CS230

Learning to Generate Handwritten Chinese Characters

Xi Chen
Stanford University
xc58@stanford.edu

Abstract

Generating handwritten Chinese characters is a challenging task due to the complex shapes and structures of Chinese characters, yet it has lots of potential beneficial application areas such as teaching calligraphy. In this project, we used CycleGAN as our baseline model, and explored the effects of different variations on the model performance. We used FID score as well as human inspection to evaluate our models quantitatively and qualitatively.

1 Introduction

In China and other east Asian countries, handwriting is an important skill from an early age. Elementary school students would spend years training and developing elegant handwriting styles by imitating calligraphers. Handwriting is a popular visual art form and calligraphies are often displayed in offices and homes as decoration. In modern communication, handwritten letters are considered to convey more respect and sometimes businesses would handwrite letters to communicate with valuable clients.

However, due to the large number of Chinese characters and the complex shapes and structures of Chinese characters, it is challenging for computers to learn the handwriting styles of Chinese characters. If handwriting Chinese characters can be automated by computer, there will be many potential application areas. For example, customer service agents can type the letter and have the computer transform it into a handwritten letter for efficiency. Teachers can generate any Chinese character in the style of famous Chinese calligraphers and students can learn from it.

This project aims to train a model that can learn the mapping between typed Chinese font and handwritten Chinese characters, so that given typed Chinese characters in a source style, the model can generate handwritten Chinese characters in the target style.

2 Related work

- **Image Style Transfer**

  Neural style transfer (e.g. VGG-19) (1) generates output images by combining the content from one image with the style from another. However, this requires paired images from source domain and target domain, and is not suitable for our task of training on unpaired collections of images.

- **zi2zi**

  zi2zi (2) is a conditional generative adversarial network based on the pix2pix (3) model. It is able to transform characters into multiple fonts with one trained model. However, similar to neural style transfer, this also requires paired training examples.

CS230: Deep Learning, Spring 2022, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

- **CycleGAN**
  Cycle-consistent GAN (CycleGAN) (4) uses unpaired examples to train the mapping between source and target domains. We chose this as our baseline model for the task of generating handwritten characters. Dense-CycleGAN (5) aims to solve a similar task, and uses DenseNet instead of residual blocks in the transform module.

3 Dataset and data preprocessing

The datasets for handwritten chinese characters are downloaded from CASIA Online and Offline Chinese Handwriting Databases (6) (7). We chose an arbitrary style (HW1252) as our target style, and a standard Chinese font SIMHEI as our source style.

To make the dataset compatible with CycleGAN model, we resized the images to 256×256. We also randomly split the dataset into training and testing sets based on an input ratio. The preprocessing implementation is adapted from `https://github.com/changebo/HCCG-CycleGAN`.

4 Methods

We adopt the model architecture from the original CycleGAN (4), as illustrated in figure 1. The input image from source domain A is fed into Generator A2B, which aims to transform it into target domain B; the generated image is then fed into Generator B2A, and the final output should be close to the original input. The original input from source domain and the generated image are fed into the discriminator, which aims to distinguish between real images and generator outputs.

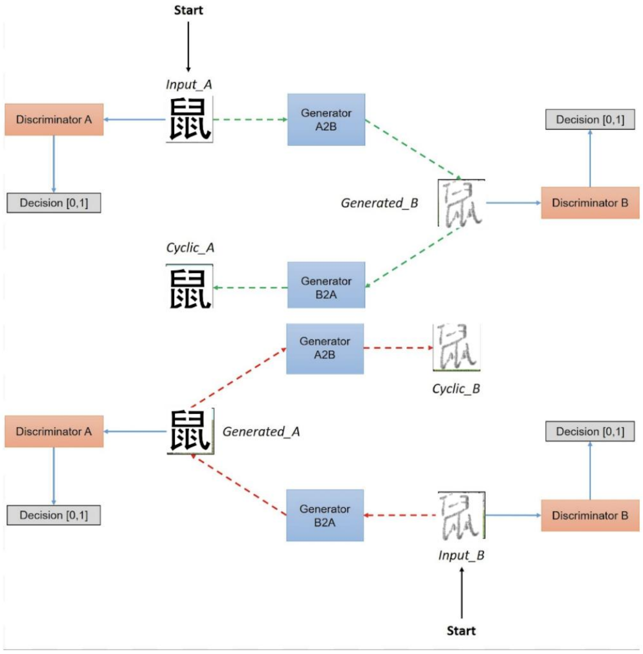


Figure 1: Simplified illustration of CycleGAN architecture (diagram adapted from (8))

As shown in figure 2, the generator consists of 3 convolutional layers (encoder), followed by 9 residual blocks (transformer), and 2 convolutional layers (decoder); the discriminator uses 70×70 PatchGAN, which consists of 5 convolutional layers (4).

The training objective is to minimize the adversarial losses and cycle consistency loss. Adversarial loss incentivizes generator to produce outputs with identical distribution as the target domain, and cycle consistency loss ensures that the image transformation cycle can bring the image back to its original state, i.e. $x \to G(x) \to F(G(x)) \approx x$.

