Image-based Chinese Script Type Classification through ResNets and Character Recognition through Few-shot Learning on a Novel Traditional Chinese Calligraphy Dataset

Bing BAI
Tokyo Institute of Technology
2 Chome-12-1 Ookayama, Meguro City, Tokyo
bai.b.ab@m.titech.ac.jp

Chingwei HSU
Tokyo Institute of Technology
2 Chome-12-1 Ookayama, Meguro City, Tokyo
hsu.c.ac@m.titech.ac.jp

Abstract

We constructed a new dataset called the Traditional Chinese Calligraphy Dataset(TCCD). Character images of the calligraphic works from the Metropolitan Museum of Art are extracted and annotated manually. This dataset consists of 5 script types of Chinese characters: seal, clerical, cursive, semicursive, and standard. Then we employed the TCCD to train the ResNet for the script type classification and get a Top 1 Accuracy of 76.0%. Next, we conducted few-shot learning by training a Siamese network to analyze the similarity between characters to show the challenges in the area of calligraphy character recognition. Finally, we analyzed the potential usage situation and direction of the dataset. You can access the dataset and tutorials in `https://github.com/Bing-BAI/calligraphy-recognition-project`.

1. Introduction

1.1. Chinese Calligraphy

Chinese calligraphy (Fig. 1) was born more than three thousand years ago. Different ways of writing, known as script type were developed over time. There are five major script types: seal, clerical, cursive, semicursive, and standard, and each have its special characteristics.

1.2. Image-based recognization

Image-based calligraphy recognization through Deep Learning can contribute to the historical dating, classification, and conservation of cultural objects. Therefore construction of calligraphy datasets are necessary. We picked digital calligraphic works from the website of the Metropolitan Museum of Art as raw data to create the dataset. All works chosen are declared to be in the public domain.

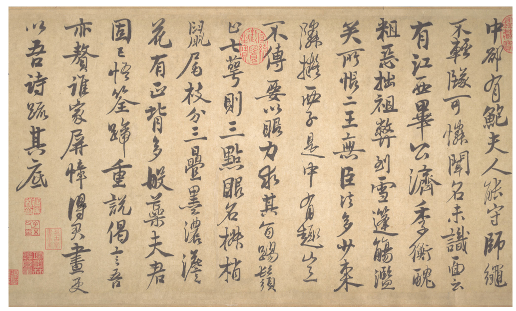


Figure 1. One page of semi-cursive script calligraphy: Poems on Painting Plum Blossoms and Bamboo from Zhao-meng jian.

1.3. Content Organization

The first section introduced the background and current situation of image-based Chinese character recognition. Then the second section explained how the dataset is created. Next in section 3, the TCCD was used to train the ResNets for the script type classification. After that, in section 4 we conducted few-shot learning through training a Siamese network. Finally, we summarized the usage of the dataset on the classification and character recognition tasks and the future direction it can be extended to.

2. Dataset Construction

We extracted and annotated character images from the calligraphies to build the Traditional-Chinese Calligraphy Dataset.

Firstly an open-sourced Chinese character detection library based on Yolo5 [4] were used to add rectangles for boxing different characters on a page of an entire calligraphic workpieces as in Fig. 2

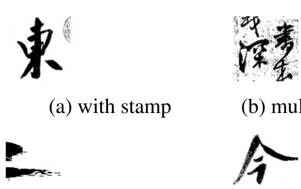
Figure 2. Boxing a single Chinese character in the entire page of a cursive calligraphy

2.1. Character extraction and Annotation

The character images need to be annotated manually to ensure its correctness. [2] What's more, some script types especially cursive are very hard to recognize even for native Chinese. So we have to define the rules to pick up and annotate the character images efficiently:

1. The boxed area should include only one Chinese character.
2. There should not have stamps on the character.
3. The target character should not include any part coming from other neighbor characters

Fig. 3 shows an example of picking character images in the bounding area. Fig. 3a includes a stamp besides the character, Fig. 3b includes multiple characters and Fig. 3c is not a complete character so they cannot be inserted into our dataset. Only image of Fig. 3d that properly includes a single character, we will annotate it in Chinese spelling and then add it to our dataset.



(a) with stamp (b) multiple characters

(c) not complete (d) proper

Figure 3. An example of character annotation

After the annotation. We constructed the TCCD with 2897 valid Chinese character images. All the bounded images are resized to 100*100 and stored in PNG format. Path to the image and the labels are stored in a csv file. Here is tuple of the data in the csv file.

