍饭杨朱上
越况秩妹界衫 择订钱讯熊俐

Figure 11 Results of comparison between our proposed model and StrokeGAN.

4.2.2 Questionnaire Survey

Since different fonts are meaningful for humans instead of for machines. We think the comparison results of image quality assessment are good references, but human feedback is more important for our research. Therefore, we design a questionnaire survey to get humans' opinions.

In our questionnaire survey, 26 subjects are all native speakers of Mandarin, and they can correctly read and write traditional Chinese. We design 8 questions in our questionnaire survey, and each question is for one font. For instance, we choose 12 character images generated by zi2zi, CycleGAN, and our proposed model for the former four questions. Compared with target font images, the participants choose a more similar one to the target font images. The participants do not know which model generates the images. The preferred image rate is showed in Table 6.

Table 6 Ratio of preferred images between target font characters and generated characters in our questionnaire survey. People think our proposed model is the most similar one to the target font.

	(1)	**(2)**	**(3)**	**(4)**
zi2zi	0.1923	0.3846	0	0.1153
CycleGAN	0.0769	0.1923	0.0384	0.0384
Proposed	0.7307	0.4615	0.9615	0.8461

The later four questions are comparisons of StrokeGAN and our proposed model for each font. The preferred image rate is as below. We can notice that the images generated by our proposed model are the most similar images to target font images seen by humans.

Table 7 Ratio of preferred images between target font characters and generated characters in our questionnaire survey. People think our proposed model is more similar to the target font compare to StrokeGAN.

	(1)	(2)	(3)	(4)
StrokeGAN	0.0384	0.0384	0.1153	0.1153
Proposed	0.9615	0.9615	0.8846	0.8846

4.2.3 Ablation Studies

Effect of Self-attention. Self-attention is a popular network since it can watch the whole picture while a convolutional network focuses on regional features. However, if we add more self-attention layers, the performance will not improve linearly. We experiment to check the proper amount of self-attention layers. Figure 12 shows three self-attention-related networks we have tried, and Figure 13 shows the results of these models. By comparing the performance of different layers of the self-attention model, we choose the model we mentioned in chapter 3 before.

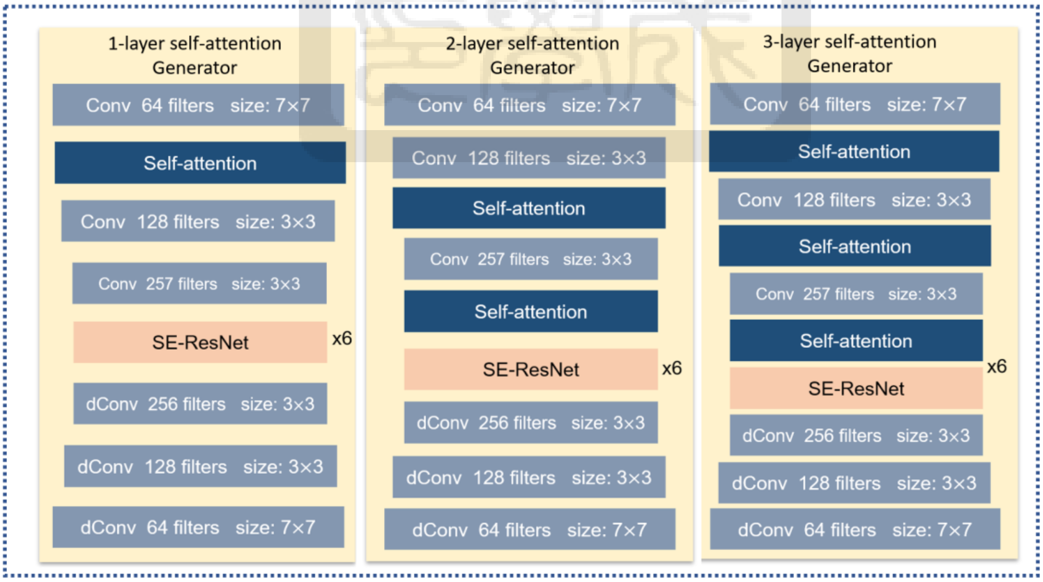

1-layer self-attention Generator
Conv 64 filters size: 7×7
Self-attention
Conv 128 filters size: 3×3
Conv 257 filters size: 3×3
SE-ResNet
x6
dConv 256 filters size: 3×3
dConv 128 filters size: 3×3
dConv 64 filters size: 7×7
2-layer self-attention Generator
Conv 64 filters size: 7×7
Conv 128 filters size: 3×3
Self-attention
Conv 257 filters size: 3×3
Self-attention
SE-ResNet
x6
dConv 256 filters size: 3×3
dConv 128 filters size: 3×3
dConv 64 filters size: 7×7
3-layer self-attention Generator
Conv 64 filters size: 7×7
Self-attention
Conv 128 filters size: 3×3
Self-attention
Conv 257 filters size: 3×3
Self-attention
SE-ResNet
x6
dConv 256 filters size: 3×3
dConv 128 filters size: 3×3
dConv 64 filters size: 7×7

Figure 12 Architecture of three type of self-attention related network.

