The Classification of Similar-looking Bronze Inscriptions “木”, “禾”, and “未” with Computer Vision Methods

Mengxuan Zhao[1]

[1]Computational Linguistics, the Graduate Center, CUNY

Abstract

The interpretation of Chinese bronze inscriptions has been a crucial subject for scholars for years. With the development of computer vision technologies, it is meaningful to try combining the issue with new methodologies. In this project, I built several Support Vector Machines with different feature sizes to classify three similar-looking bronze inscriptions. Multiple denoising and filtering methods are involved to extract effective features to describe the different shapes of different character types. With results that are 20% above the baseline, some of the methodologies are proved to be effective. Yet, future work are expected to explore further in this topic.

1 Introduction

Chinese bronze inscriptions are writing in a variety of Chinese scripts on Chinese ritual bronzes such as bells and tripodal cauldrons from the Shang dynasty (2nd millennium BC) to the Zhou dynasty (11th – 3rd century BC) and even later. The bronze inscriptions are one of the earliest scripts in the Chinese family of scriptsWikipedia: The Free Encyclopedia [2018].

The interpretation of bronze inscriptions is a crucial topic for philologists and archaeologists. Traditionally, researchers compare the rubbings of bronze inscriptions to reliable sources such as dictionaries, classic literature and other defined inscriptions. In other words, copies of the rubbings are the main study material in this field. Now, With the development of modern image recognition technologies, it might be a good idea to try to train computers to achieve such tasks to aid human scholars.

To recognize bronze inscriptions with computer vision techniques confronts several similar difficulties as human beings. First, the number of data is very limited. Even though China has an enormous number of culture relics, it is not comparable to the boost of information of today - the famous Mao Gong Ding contains the largest number of inscriptions, and the number is 497[1], which is only the length of a few tweets. Second, because the long history, lots of the bronzes are rusty. As a result, the images of the inscriptions become very noisy. Third, the configuration of bronze inscriptions are significantly unstandardized. This is due to geographical, political, and manufactural reasons. During the period of bronze inscriptions, China was composed of many different vassal states, and each of them had their own standard for writing system. According to Chen [2003], early bronze inscriptions were almost always cast, while later inscriptions were often engraved after the bronze was cast. That is another reason why bronze inscriptions varies in shapes and styles. All of the above result in the challenges of distinguish and recognize bronze inscriptions from rubbing images.

[1] $https://en.wikipedia.org/wiki/Mao\_Gong\_ding$

(a) Whole character (b) Radical

Figure 1: Examples of original inscription images

2 Methodology

The purpose of this project is to classify three similar-looking bronze inscription types (“木”, “禾”, and “未” in simplified form). There are several reasons for choosing the three characters. First, they are common words in bronze inscriptions. There are a comparatively large number of examples of various wiring styles found in bronzes of different vessel states in a broad range of time. Second, they are similar in configuration, while at the same time distinguishable with sufficient visual clue. Finally, all of the three characters have very simple structure, and this ensures that the image processing methods work with minimum possible disturbance.

2.1 Data

I use the data from Multi-function Chinese Character Database[2] . The images of bronze scripts are sorted by meanings in the website. I download all the images of “木”, “禾”, and “未”. However, since the number is not appropriate for training a machine learning model, I also collect compound characters that contain “木”, “禾”, or “未” as radical, and manually remove the unrelated part. Fig.1 gives both examples. All images are up to 75×75 pixels. The sample sizes are: 114 for “木”, 78 for “禾” and 24 for “未”.

2.2 Denoising

I take multiple steps to denoise the images. Fig.2 demonstrates the process. For the convenience of display, images are reversed as dark characters with light background. A γ-correction with $\gamma = 0.3$ is applied to desaturate the background noise. Otsu thresholding is applied to remove noise which is less-frequent colors. I use histogram equalization to bring back the salient target character. To make all images the equal size, I pad smaller images to 75×75 pixels. This normalization is necessary, and it makes convolution easier. Two denoise filters are applied on the padded image: a 3×3 median filter, and an α-trimmed filter of kernel size 5×5 and $\alpha = 5$. The two results are combined with equal weights. Sauvola thresholding of window size 75×75 is applied to the combined image. Finally, 3×3 median filter is applied one more time.

