Create Your Own Chinese 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 stroke-by-stroke basis. We extracted strokes from a set of handwritten characters by identifying the primary orientation of each stroke. We then adopted autoencoder and self-organizing maps to cluster the extracted strokes into 100 different groups. Using the neural networks obtained, we clustered 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.**

I. INTRODUCTION

Chinese 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 choose 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 adopt methods from [2] and successfully extract 1297 strokes out of over 200 handwritten characters. Using the extracted strokes as our training set, we apply unsupervised learning and cluster the strokes into 100 groups with the aid of MATLAB Neural Network Toolbox. We then extracted and clustered calligraphy-style strokes using the trained neural networks. After identifying which clusters the strokes extracted from an input character belong to, we replace each stroke with the calligraphy-style strokes within the same cluster. The structure of the input handwritten character, which is mostly due to the habits of the writer, is

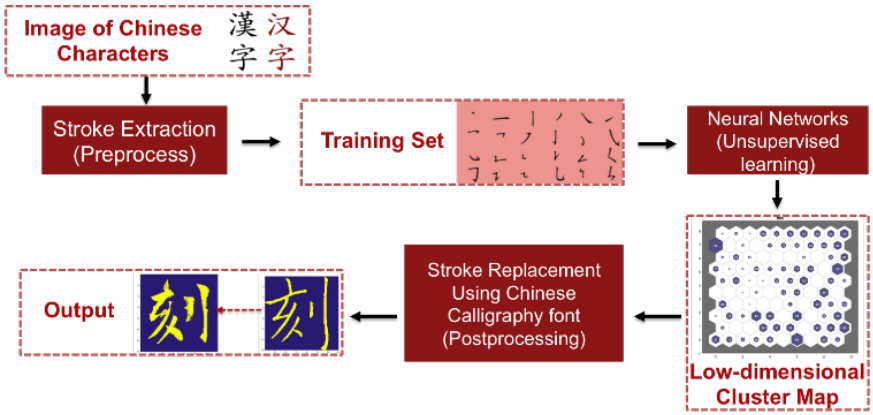

Figure 1: The workflow of the algorithm.

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 the training of neural networks in section III. Examples of application results are shown in section IV.

II. DATASET AND PREPROCESSING

2.1 Dataset source

We used handwritten characters from "陳忠建字庫" as the main training set for our neural networks. "陳忠建字庫" is a collection of about 180,000 single characters of Chinese calligraphy. We also classified a set of Chinese calligraphy artwork from "陳忠建字庫", from the work of the calligrapher Chu Suiliang.

2.2 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 stroke 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,θ) has a peak at θ_0, it means the orientation of the stroke at pixel A is along the angle of θ_0. We store the peaks of BBOD for each pixel in a sparse 3-D matrix, bbod(x,y, θ_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)).

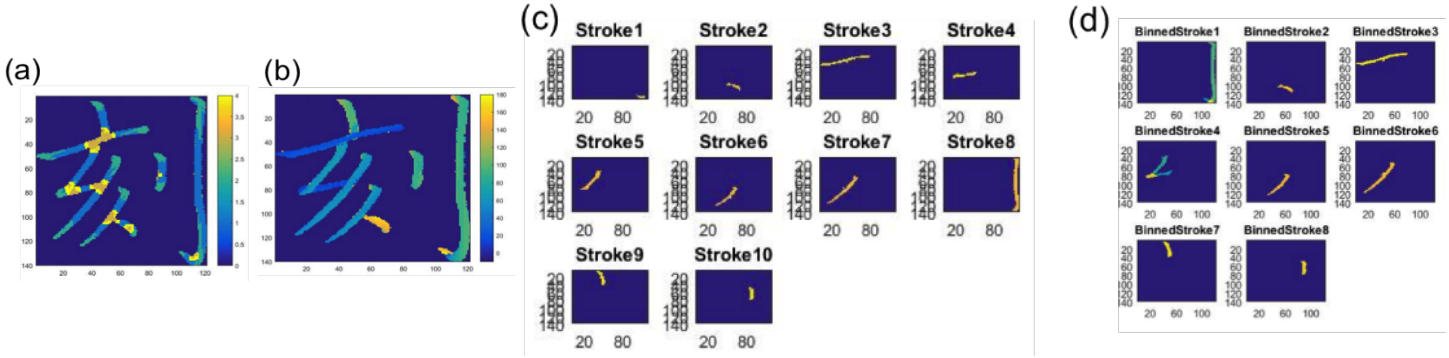

Figure 2: Extracting strokes from "刻". (a) PBOD map. (b) BBOD map. (c) Sub-strokes extracted as an intermediate result. (d) Connecting the substrokes and making legitimate strokes as training data.

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)).

III. NEURAL NETWORKS

In this section, we illustrate the training of neural networks for stroke clustering. We adopt unsupervised learning using the MATLAB Neural Network Toolbox and successfully clustered strokes into 100 groups.

3.1 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).

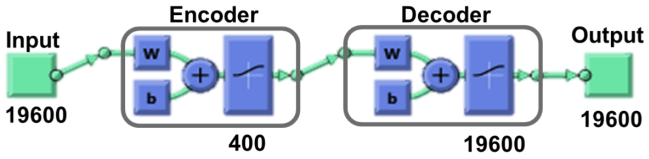

Figure 3: Autoencoder neural network.

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 step, we use the 400-demension feature matrix for stroke clustering.

