[[1]](#footnote-1)

Create Your Own Calligraphy Artwork

Haoli Guo, Tao Jia and Yujie Zheng

*Abstract*—In this work, we aim to transform handwritten Chinese characters into calligraphy style characters on a stoke-by-stoke basis. We extracted stokes from a set of handwritten characters by identifying the primary orientation of each stroke. We then adopted neural network to classify the extracted strokes into 100 different categories. Using the classification model obtained, we classified strokes from calligraphy artworks as well. By identifying the strokes of a handwritten character and replacing it with the corresponding calligraphy stroke, we successfully created calligraphy style characters from handwritten characters while preserving the unique structures of the original characters.

# INTRODUCTION

C

hinese calligraphy has been an integral part of Chinese culture since its origination around four thousand years ago. Till this day, Chinese calligraphy, or as Joan Stanley-Baker, a professor of Art History at Taiwan National University of Arts puts it, “sheer life experienced through energy in motion”, still gives us great aesthetic pleasure. In this project, we build a platform that enables users to create their own Chinese calligraphy artworks without the need to hold the brush or even any knowledge about Chinese calligraphy. To be more specific, our algorithm takes in handwritten Chinese characters and produces personalized Chinese calligraphy artwork for the user.

A lot of existing work focuses on handwritten Chinese character recognition [1]-[2]. Our work, however, aims to recreate handwritten Chinese characters with features of Chinese Calligraphy artworks. We chose to perform our algorithm on stroke level instead of character level because this will allow us to preserve personal handwriting style of the users. This requires us to extract strokes from characters first. We adopted methods from [2] and successfully extracted [#] strokes from [#] handwritten characters. Using the extracted strokes as our training set, we obtain a classification model for the strokes using the MATLAB Neural Network Toolbox. We then extracted and classified calligraphy-style strokes using the trained model. After predicting the labels of the strokes extracted from an input character, we replace each stroke with the corresponding stroke written in a Chinese calligraphy font. The structure of the input handwritten character, which is mostly due to the habits of the writer, is preserved when we replace strokes. Figure 1 shows the workflow of our algorithm.

The structure of the report is as follows. We discuss the preprocessing of the dataset in section II. We illustrate our classification model in section III. Examples of application results are shown in section IV.

