

Oracle-Bone Inscription Recognition Based on Deep Convolutional Neural Network

Guoying Liu*, Feng Gao

Department of Computer and Information Engineering, Henan Key Laboratory of Oracle Bone Inscriptions Information Processing and Collaborative Innovation Center of International Dissemination of Chinese Language of Henan Province, Anyang Normal University, Anyang, China.

* Corresponding author. Tel.: 86+03723300632; email: lgydatabase@163.com

Manuscript submitted July 10, 2018; accepted September 14, 2018.

doi: 10.17706/jcp.13.12.1442-1450

Abstract: In this paper, we investigate the recognition of oracle bone inscriptions (OBIs). By making use of the powerful ability of convolutional neural network (CNN) to describe image features, we design a CNN-based method to recognize OBIs. Our key insight is to build a CNN model for the recognition of OBI characters, which employs only image filters with 3×3 receptive field. Some commonly used techniques such as rectified linear unit (ReLU) and data augmentation have been employed to guarantee the accuracy of our method. Besides, the adaptive moment estimation (Adam) algorithm is used for training because of its straightforward implementation, computational efficiency and good empirical performance. In this paper, an OBI dataset with 44868 OBI images including 5491 different OBIs is established to testify the performance of the proposed method. Experimental results have shown the superiority to the state-of-the-art method.

Key words: Character recognition, deep convolutional neural network, oracle-bone inscription, oracle-bone inscription recognition, small kernel size.

1. Introduction

Oracle bone inscriptions (OBIs) are some of the oldest characters of Chinese words, which were hieroglyphic signs inscribed onto cattle bones or turtle shells with sharp objects about 3000 years ago. OBIs are very important to exploit the political systems, economic status and social lives of Shang Dynasty (about 1600 B.C. -1046 B.C.), the first dynasty recorded by Chinese words found so far. However, few people have the literacy of OBIs. The recognition of OBIs, which interdisciplinarily combines archaeology, history, philology and literature, requires people to have plenty of knowledge and years of experiences.

As shown in Fig. 1, some OBI examples are depicted. It is easy to find that each character looks like an undirected graph composed of intersections, lines or curves. Therefore, many researchers attempted to recognize OBIs by making use of the graph theory. By cascading two-level graph coding and one-level endpoint feature coding, Feng Li and Xinlun Zhou proposed to automatically recognize off-line OBIs [1], [2]. Shaotong Gu [3] described OBIs by topological coding and recognized them based on topological registration. Qingsheng Li [4] coded each OBI by the inverse adjacent matrix and recognized OBIs based on the theory of graph isomorphism.

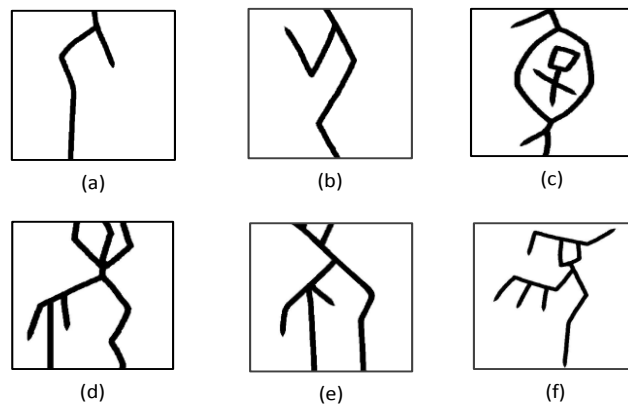


Fig. 1. Some examples of OBIs: (a) the OBI “人” (person); (b) the OBI “匕” (dagger); (c) the OBI “孕” (pregnancy); (d) the OBI “老” (old); (e) a variant of (d); (f) another variant of (d).

Some other researchers tried to recognize OBIs by different kinds of character features. Xiaoqing Lv etc. [5] used Fourier descriptor of curvature histogram to describe OBIs, Lin Meng etc. [6]-[8] employed Hough Transform to extract line features of OBIs, While Feifei Feng etc. [9] adopted Mathematical Morphology to extract character features. In [10], Jun Guo etc. proposed a hierarchical representation for OBIs, which combines a Gabor-related low-level representation and a sparse-encoder-related mid-level representation.