After the whole procedure, most of the background noise is removed from the images, while at the same time the character is remained. However, the result of denoising is not satisfying for images with severe noise, and that is one of the main factor influencing the classification performance.

2.3 Generating Artificial Data

As is mentioned above, the number of samples in each group is significantly unbalanced. The model would be very biased if it is trained with very unbalanced data. Following Simard et al. [2003], I apply two methods on images of “未”, and triple its size to 72 images.

[2] $https://humanum.arts.cuhk.edu.hk/Lexis/lexi-mf/bronzePiece.php$

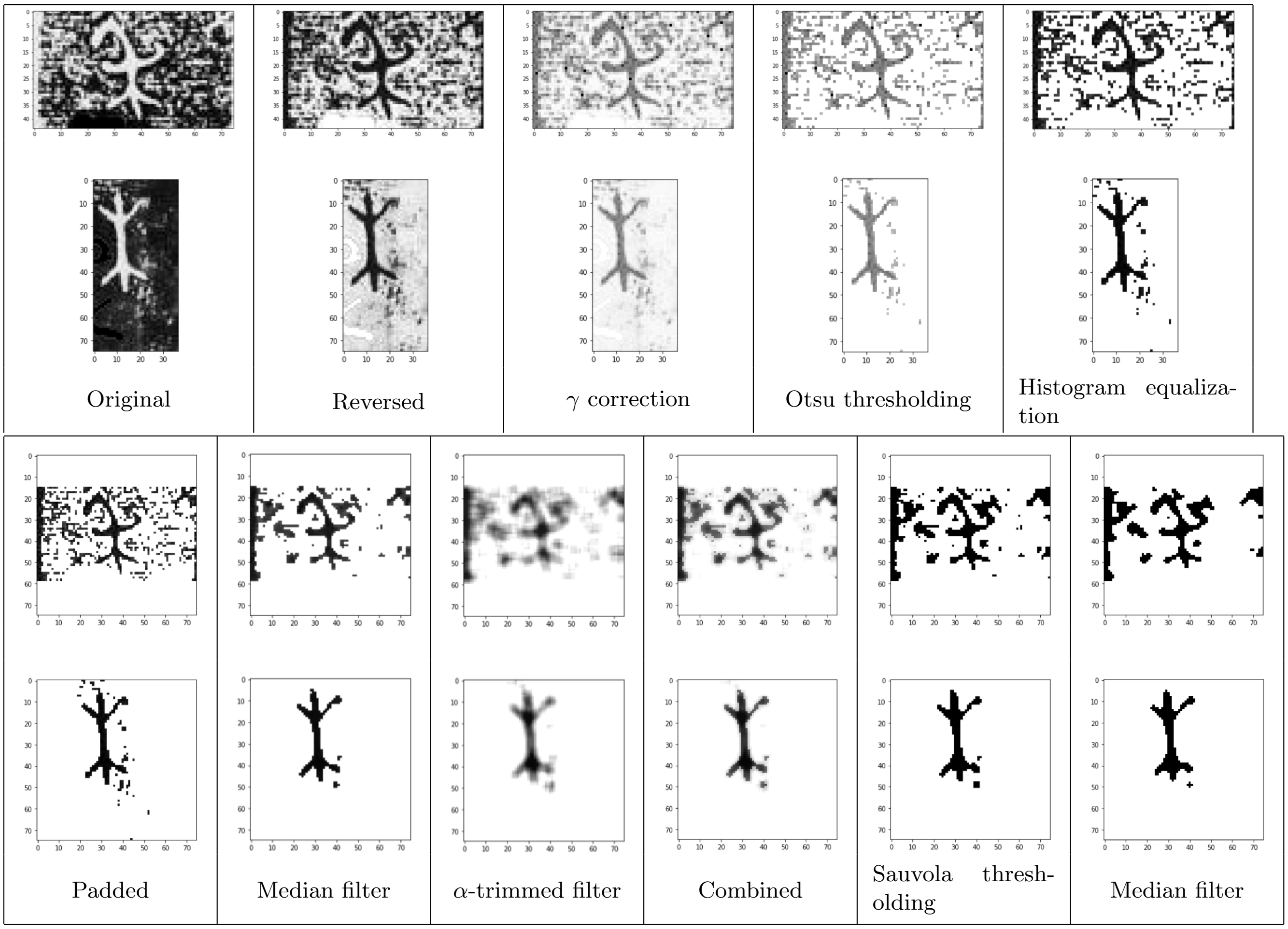
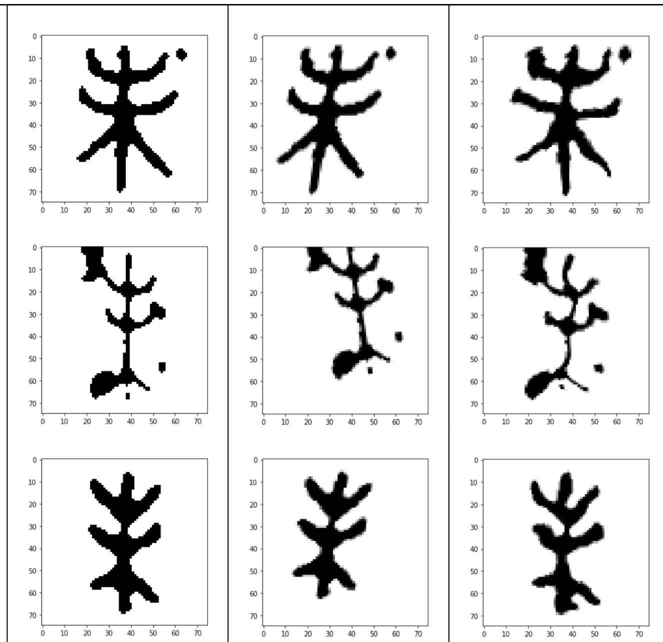