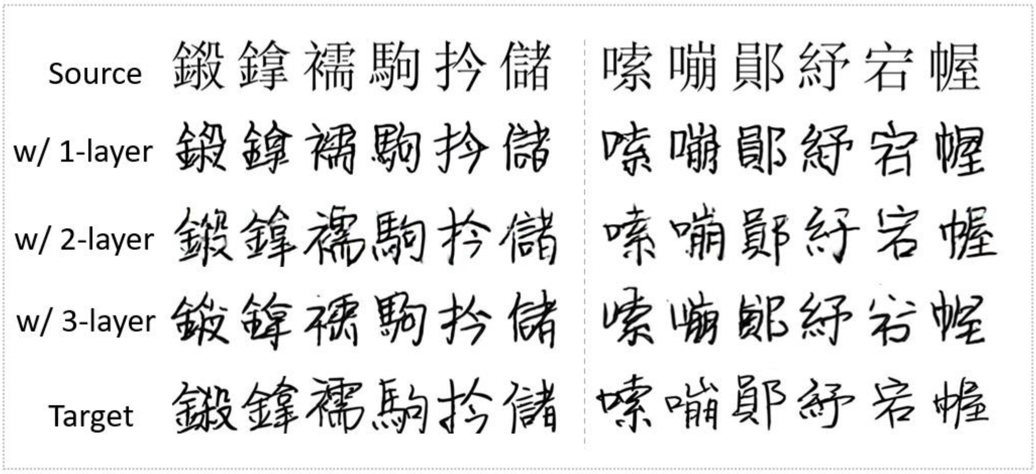


Figure 13 Comparison of three types of self-attention-related network. As picture showed above, we can notice that if the self-attention layer is used in low-dimension, the strokes of characters will be clear. Yet, it might lose some content information. We choose 2-layer self-attention generator as our proposed model network.

Effect of contextual loss. Contextual loss is powerful for image style transfer. It not only considers context but also semantics. In a small region, it compares similar semantics but considering the context for the entire image. Since it denotes the layer of the pre-trained VGG19 model, it highly saves our time for training. Table 8 shows the result of three datasets whether they train with contextual loss or not. We can see that it saves more than half the time.

Table 8 Comparison of the proposed model with contextual loss and without contextual loss

	(1)	**(2)**	**(3)**
w/ L_{cx}	59	19	199
w/o L_{cx}	399	499	369

4.3 Other Applications

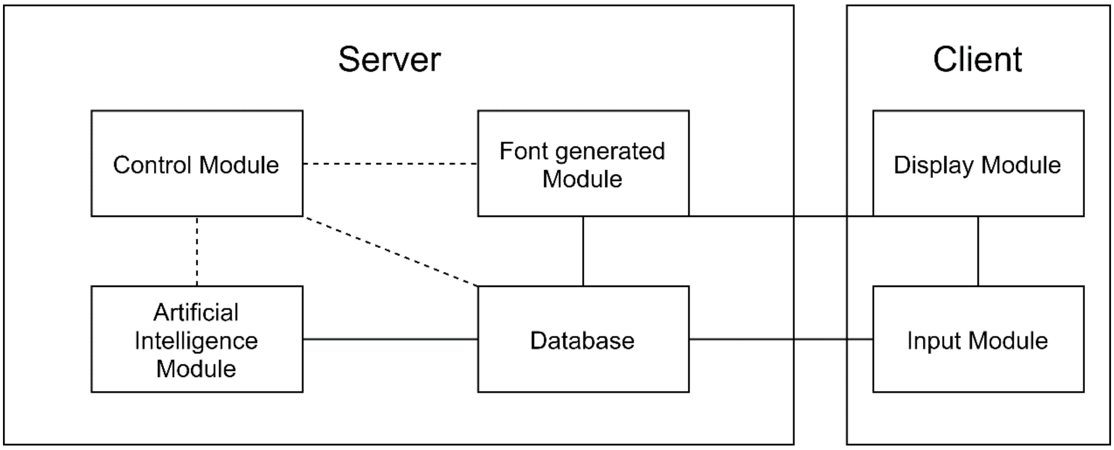


Figure 14 Architecture of our system using to generate hand-written Chinese font.

We design a system to test our model and generate personal hand-written Chinese fonts. The architecture of this system is like Figure 14 above. It belongs to a client/server architecture. While the server contains a control module, database, artificial intelligence module, and font generated module, the client includes an input module and display module.