All methods mentioned above recognize OBIs in a traditional way, which is composed of three steps: image preprocessing, feature extraction and pattern recognition. However, due to the fact that there existed no unified specifications on how to write a certain OBI in the ancient Shang Dynasty, many OBIs have different variants, e.g., Fig. 1(e) and Fig. 1(f) are two variants of Fig. 1(d). In many cases, these variants usually have very different shapes and topological structures. Therefore, the traditional methods tend to show some obvious shortages due to their difficulty in employing the same set of pre-extracted features used for the accurate depiction of different variants.

With the boom of deep learning in recent years, the convolutional neural network (CNN) has been successfully applied into the field of offline Chinese character recognition [11]-[13]. Such a success should be owed to the fantastic properties of CNN about shift, scale and deformation invariance. Therefore, in this study, we employ CNN to recognize OBIs by making use of its powerful ability to describe image features of different OBI invariants. As far as we know, it is the first work to recognize OBIs based on the CNN model.

The remainder of this paper is organized as follows: Section II gives a brief review of CNN; Section III introduces the proposed architecture of OBI recognition; Section IV provides the experimental results and analysis; Section V discusses some limitations and concludes the paper.

2. A Brief Review of CNN

CNN [13], [14] is a hierarchical neural network model that is constituted by several cascaded layers including convolutional layers, pooling layers, ReLU layers, fully connected layers and loss layers. The convolutional layers serve as the feature extractors which producing a hierarchical representation of the input image, capturing both local and global features.

Let the input N feature maps of a convolutional layer be denoted as $\{I_n | n = 1, 2, \dots, N, I_n \in R^{M \times N}\}$, the t^{th} convolutional filter be denoted as $F^t \in R^{S \times S}$, and b^t be the bias parameter, the convolutional operation can be denoted by

$$O = f[(\sum_{n=1}^N I_n * F^t) + b^t], \quad (1)$$

where $f(\cdot)$ is the ReLU nonlinear activation function with the form of $f(x) = \max(0, x)$.

The pooling layer is always inserted between two successive convolutional layers and has the function of progressively reducing the spatial size of the input representation, resulting in great reduction of model parameters and the cost of computation. It operates independently on every depth slice of input and resizes it spatially with two most common forms: max pooling or mean pooling.

The loss layer defines the training methods of penalizing the deviation between the predicted and true labels. For the m^{th} sample, let $y^{m,k}$ denote its predicted probability of being classified as class k , and $y^m \in \{1, 2, \dots, K\}$ denote the true label, the cross-entropy loss function can be given by

$$Loss = -\frac{1}{M} \sum_{m=1}^M \sum_{k=1}^K [I(y^m = k) \log(y^{m,k})], \quad (2)$$

where M is the number of samples and $I(T) = \begin{cases} 1, & T = true \\ 0, & T = false \end{cases}$ is the indicator function. The softmax regression function is usually used to calculate the predicted probability $y^{m,k}$, and the stochastic gradient descent (SGD) algorithm is a common choice to minimize $Loss$.

3. Methodology

Considering the size of OBI images, we build an 8-layer network for the recognition of OBIs.

3.1. Architecture

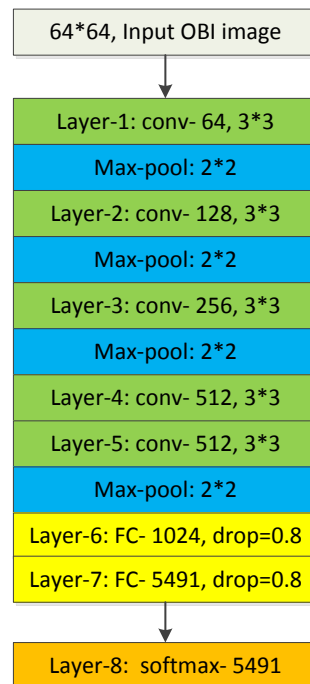


Fig. 2. The architecture of the employed neural network.