Figure 2: Procedures of image denoising

Affine transformation is combinations of simple geometrical transformation including rotation, shear, and scaling up or down. Elastic transformation generates random disformation with Gaussian σ convolution and distorts images based on a scaling factor α. In my project, I use $\sigma = 4$ and $\alpha = 34$ for elastic transformation, which are the same as Simard et al. [2003]. As for affine transformation, I generate random scale from 1 to 1.2, random rotation and shear from -0.2 to 0.2. Fig.3 displays the comparison among the original images, and images generated with the two methods.

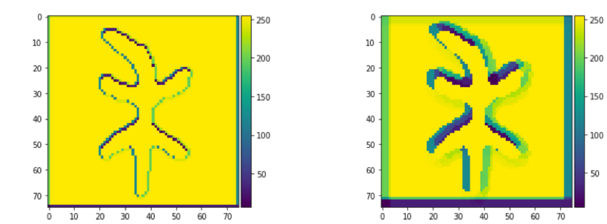
2.4 Features

I extract features with multiple methods. With Gray Scale Co-occurrence Matrices of distance = 1 and angle = [0, $\frac{1}{4}\pi$, $\frac{1}{2}\pi$, $\frac{3}{4}\pi$], I collect Haralick features including entropy, maximum probability, contrast, ASM and etc. A 3×3 and a 9×9 Local Binary Pattern are generated (Fig.4), and the histograms are used as features. Several filters are applied to extract shape information such as contour, edges and corners: Laplacian Pyramid (depth = 2), Sobel filters (kernel size = 3×3), Roberts filters (kernel size = 2×2), Laplacian filter (kernel size = 5×5), Harris filter (k = 0.1), filter banks (depth = 1, h0 = ($\frac{1}{2}$, $\frac{1}{2}$), h1 = ($\frac{1}{2}$, -$\frac{1}{2}$)). Some statistics are calculated based on the results of the filtered images as features, such as