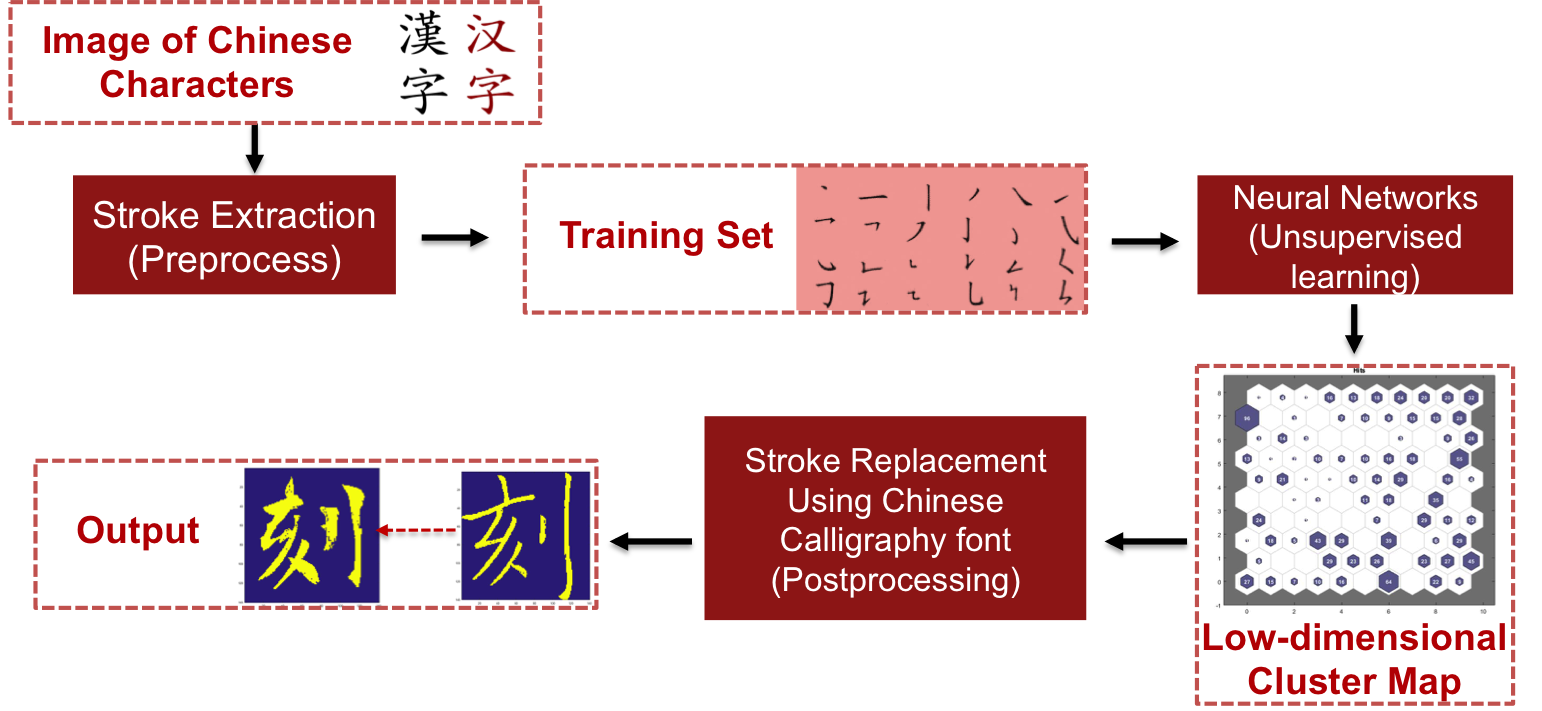


Figure 1: The workflow of the algorithm.

# Dataset and Preprocessing

## Dataset source

We used characters from [] as our main training set for classification model.

We also classified a set of Chinese calligraphy artwork from http://163.20.160.14/~word/modules/myalbum/,"陳忠建字庫", a collection of about 180,000 single characters of Chinese calligraphy.

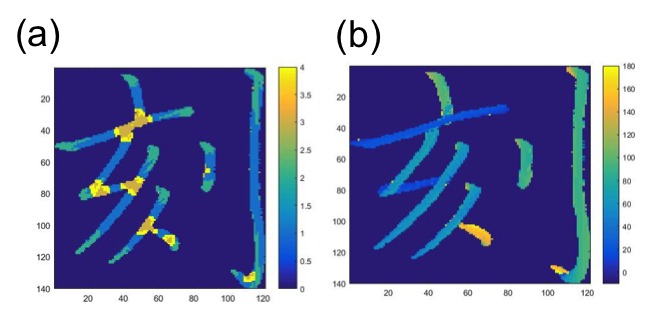
## Data Preprocessing – stroke extraction

We refer to the method from [2] to extract primary strokes from Chinese characters. We first distinguish edges, body parts and singular regions of a given character by computing its Point to Boundary Orientation Distance (PBOD) map (Figure 2 (a)). Different parts are characterized by different number of branches they connect – edges are only connected to one main branch, body connects two while singular parts connect three or more. For every pixel of the input character, we compute its orientation distance between this pixel and the boundary point along different directions. For edges, we expect to see only one peak in PBODs as they are only connected to one branch hence one primary orientation. For body parts, we expect to find two peaks. For singular parts, we should be able to find at least three peaks in PBODs. Using this method, we successfully identified key parts of Chinese characters.

The second step for strokes extraction is to compute Boundary-Boundary Orientation Distance (BBOD) to find the orientation of each pixel. BBOD refers to the distance between the two stroke boundary points intersected by a line passing through a certain pixel. Therefore, it is a function of pixel and orientation angle. For a certain pixel A, if BBOD(A,) has a peak at 0, it means the orientation of the stroke at pixel A is along the angle of 0. We store the peaks of BBOD for each pixel in a sparse 3-D matrix, bbod(x,y,0) (Figure 2 (b)).

In the third step, we separate the character into strokes by calculating the connected components in the BBOD matrix. The connected components are connected in (x,y) and have similar orientation. However, the real-world strokes in Chinese characters have twists and kinks, and different parts of the same stroke may have different orientations. Thus, this step is an over-separation to the strokes (Figure 2(c)).