The users first use their phones to connect to the client website. After registering and logging in, the database module will keep the users' information and return the Chinese characters to the display module. The characters will be shown on the screen, and the users can use their phones to input the hand-written characters. The above steps will repeat until collecting enough Chinese characters. Then, the control module will make the artificial intelligence module start training. After that, the generated module will use the results from the artificial intelligence module to make TrueType font files that are similar to the users' hand-written style. The users can download the files and enjoy the fun typing experience.

The results of some of the subjects are like Figure 15. Besides, we collect four subjects' hand-written glyphs as new datasets, and the whole datasets can be downloaded on github (https://github.com/ishtien/HandwrittenChineseDatasets). They are the third type of dataset we mentioned before.

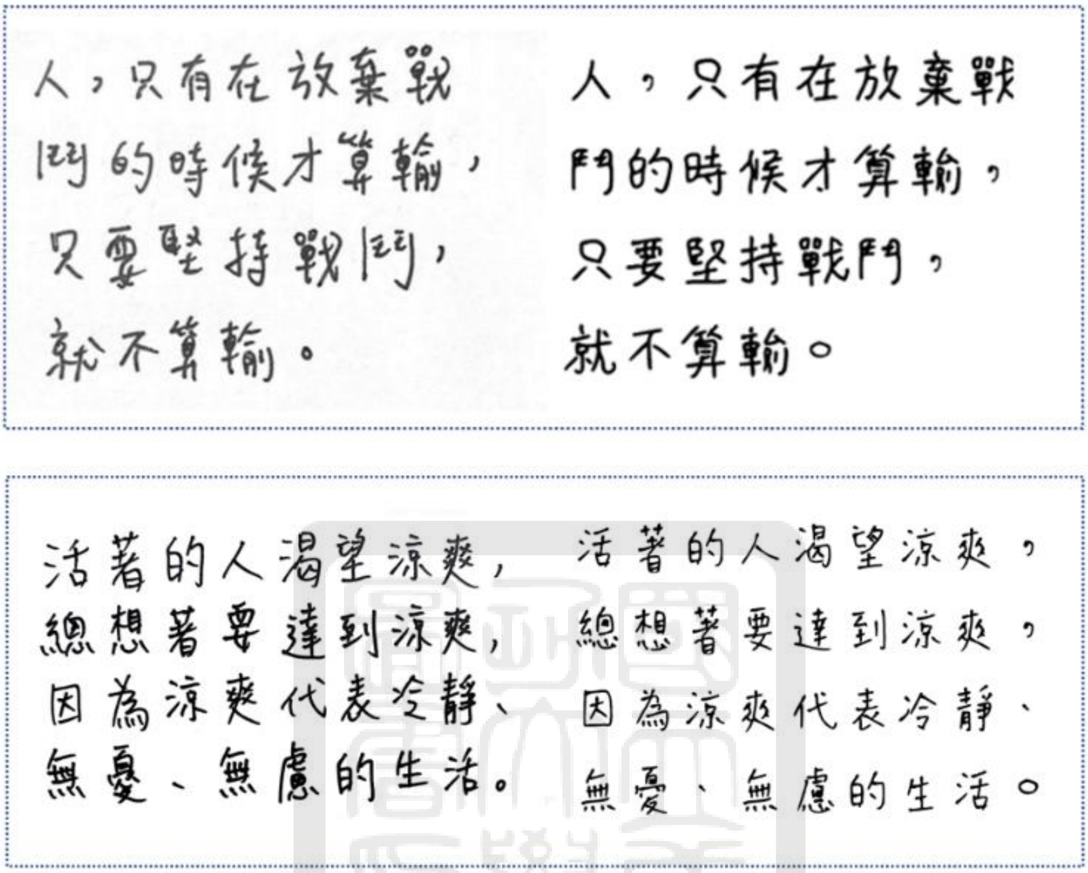


Figure 15 The sentence on the left image is written by the subject, and the right one is a typing result of a TrueType font file on the computer.

Chapter 5. Conclusion and Future Work

5.1 Conclusion

In this paper, we propose a novel model for hand-written Chinese characters transfer. We learn Chinese characters glyph with three levels. First, like many other generative adversarial networks, we learn the integrated images in the training process. Second, we use a local discriminator to concentrate on complicated regions, and it can help the generator to generate high-quality images. Last, we add stroke encodings to our generator. Because strokes are the smallest component in Chinese characters, it greatly improves our proposed model. Because of this three-level learning method, our model can generate high-quality Chinese characters images. We also add some state-of-art architecture and loss functions in our proposed model. Moreover, we compare our model with the existing models nowadays and get good performance. Not only do we provide new Chinese hand-written datasets for researchers, but we offer a system to generate personal font files. Additionally, it can be used as a convenient collector of font characters.