3. Script Types Classification

Given an character image, we what our neural networks can recognize which kind of script type it belongs to. The 5 script types of character in the dataset are distributed as in Fig. 4

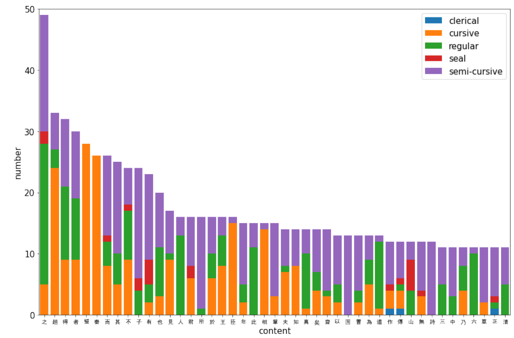


Figure 4. Distribution of the character type script

3.1. ResNet

Residual Networks, or ResNets can be used to is to train models for the script type classification. [5]

3.2. Result

Here is the training result of .This Part is implemented by my teammate Mr. Hsu. I confirmed the result.

The dataset was splited to train-validation-test with a ratioo of $8:1:1$. Since the data amount is limited so I think cross validation maybe a better choice.

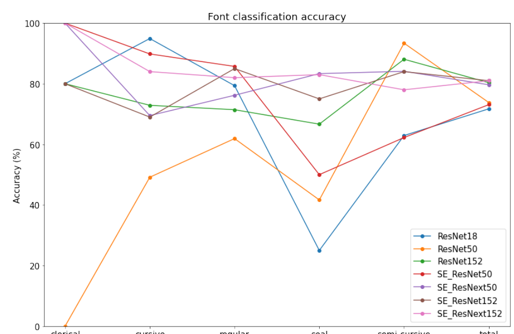


Figure 5. Classification On Resnet

4. Few-shot Learning

Few shot learning is a kind of meta-learning. First it learns a similarity function from large-scale training dataset which can tell us how similar two images are. Then apply the similarity function for prediction.

4.1. Siamese Dataset

We picked the "semi-cursive" characters as the source for creating Siamese Dataset. Because the "semi-cursive" type characters are widely distributed in the high-frequency-characters according to Fig. 4.

Then the "semi-cursive" characters appeared over 10 times (14 different characters that each of them appeared more than 10 times) are used to training data. Meanwhile the ”semi-cursive” characters appeared exactly 7 times (there are 5 characters that each of them appeared 7 times) are used as testing data.

Next, We need to prepare positive samples and negative samples in the filtered training data.

Positive samples tell the model what kind of things are of the same kind while negative samples tell what are different kinds.

Positive samples are obtained in this way: Randomly sample an image from the training set, then same another image from the same class. this is a positive sample pair and we label it as 1 it means the two are of the same kind.

Negative samples are constructed in this way: Randomly sample an image from the entire training set. Then exclude the class and randomly sample an image from the rest of the training set. label the pair as 0. 0 means the two images are different.

We have built the supporting set (for training) and testing set.(for evaluation) [3]

4.2. Siamese Network

The Siamese Network can be a similarity function. The network are defined as:

```
class SiameseNetwork(nn.Module):
    def __init__(self):
        super(SiameseNetwork,
            self).__init__()
        self.cnn1 = nn.Sequential(
            nn.ReflectionPad2d(1),
            nn.Conv2d(1, 4, kernel_size=3),
            nn.ReLU(inplace=True),
            nn.BatchNorm2d(4),

            nn.ReflectionPad2d(1),
            nn.Conv2d(4, 8, kernel_size=3),
            nn.ReLU(inplace=True),
            nn.BatchNorm2d(8),


            nn.ReflectionPad2d(1),
            nn.Conv2d(8, 8, kernel_size=3),
            nn.ReLU(inplace=True),
            nn.BatchNorm2d(8),

        )

        self.fc1 = nn.Sequential(
            nn.Linear(8*100*100, 500),
            nn.ReLU(inplace=True),

            nn.Linear(500, 500),
            nn.ReLU(inplace=True),

            nn.Linear(500, 5))

    def forward_once(self, x):
        output = self.cnn1(x)
        output =
            output.view(output.size()[0], -1)
        output = self.fc1(output)
        return output

    def forward(self, input1, input2):
        output1 = self.forward_once(input1)
        output2 = self.forward_once(input2)
        return output1, output2
```

Finally, apply the pairwise similarity upon the output and obtain a number. The final output measures the similarity between the 2 input images.