Finally, we combine the over-separated strokes into the real, well-defined strokes with all the curves and kinks. We calculate the “similarity” between any two strokes using the rule of Chinese writing, and determine that the two strokes should be connected if the similarity is high. Specifically, two strokes have higher similarity if they share more pixels where either (1) it is not at crossing of strokes or (2) they have similar orientation. (Figure 2(d))



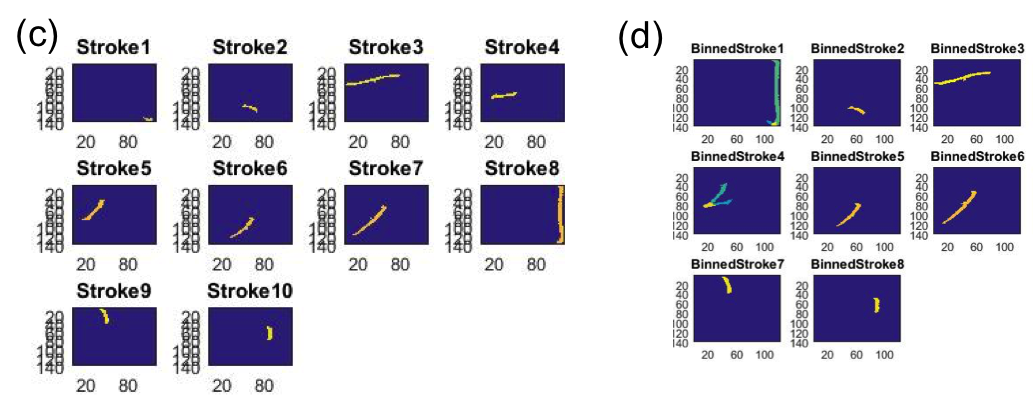


Fig. 2 Extracting strokes from “刻”. (a) PBOD map. (b) BBOD map. (c) Sub-strokes extracted as an intermediate result. (d) Connecting the substrokes and making legitimate strokes as training data.

# Classification Model

In this section, we illustrate our model for stroke classification. We adopt unsupervised classification model using the MATLAB Neural Network Toolbox and successfully classified strokes into 100 categories.

## Autoencoder neural network

The encoder maps the input to a hidden representation. The decoder attempts to map this representation back to the original input. In our model, an input image with 19600 dimensions (each stroke is stored as a 140 pixel by 140 pixel image) is represented in a space with 400 dimensions (Figure 3).

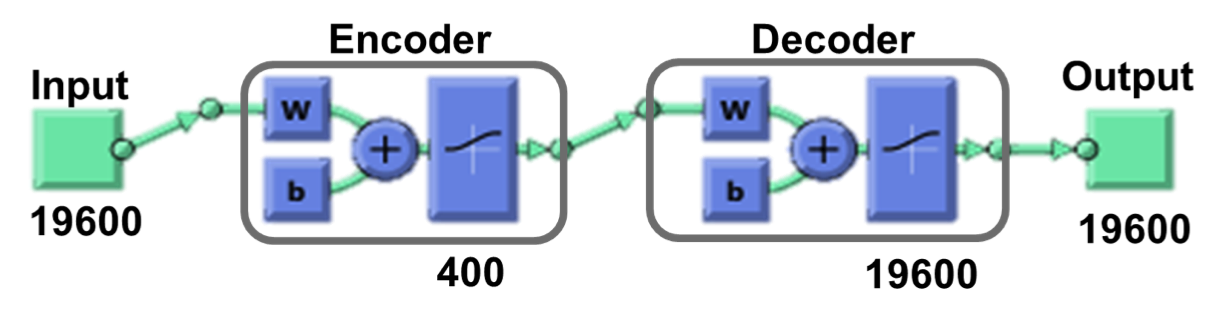


Figure 3: Autoencoder neural network.