As shown in Fig. 2, the OBI image is passed through the cascaded convolutional layers, where only filters with 3×3 receptive field are employed. All of the convolutional strides are fixed to one. In order not to lose too much information, we gradually increase the number of feature maps from 64 (Layer-1) to 512 (Layer-4 and Layer-5) after each max-pooling operation. The pooling operation is widely used to obtain shift invariance, which is important to OBI recognition. In this architecture, the max-pooling operations use

a 2×2 window with stride 2 to halve the size of feature maps. The input OBI image is 64×64 , therefore, after 5 convolutional layers and 4 max-pool layers, the feature maps are flattened and concatenated into a vector with the dimensionality of 8192. Two fully-connected (FC) layers (Layer-6 and Layer-7), which have 1024 and 5491 hidden units respectively, are then followed. Finally, a softmax layer is used to perform the 5491-way classification. In other words, this architecture, which is totally different to all previous works mentioned above, aims to recognize as many as 5491 different OBI characters, instead of 261 [10] or less [6]-[8]. Besides, the dropout is only employed in the two FC layers, and the dropout probability is fixed to 0.8 for training.

3.2. Activation

Activation function is the key to adding non-linearity into the network. The leaky rectified linear unit (Leaky ReLU) is of the most popularity with the form:

$$f(x) = \begin{cases} \lambda x, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (3)$$

By using ReLU, the training process can converge to the optimum more rapidly, and the performance of the network can also be obviously improved compared to the conventional activations such as sigmoid and tanh. In our network, we set $\lambda = 0$, which makes the standard ReLU.

3.3. Initialization

For all layers in our network, we initialize all biases as zeros and initialize weights by using the zero-mean Gaussian distribution $N(0, \sigma^2)$, which is very popular in deep neural networks. In order to determine the value of σ , we employ the Xavier initialization algorithm [6], which uses different σ for different layers according to the number of input and output neurons for a specific layer. Let n_{in} and n_{out} respectively denote the number of input and output neurons for a certain layer, the variance of the zero-mean Gaussian distribution is defined as

$$\sigma^2 = \frac{2}{n_{in} + n_{out}} \quad (4)$$

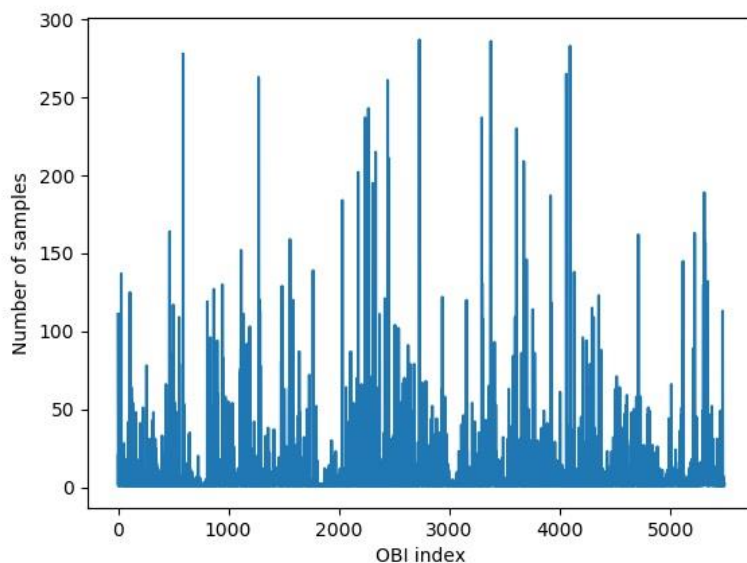


Fig. 3. Sample number of OBIs.