4.3. Contrastive Loss

Contrastive Loss [1] was used as loss function. Training loss of the Siamese network are shown in Fig. 6.

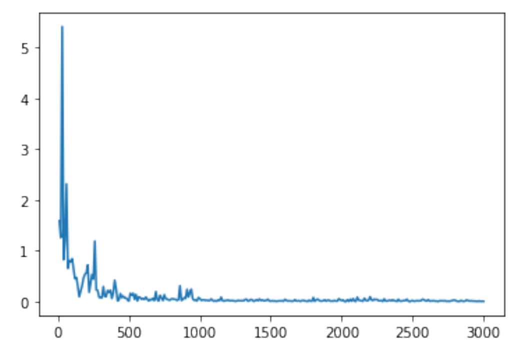


Figure 6. Training loss after 50 epoches.

4.4. Experiment Result

We selected 4 iterations (Fig. 7, Fig. 8, Fig. 9, Fig. 10,) to compare the similarity of 2 character images in the test set. Get an Top 1 accuracy of 62.85%.

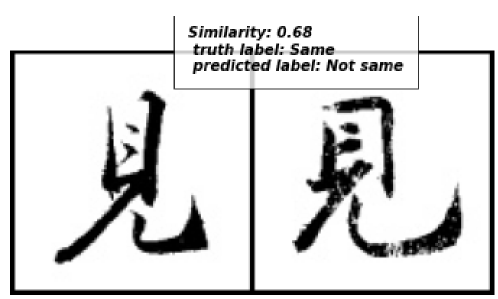


Figure 7. Same but wrongly predicted

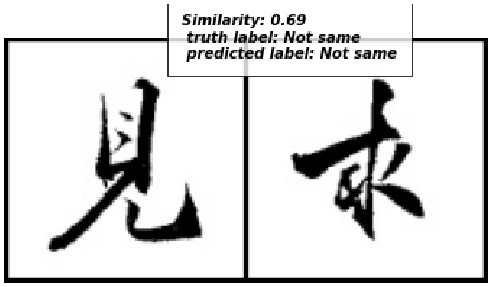


Figure 8. Not same and properly predicted

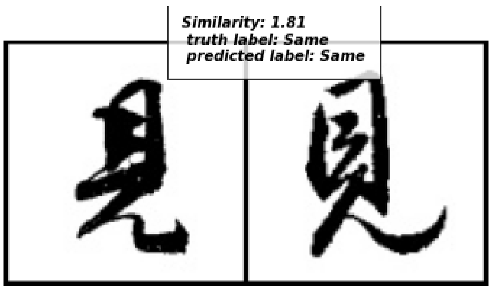


Figure 9. Same and properly predicted

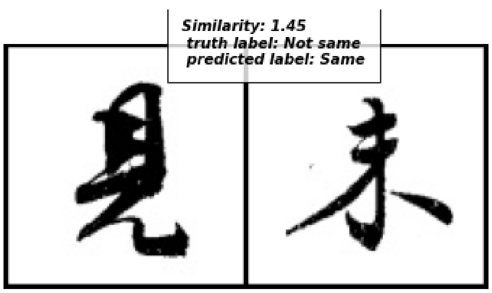


Figure 10. Not same but wrongly predicted

5. Conclusion

We created the Traditional Chinese Calligraphy Dataset. It includes 5 types of characters annotated with labels "content" (what the character is), "author", "script type".

It can be employed for various image-based machine learning tasks such as character recognition, calligraphy script type classification, author confirmation task. It will be published on Kaggle to help more researchers to contributes in this area.

References

[1] Contrastive loss explained, 2020. https://towardsdatascience.com/contrastive-loss-explaned-159f2d4a87ec. 3

[2] Bing BAI. Data annotator, 2022. https://colab.research.google.com/drive/1DW0Pgzrwhc8GERCNHYwDYPckuubl-mEC?usp=sharing. 2

[3] Bing BAI. Training siamese network, 2022. https://drive.google.com/file/d/1WIjdH9DKS5pzHDn9rezmdB_RRREmb9b7/view?usp=sharing. 3

[4] Eiuyc. Chinese-character-detection-and-recognition, 2021. https://github.com/Eiuyc/Chinese-character-detection-and-recognition. 1

[5] Ching wei HSU. Script type classification based on resnets, 2022. https://drive.google.com/file/d/1uBp8yTtl_U3qtNNOViFmALxe9Z-1FCr8/view?usp=sharing. 2