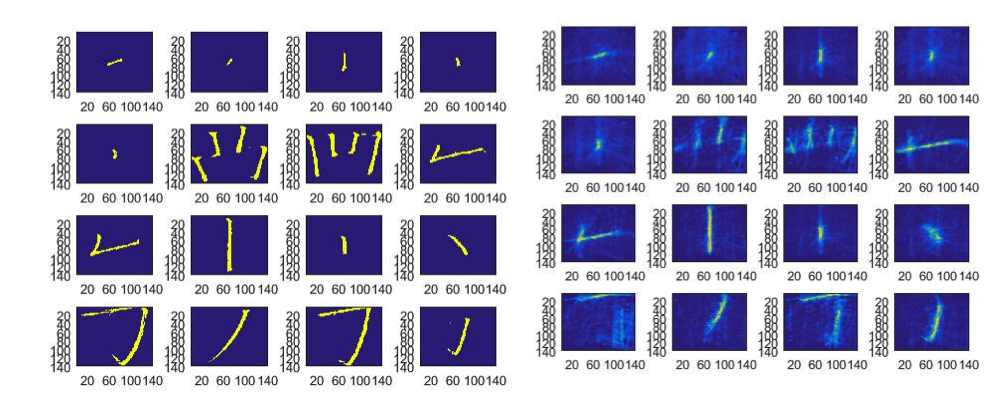


Figure 4: Input (left) and output (right) of autoencoder.

This step essentially reduces the dimension of the input data. Figure 4 shows the input and output results of the autoencoder. The output strokes preserve the primary shape of the input strokes. In the following steps, we will use the reconstructed strokes as our training set for the classification model.

## Self-Organizing Map (SOM)

SOM classify the input data according to how they are grouped in the space (Figure 5).

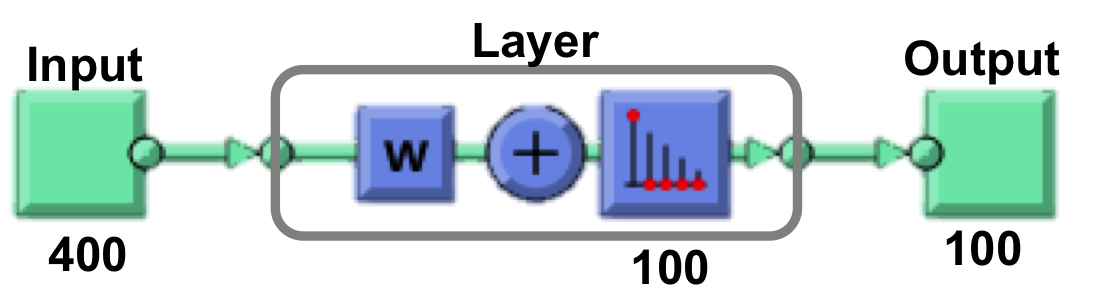


Figure 5: Cluster with SOM neural network.

Although there are only 31 basic strokes in Chinese characters, we set the number of neuron to be 100 in order to capture more characteristics of different types of Calligraphy. Figure 6 shows the clustering result. Figure 7 is an example of a cluster identified in our model.

Our classification model is highly successful as the misclassification rate is 0% for both training and test data. (1-2 sentences about how we did the error test).