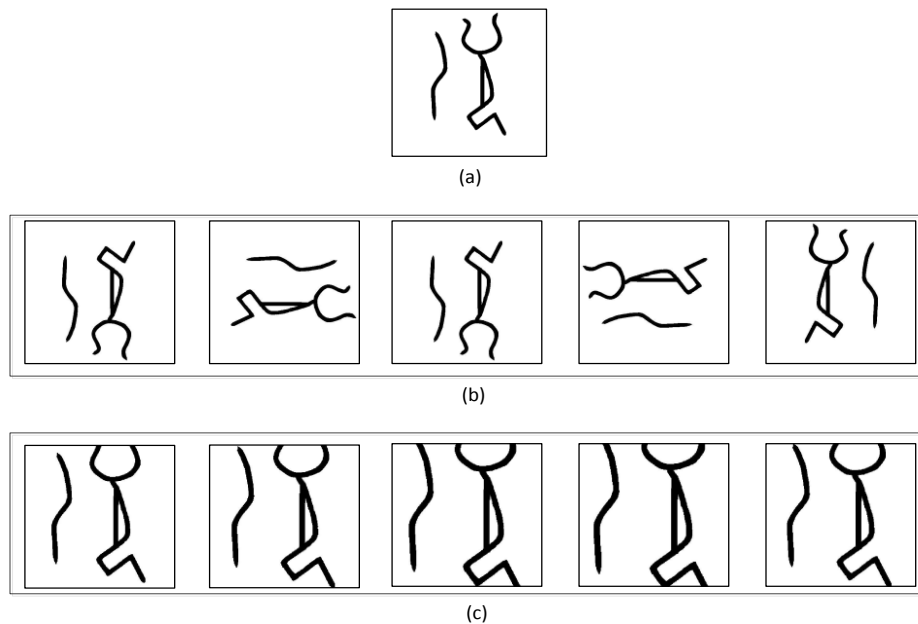


Fig. 4. An example of data augmentation: (a) OBI sample with index 02085; (b) results of randomly flipping (a) 5 times; (c) results of randomly cropping (a) 5 times.

3.4. Training

After proper initialization, we train our network by the adaptive moment estimation (Adam) algorithm [15], which computes individual adaptive learning rates for different parameters from estimates of first and second moments of the gradients. It is straightforward to implement, computationally efficient, showing good empirical performance and comparing favorably to other SGD (Stochastic Gradient Descent) based methods. There are some super parameters involved in this algorithm including the learning rate l_r , the exponential decay rate for the 1st moment estimates β_1 , and the exponential decay rate for the 2nd moment estimates β_2 . In this paper, we set $l_r = 0.1$, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Besides, the mini-batch size is set to be 128.

4. Experiments

We conduct experiments for the recognition of OBIs and compare our methods with some previously reported ones.

4.1. Dataset

With the help of OBI experts, we collect OBI images by manually drawing each line or curve of the scanned rubbings with handwritten board. In this dataset, there are 44868 OBI images including 5491 different OBIs. Fig. 1 shows some examples in this dataset. In order to make a clear understanding of our dataset, we count the number of each OBI in Fig. 3. As can be seen, the sample number of each OBI varies very much, the highest value reaches 277, while more than half ones are lower than 10. Such an imbalance undoubtedly increases the difficulty of training. Therefore, some preprocessing steps must be taken to augment OBI characters with fewer samples.

In this paper, for OBIs whose sample numbers are lower than 10, we augment the OBI images by randomly cropping and flipping samples 5 times. For instance, an OBI sample shown in Fig. 4(a) is augmented, whose results of randomly flipping and cropping are shown in Fig. 4(b) and Fig. 4(c) respectively.

4.2. Training Results of Our Network

In order to train our network, we divide our dataset into two half parts: training set and testing set. We set the max iteration number as 20000. During the training procedure, we record the values of loss and top 1 accuracy, which are shown in Fig. 5(a) and Fig. 5(b), respectively. As shown in the figure, with the iterations, the training accuracy increases and the training loss decreases gradually. Finally, when the iteration procedure stops, we obtain the top-1 accuracy value of 0.9375 and the loss value of 0.7846.