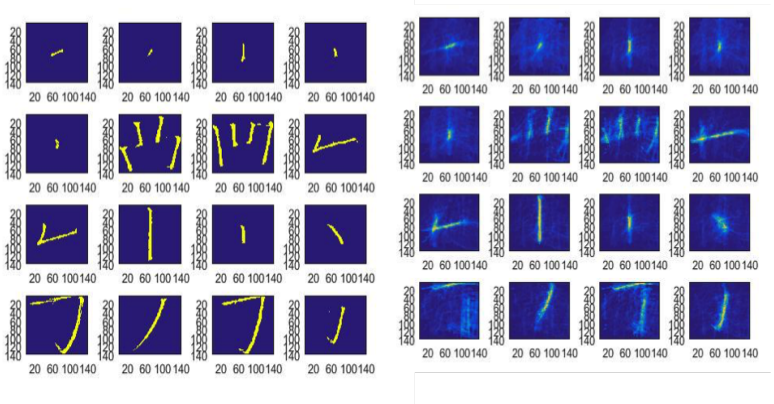

Figure 4: Input (left) and output (right) of autoencoder.

3.2 Self-Organizing Map (SOM)

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

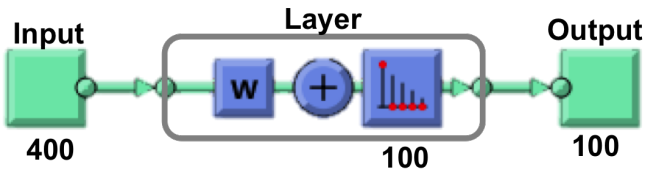

Figure 5: Cluster with SOM neural network.

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.

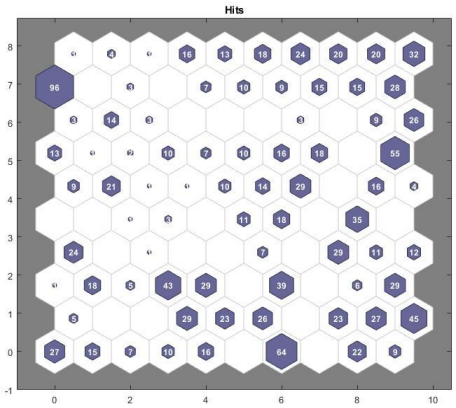

Figure 6: Clustering result by SOM neural network. 100 neurons were chosen. The number labeled for each cluster refers to the number of strokes in that cluster. The size of the hexagon is proportional to the number of strokes found in this cluster.

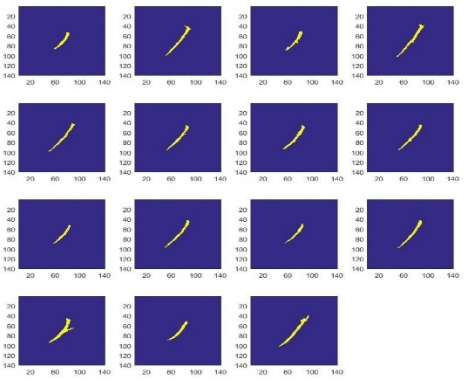

Figure 7: Elements from a cluster. In this cluster, all the elements are stroke "pie" with relatively short length.

To test the clustering results, we did softmax classification on the test set 1 (which has 325 handwritten strokes), and the test set 2 (which has 214 calligraphy-style strokes). The labels are chosen to be the stroke types (like "pie" or "heng"). Surprisingly, the test error defined this way is zero for both test sets. Namely, the predicted stroke labels are exactly that of labels recognized by human.

3.3 Clustering Calligraphy Strokes

We use the trained SOM in section 3.2 to cluster calligraphy strokes as well. Stroke extraction for calligraphy artworks is harder than for handwritten characters. However, we only need to cluster a selected set of calligraphy characters since we only need to ensure at least one calligraphy stroke in each cluster.

IV. 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 cluster the extracted strokes using neural networks illustrated in section III.
4. We identify the cluster each stroke belongs to and replace them with the calligraphy strokes stored in that cluster.
5. We output reconstructed characters.

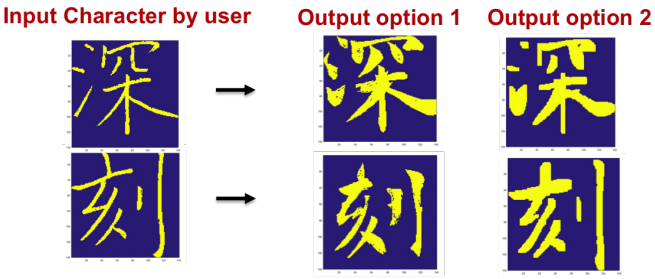

Figure 8: An example of input and output of our model. More than one output can be generated.

V. Conclusion and Future Work

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

We only implemented our platform for one type of calligraphy due to limited time. Also, our current stroke extraction method cannot work on the calligraphy styles with connected strokes. A stronger version of stroke extraction algorithm will enable us to work with a wider range of calligraphy types and add more variety to our output characters.

Besides the current application (creation of calligraphy with personal style), we can also use this work on an online judge of calligraphy practice. A user's calligraphy work can be compared to the existing calligraphy stroke by stroke, and a score can be generated by the similarity, from which the user will be able to improve his writing style.

VI. Acknowledgment

We would like to thank Prof. Andrew Ng and Prof. John Duchi for the amazing class. We also want 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.

VII. References

[1] Lu, S.W., Ren, Y. and Suen, C.Y., 1991. Hierarchical attributed graph representation and recosgnition 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.