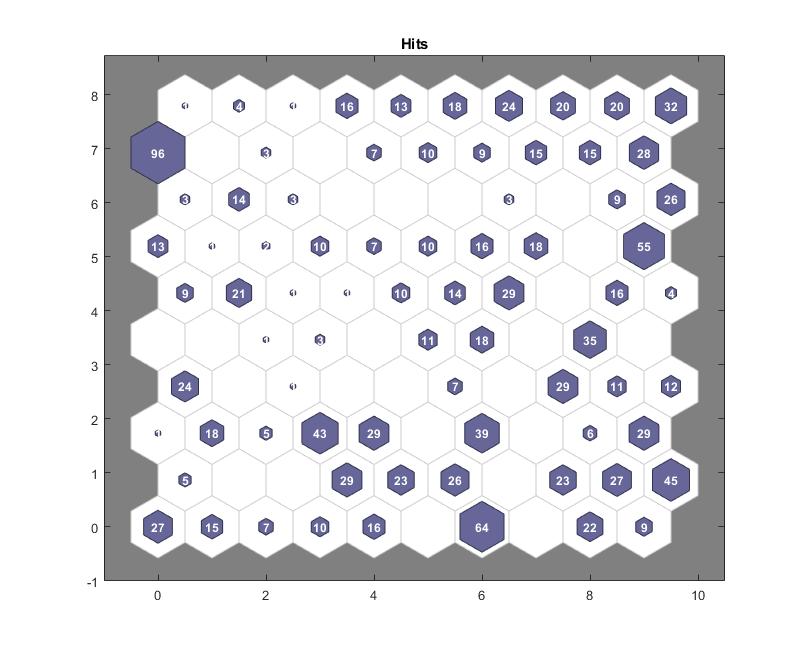


Figure 6: Clustering result by SOM neural network. 100 neurons were chosen. The number labeled for each cluster refers to []. The size of the hexagon is proportional to the number of strokes found in this cluster.

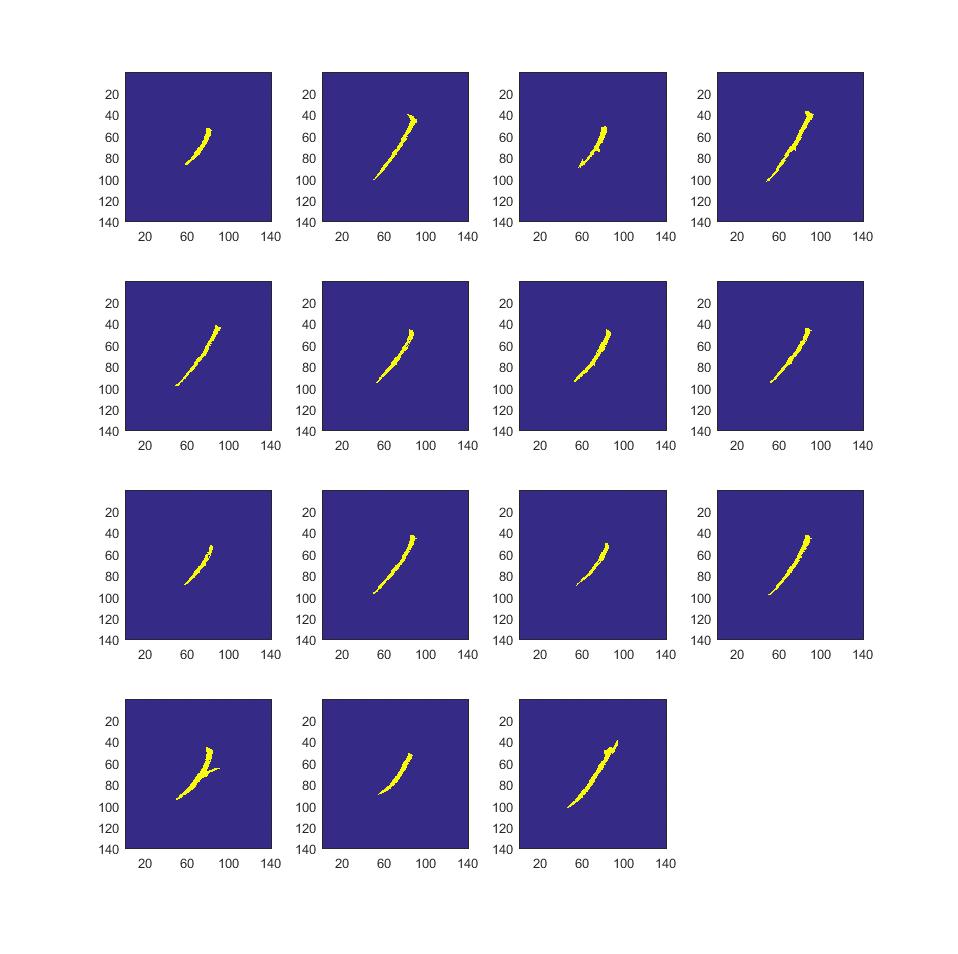


Figure 7: Elements from a cluster. In this cluster, all the elements are stroke “pie” with relatively short length.

## Classify Calligraphy Strokes

We use the trained classification model obtained in section B to classify calligraphy strokes as well. Stroke extraction for calligraphy artworks is harder than for handwritten characters. However, we only need to classify a selected set of calligraphy characters since we only need to ensure at least one calligraphy stroke in each classified category.