5.2 Future Work

Although our model can transfer hand-written Chinese characters well, there are still many tasks to do. It may take many resources to train a model, so reducing training resources is an important issue. Either using another pre-trained model or design a more efficient model will be a challenging and interesting task. Besides, the quantity of characters in datasets is an important factor to generate high-quality images. Users must write lots of characters to generate new fonts, which decreases their desire to make their fonts. Some strong methods of data augmentation may solve this problem. In addition, by training different hand-written Chinese characters from many people, it might be possible for a machine to learn some features and then create its glyphs. Chinese characters are like some of the totems and art design, so applying our proposed model to other scenarios is also a good direction for future research. Moreover, our system to generate personal font is very convenient, but we can make it more user-friendly. Now, we use frames to remind users to write specific sizes of characters. Maybe automatically detect the characters can solve the problem that we can only use the same size of characters for training, and we can make the system client mode become a notebook app or website that can simultaneously adjust the generated fonts and write notes, which would make the application more practical. There are so many interesting and challenging tasks to do, and we hope we can accomplish these tasks as soon as possible.

Reference

[1] Yuchen Tian. zi2zi: Master Chinese Calligraphy with Conditional Adversarial Networks. https://github.com/kaonashi-tyc/zi2zi, 2017.

[2] Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

[3] Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2223-2232).

[4] Liu, X., Meng, G., Xiang, S., & Pan, C. (2019). FontGAN: A Unified Generative Framework for Chinese Character Stylization and De-stylization. arXiv preprint arXiv:1910.12604.

[5] Wu, S. J., Yang, C. Y., & Hsu, J. Y. J. (2020). CalliGAN: style and structure-aware Chinese calligraphy character generator. arXiv preprint arXiv:2005.12500.

[6] Zeng, J., Chen, Q., Liu, Y., Wang, M., & Yao, Y. (2020). StrokeGAN: Reducing Mode Collapse in Chinese Font Generation via Stroke Encoding. arXiv preprint arXiv:2012.08687.

[7] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[8] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … & Bengio, Y. (2014). Generative adversarial networks. arXiv preprint arXiv:1406.2661.

[9] Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

[10] Arjovsky, M., Chintala, S., & Bottou, L. (2017, July). Wasserstein generative adversarial networks. In International conference on machine learning (pp. 214-223). PMLR.

[11] Nguyen, T. D., Le, T., Vu, H., & Phung, D. (2017). Dual discriminator generative adversarial nets. arXiv preprint arXiv:1709.03831.

[12] Azadi, S., Fisher, M., Kim, V. G., Wang, Z., Shechtman, E., & Darrell, T. (2018). Multi-content gan for few-shot font style transfer. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7564-7573).

[13] Gao, Y., Guo, Y., Lian, Z., Tang, Y., & Xiao, J. (2019). Artistic glyph image synthesis via one-stage few-shot learning. ACM Transactions on Graphics (TOG), 38(6), 1-12.

[14] Wang, Y., Gao, Y., & Lian, Z. (2020). Attribute2Font: creating fonts you want from attributes. ACM Transactions on Graphics (TOG), 39(4), 69-1.

[15] Li, C., Taniguchi, Y., Lu, M., & Konomi, S. I. (2021). Few-shot Font Style Transfer between Different Languages. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 433-442).

[16] Yuchen Tian. Rewrite: Neural Style Transfer For Chinese Fonts. https://github.com/kaonashi-tyc/Rewrite. 2017

[17] Wen, C., Pan, Y., Chang, J., Zhang, Y., Chen, S., Wang, Y., … & Tian, Q. (2021). Handwritten Chinese font generation with collaborative stroke refinement. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 3882-3891).

[18] Park, S., Chun, S., Cha, J., Lee, B., & Shim, H. (2020). Few-shot Font Generation with Localized Style Representations and Factorization. arXiv preprint arXiv:2009.11042.

[19] Gao, Y., & Wu, J. (2020, April). GAN-Based Unpaired Chinese Character Image Translation via Skeleton Transformation and Stroke Rendering. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 01, pp. 646-653).

[20] Huang, Y., He, M., Jin, L., & Wang, Y. (2020, August). RD-GAN: Few/Zero-Shot Chinese Character Style Transfer via Radical Decomposition and Rendering. In European Conference on Computer Vision (pp. 156-172). Springer, Cham.

[21] Mechrez, R., Talmi, I., & Zelnik-Manor, L. (2018). The contextual loss for image transformation with non-aligned data. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 768-783).

[22] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008).

[23] Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7132-7141).

[24] Suzuki, S. (1985). Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing, 30(1), 32-46.