Original	Affine transformation	Elastic transformation

Figure 3: Generating artificial data with 2 methods



(a) LBP 3×3 (b) LBP 9×9

Figure 4: Local Binary Pattern

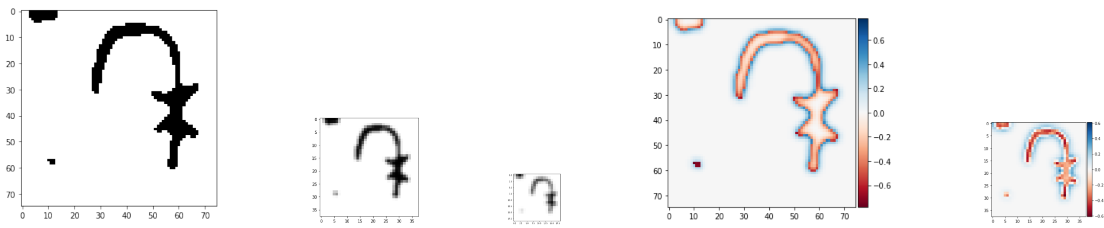
distributional probabilities of positive and negative values or non-zero pixels[3], mean and standard deviation of number of non-zero pixels per line or per image and etc. The results of the filters are demonstrated in Fig.5, 6, 7, 8 and 9.

3 Experiments and Results

I built a linear Support Vector Machine model to classify the data. The split ratio of training set and testing set is 80% and 20%, and the number of samples in each class is: 114 for “木”, 78 for “禾” and 72 for “末”. I calculate the baseline with ”dummy classifier” and it is 0.434.

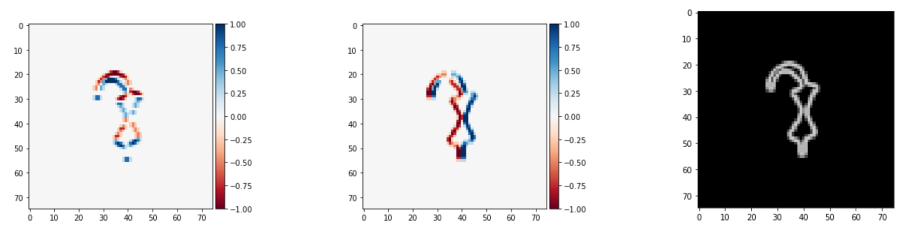
In experiment I, I use all of the 777 dimensions of features, and the LinearSVM with default parameters in Python scikit-learn package. In experiment II and III, I reduce the number of dimensions to 41 with

[3]For the convenience of demonstration I make all characters dark with light background. However, when calculating ”non-zero pixels”, I reverse the images, making the background black (pixel value = 0).



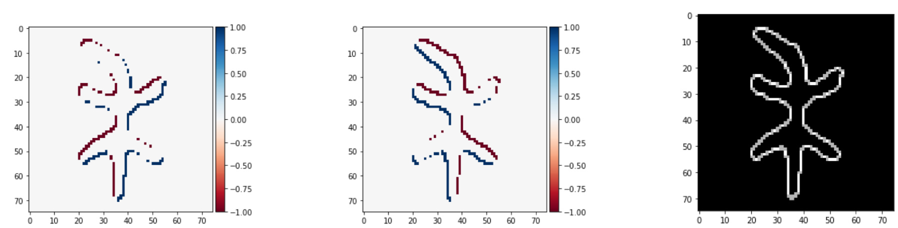
(a) (b) (c) (d) (e)

Figure 5: Laplacian pyramids



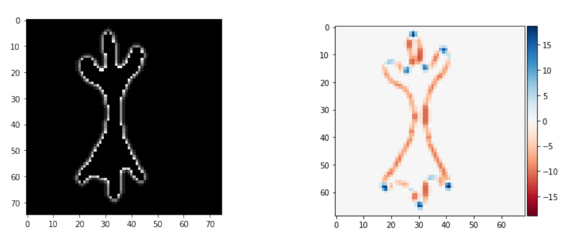
(a) Sobel filter: horizontal (b) Sobel filter: vertical (c) $\sqrt{(a)^2 + (b)^2}$ result