# Application

In this section, we illustrate how our model works to generate personalized calligraphy artworks. Figure 8 shows an example.

1. We first accept inputs from users.
2. We then extract strokes from these two characters using method illustrated in section II.
3. We ran the identified strokes by our classification model illustrated in section III.
4. We identify the clusters each strokes belong to and replace them with the calligraphy strokes stored in that cluster.
5. We output reconstructed characters.

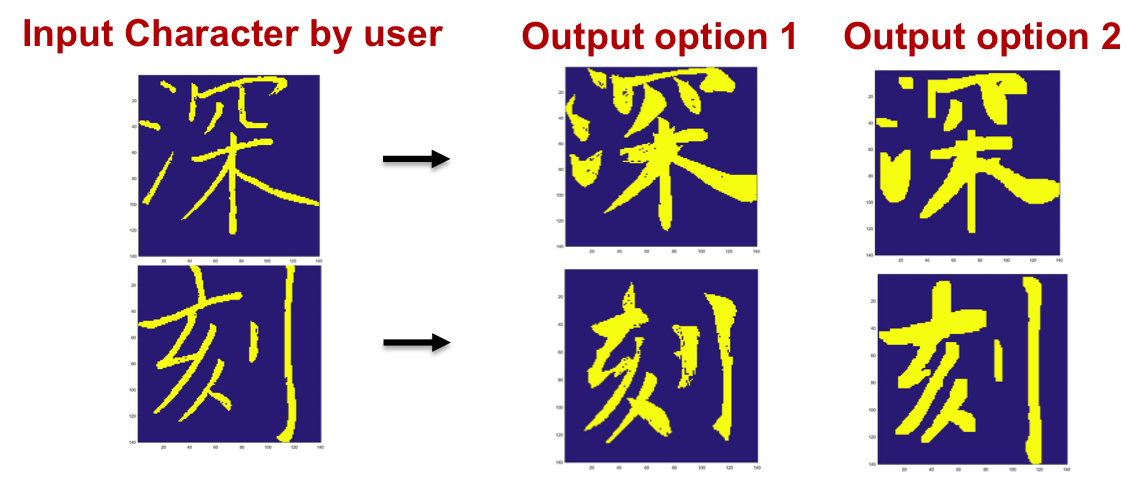


Figure 8: An example of input and output of our model. More than one output can be generated.

# Conclusion and Future Work

## In this project, we designed a platform which users can transform their handwritten characters into Chinese calligraphy artworks. Our stroke-by-stroke basis ensures that the originality and personalized features are well preserved in the output characters.

We only implemented our platform for one type of calligraphy due to the difficulties we encountered while trying to extract strokes from some calligraphy types. Better stroke extraction methods will enable us to work with a wider range of calligraphy types and add more variety to our output characters.

[some other things we can do or play with?]

Acknowledgment

We would like to thank Hao Sheng, the assigned TA for our project for providing us constructive and insightful comments throughout different stages of this project. We are particularly grateful for Hao for pointing us to the direction of the new MATLAB Neural Network Toolbox, which proves to be crucial for this project.

References

1. Lu, S.W., Ren, Y. and Suen, C.Y., 1991. Hierarchical attributed graph representation and recognition of handwritten Chinese characters. Pattern Recognition, 24(7), pp.617-632
2. Cao, R. and Tan, C.L., 2000. A model of stroke extraction from Chinese character images. In Pattern Recognition, 2000. Proceedings. 15th International Conference on (Vol. 4, pp. 368-371). IEEE.
3. Han, C.C., Chou, C.H. and Wu, C.S., 2008. An interactive grading and learning system for chinese calligraphy. Machine Vision and Applications, 19(1), pp.43-55.
4. Xu, S., Lau, F.C., Cheung, W.K. and Pan, Y., 2005. Automatic generation of artistic Chinese calligraphy. IEEE Intelligent Systems, 20(3), pp.32-39.
5. Xu, S., Jiang, H., Lau, F.C.M. and Pan, Y., 2007, July. An intelligent system for Chinese calligraphy. In Proceedings Of The National Conference On Artificial Intelligence (Vol. 22, No. 2, p. 1578). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

1. The SUNet ID for the authors are haoliguo, taoj and yjzheng respectively. [↑](#footnote-ref-1)