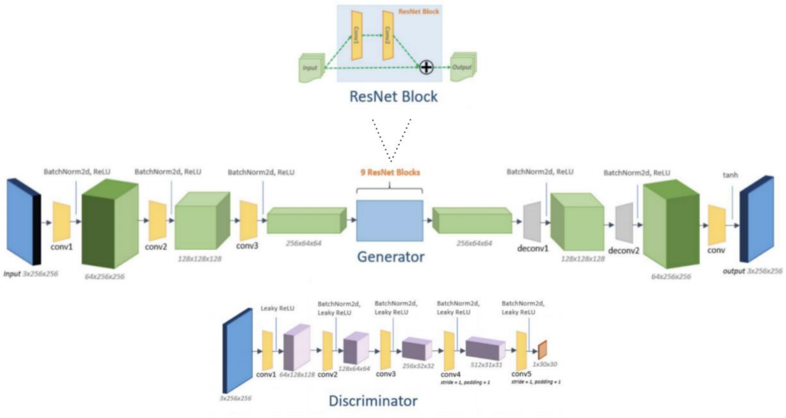


Figure 2: generator and discriminator model architecture (diagram adapted from (9))

This can be formularized as:

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X) + \lambda \dot{L}_{cyc}(G, F), \qquad (1)$$

where Adversarial loss for G (and similarly for F) is:

$$L_{GAN}(G, D_Y, X, Y) = E_{y \sim p_{data}(y)}[logD_Y(y)] + E_{x \sim p_{data}(x)}[log(1 - D_Y(G(x))], \qquad (2)$$

cycle consistency loss is:

$$L_{cyc}(G, F) = E_{x \sim p_{data}(x)}[||F(G(x))x||_1] + E_{y \sim p_{data}(y)}[||G(F(y))y||_1], \qquad (3)$$

and λ controls the relative importance of the two objectives. The goal is to solve for

$$G^*, F^* = arg \min_{G,F} \max_{D_x, D_Y} L(G, F, D_X, D_Y) \qquad (4)$$

5 Experiments

5.1 Evaluation metrics

We use FID (Fréchet Inception Distance) score as the quantitative metric, as well as human evaluation to evaluate our models.

FID score measures the similarity between the generated images and the real images, by calculating the Frechet distance between the multivariate Gaussians of the activations from Inception v3 model for the two collections. A lower FID score indicates greater similarity (11).

The formula for FID score is:

$$d^2 = ||\mu_1 \mu_2||^2 + T_r(C_1 + C_2 2\sqrt{C_1 C_2}), \qquad (5)$$

where μ is the feature-wise mean, and C_1, C_2 are covariance matrices.

Although FID score is able to assess the content accuracy, it does not give a good measure of style discrepancy (5). Therefore we also manually inspected the generated images for each variation, and compared them with the real images from the target style.

5.2 Experiment details and results

Using the original CycleGAN as our baseline model, we explored the effects of the following variations on the model performance. For each variation, we saved the model for every 20 training epochs, and we evaluated the results from epoch 80 and 200.

For all variations, we used Adam optimizer with batch size of 1. We trained the models for 200 epochs, using learning rate 0.0002, with linear decay after 100 epochs, and cycle loss weight $\lambda = 10$.

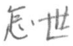
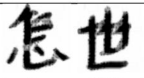
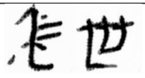
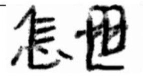
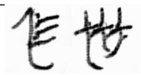
- **Training dataset size**

  Well-performing models usually rely on larger training datasets to learn from. However for our particular task, the training data is very limited since it is difficult to obtain a large number of writing samples. In addition, training on large datasets takes longer and requires more computational resources.

  To study the impact of training dataset size on the model performance, we trained our model on hw1252 dataset, using 0.1 train/test split ratio (i.e. 376 training images) and 0.3 train/test ratio (i.e. 1127 training images), and the result is shown in tables below.

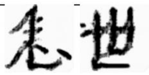
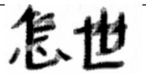
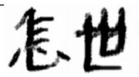
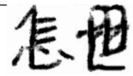
  The model trained with larger dataset has lower FID scores. Although the output resembles the target handwritten style better, there are more noticeable missing (or extra) strokes in the generated images. In addition, we observed that for the larger dataset, the generated characters after epoch 200 appear worse than those from epoch 80. This might be due to over-fitting of the training data.

	0.1 train/test split ratio	0.3 train/test split ratio
FID score after 80 epochs	106.435	54.970
FID score after 200 epochs	86.612	54.503

source font	target style hw1252	training epoch	0.3 train/test split ratio	0.3 train/test split ratio
怎 世		80		
		200		

- **Deeper network (i.e. more residual blocks)**

  The original CycleGAN model uses 6 residual blocks for low resolution (128 × 128) images and 9 residual blocks for high resolution (256 × 256) images (4). We hypothesize that our task of generating Chinese characters may not benefit much from deeper networks, since unlike photos or paintings, higher resolution character images might not indicate more features to encode. We trained our model on hw1252 dataset using 376 training images with both 6 and 9 residual blocks, and the results are shown in table below. We observed that ResNet6 has a slightly better appearance than ResNet9, as indicated by lower FID score and human inspection.