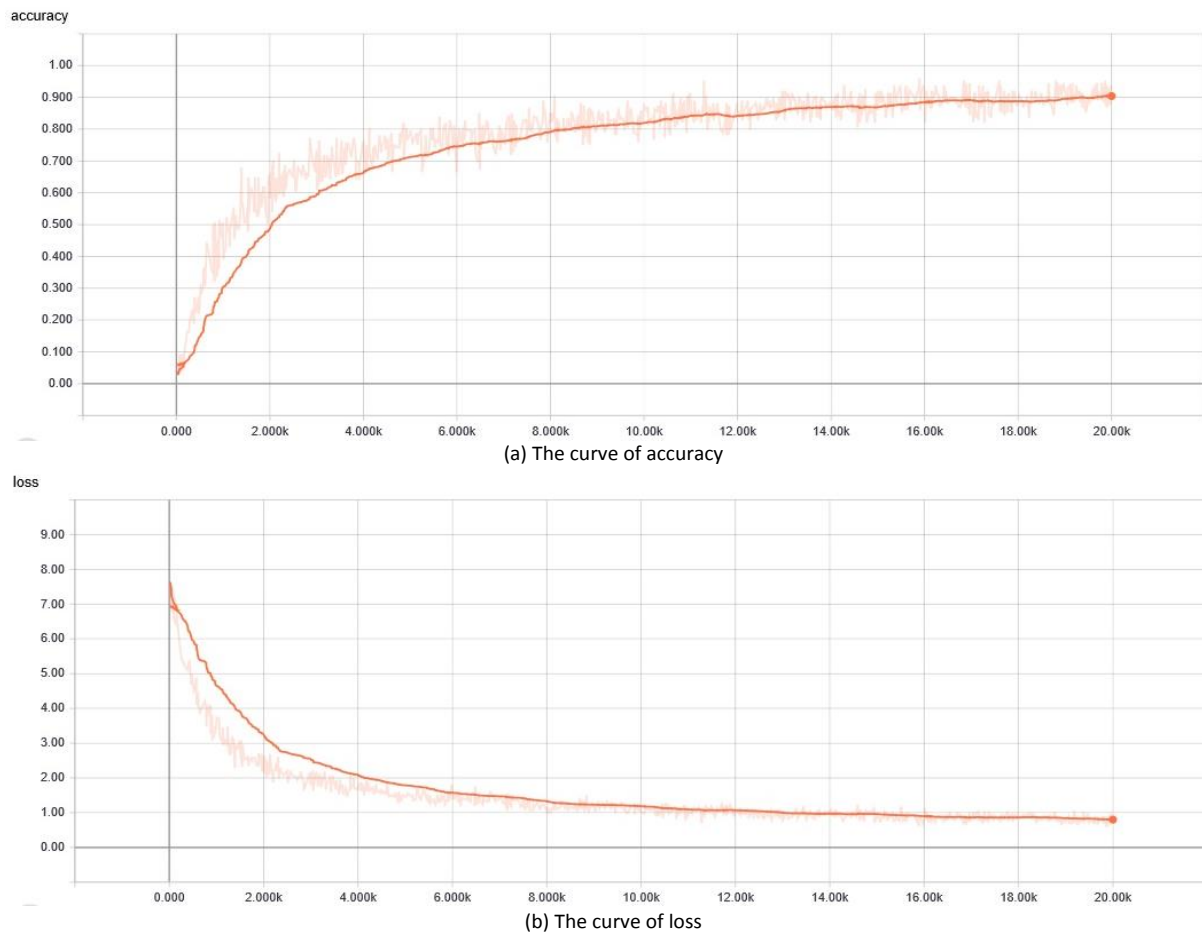


Fig. 5. The training process of our network: (a) the curve of accuracy; (b) the curve of loss function.