Figure 6: Sobel filters



(a) Roberts filter: diagonal (b) Roberts filter: diagonal (c) $\sqrt{(a)^2 + (b)^2}$ result

Figure 7: Roberts filters



(a) Laplacian filter (b) Harris filter

Figure 8

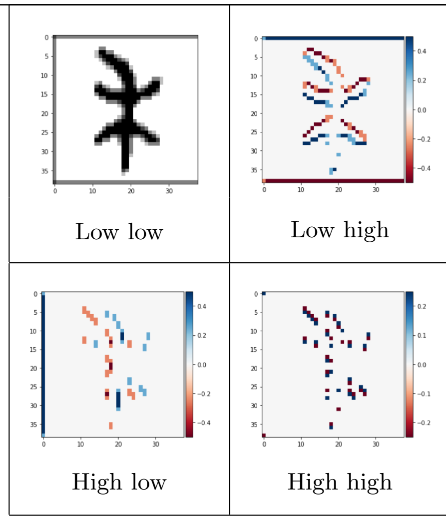


Figure 9: Filter banks

	No. of features	Precision	Recall	F1 Score
Exp.I	777	0.784	0.751	0.759
Exp.II	41	0.811	0.779	0.792
Exp.III	2	0.784	0.697	0.714

Table 1: Experiment I - III results

	No. of features	Average Accuracy
Exp.IV	777	0.602
Exp.V	41	0.636
Exp.VI	2	0.519

Table 2: Leave-One-Out results

"SelectKBest" function of χ^2 method (Fig.10), and to 2 (which explains 96.7% of variance) with PCA. The experiments are run a few times, and the best results are shown in Table1.

However, although the best results of experiment I - III looks satisfying, the models perform quite unsteadily. I then use the Leave-One-Out cross-validator, and the results are shown in Table2. The Leave-One-Out cross-validator takes one sample out as testing data and iterates through the dataset. I keep track of each result and calculate the average. The results of Leave-One-Out reflects the overall performance from a different perspective. As is shown, the model with 41 features performs the best, which is corresponding to the earlier experiments.

4 Discussion and Conclusion

With results of 20% above the baseline, the classification tasks are a successful try, yet can be improved in many ways. Noise remains to be a critical problem that derogates the accuracy of prediction. Considering the fact that the amount of bronze inscriptions discovered so far is not large, manual noise removal should be involved as the initial steps. That will significantly improve the performance. On the other hand, more effective features need to be explored. To solve the problem of small data size, I use artificial data generated

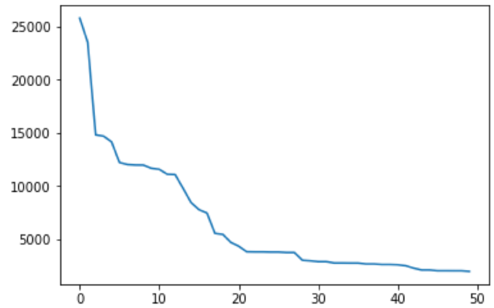


Figure 10: The first 50 distinctive features

with different methods, and that improves the performance of the models. Artificial data is a beneficial complement of the original data, and it should be used in similar projects. Additionally, with larger data size, it is worthy to try convolutional neural network and other deep learning methods.

References

Z. Chen. 秦系文字研究：漢字史的角度考察 *[Research on the Qin Lineage of Writing: An Examination from the Perspective of the History of Chinese Writing]*. 中央研究院·歷史語言研究所 [Academia Sinica, Institute of History and Philology Monograph], Taipei, 2003.

P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, volume 2, page 958–962, 2003.

Wikipedia: The Free Encyclopedia. Chinese bronze inscriptions, 2018. URL `https://en.wikipedia.org/wiki/Chinese_bronze_inscriptions`.