	6 residual blocks (ResNet 6)	9 residual blocks (ResNet 9)
FID score and example outputs after 80 epochs	68.841 	106.435 
FID score and example outputs after 200 epochs	83.855 	86.612 

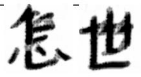
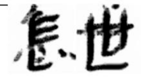
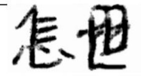
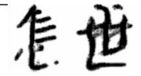
- **Identity learning**

  Original CycleGan implementation applies Identity learning to achieve color preservation in the generated images, i.e. the generator is regularized to a near identity mapping when real images from the target domain are provided as input. The identity loss is formularized as:

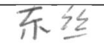
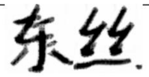
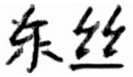
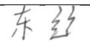
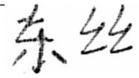
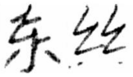
$$L_{identity}(G, F) = E_{y\sim p_{data}(y)}||G(y)y||_1] + E_{x\sim p_{data}(x)}[||F(x)x||_1]. \tag{6}$$

  We hypothesize that given our dataset do not have color information, model performance would not be significantly impacted by identity learning.

  We trained the models with and without identity learning, and the result is shown below. Although the FID score is lower with identity learning, we did not observe a significant difference in generated characters, which is consistent with our hypothesis.

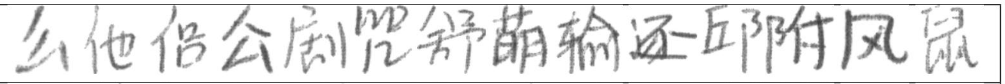
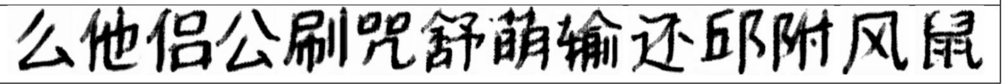
	No identity learning	With identity learning
FID score and example outputs after 80 epochs	106.435	114.471
		
FID score and example outputs after 200 epochs	86.612	60.170
		

- **Different handwritten style datasets**
  We trained our model on a different dataset to learn the impact of different target styles on model performance. We chose hw1169 style, which contains more cursive and 'connected' strokes compared to hw1252. The train/split ratio of 0.1 is used, and the result is shown below.

	FID score and examples after 80 epochs	FID score and examples after 200 epochs
source font	东 丝	
Target style Hw1252		
Generated image hw1252	106.435	86.612
		
Target style Hw1169		
Generated image hw1169	121.931	52.716
		

5.3 Analysis

From our experiments, we observed that more training epochs in general can result in better resemblance of the target style, except for the larger training set, which might be due to over-fitting. The best performing model uses 0.1 train/test ratio, with 6 residual blocks and no identity learning. Below table shows more generated characters using this model.

hw1252	
generated	

From a quantitative standpoint, we used FID score to evaluate our model performance, and the model with 0.3 train/test split (with ResNet 9 and no identity learning) has the best performance. However, FID score is not a good indicator of style resemblance. In addition, the Inception v3 model is trained for general image classification tasks, and not specific for Chinese handwritten characters. Due to these limitations, a lower FID score does not necessarily indicate better models.

6 Conclusion/Future Work

From our experiments, we learned that a well-performing model for generating handwritten characters can be achieved after a decent amount of training using a small training dataset. It might be beneficial to train on larger datasets if data and computational resources are available.

For future work, we can explore variations of loss functions, such as applying linear decay to cycle consistent weight (10), since this might prevent the generator from producing realistic images during later stages of training.

We can also explore other quantitative metrics to evaluate our model. For content accuracy, we can use the HCCR-GoogLeNet model, which is trained specifically for classification of Chinese handwritten characters (12). For a better measurement of style discrepancy, we can adopt the style loss function in neural style transfer algorithm (1).

References

[1] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015.

[2] https://kaonashi-tyc.github.io/2017/04/06/zi2zi.html

[3] https://phillipi.github.io/pix2pix/

[4] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired imageto-image translation using cycle-consistent adversarial networks. ICCV, 2017

[5] B. Chang, Q. Zhang, S. Pan and L. Meng. Generating Handwritten Chinese Characters using CycleGAN. WACV, 2018

[6] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang. Casia online and offline chinese handwriting databases. In ICDAR. IEEE, 2011.

[7] http://www.nlpr.ia.ac.cn/databases/handwriting/Download.html

[8] https://hardikbansal.github.io/CycleGANBlog/

[9] http://cs230.stanford.edu/projects_fall_2019/reports/26256603.pdf

[10] https://ssnl.github.io/better_cycles/report.pdf

[11] Martin Heusel et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. 2018. arXiv: 1706.08500v6

[12] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In NIPS, 2016