4.3. Testing Results of Our Network

We test the performance of our network on the testing set, and the top-1 and top-3 accuracy of our network are 91.56% and 97.16%, respectively. Besides, we record the top-100 most serious confusion instances in Table 1. In this table, OBI-1 and OBI-2 represent the indexes of two different OBIs, and CN is the number of instances belonging to OBI-1 that are misclassified as OBI-2. For example, the OBI “子” with the index of 005000 has 238 different samples, and the OBI “𠂇” with the index of 150075 has 105 different samples, and the corresponding images are shown in Fig. 6(a) and Fig. 6(b), respectively. Both OBIs look very similar; therefore, they have very serious misclassification. As can be seen from the underlined bold numbers in Tab. 1, there are 45 “子” instances are misclassified as “𠂇”. Therefore, from Table 1, we can find some OBI pairs that are difficult to be correctly recognized, which inspires us to make use of more contextual information to increase the accuracy.

4.4. Comparison with the State of the Art

Table 1. The List of Top 100 Most Serious Confusion

OBI-1	OBI-2	CN	OBI-1	OBI-2	CN	OBI-1	OBI-2	CN	OBI-1	OBI-2	CN
004000	004002	193	021002	021000	21	134014	093000	14	134014	134006	11
001004	001000	183	063022	063021	21	150005	150006	14	150010	014000	11
038000	039000	107	018035	018033	20	151000	070000	14	003002	003004	10
073014	073000	102	001000	001001	18	001125	001124	13	003013	003126	10
004002	004000	91	010150	010091	18	004022	004021	13	003013	003128	10
011001	011000	64	033007	033002	18	039000	039005	13	018031	018039	10
001004	001001	61	021000	021002	17	076000	020000	13	026001	026000	10
150015	150100	53	030000	033005	17	151121	001170	13	032003	032000	10
005000	150075	45	150010	069000	17	001014	001000	12	039000	038000	10
006059	058000	39	011000	150059	16	033003	033000	12	039011	039000	10
010007	010000	39	020000	014000	16	047000	046001	12	051025	051023	10
039005	039000	38	044001	045000	16	073000	073014	12	051025	051024	10
093001	093000	37	062003	062002	16	089008	089010	12	066001	150004	10
001001	001000	36	064000	141014	16	115008	115029	12	069000	150010	10
094002	094023	31	070011	150026	16	125003	018000	12	091099	048045	10
085048	085049	30	999056	999110	16	150079	003077	12	100000	073000	10
150010	020000	29	001141	083000	15	151097	151098	12	102010	102004	10
150075	005000	28	069000	020000	15	999083	038032	12	107023	107022	10
016012	016013	26	079006	079003	15	001120	001124	11	122066	122064	10
001013	001008	25	141014	064000	15	003080	003077	11	141002	141003	10
010000	010007	25	006098	006000	14	077033	077032	11	009000	009002	9
030000	034000	23	017000	017001	14	077115	077116	11	027004	027008	9
096012	096017	23	032002	031045	14	087024	087025	11	029004	083000	9
004002	058000	22	048075	048073	14	091099	020005	11	044000	046000	9
016000	015000	21	063062	063056	14	114016	114017	11	047000	048021	9

To the best of our knowledge, few of the state-of-the-art methods in OBI recognition are tested on a dataset including almost all the OBI characters. Most of them only employed datasets including no more than 300 OBI characters [1]-[8], [10]. Among them, Guo etc. [10] used a database composed of 20,039 OBI character instances belonging to 261 categories, which is the largest OBI dataset used in the previous works. To further investigate our method, we implemented Guo's work [10] and made a comparison of experimental results on this dataset. Following [10], we use two 1500-word codebook BoWs for the hierarchical representation, and use k-NN (k-nearest neighbors) and LR (Logistic Regression) as the classifiers. The experimental results are recoded in Table 2. It is appealing that our method achieves better recognition performance.

Table 2. Comparisons of Different Methods

Method	Accuracy (top 1)	Accuracy (top 3)
Hierarchical representation + k-NN	44.67%	53.19%
Hierarchical representation + LR	52.82%	63.37%
Our method	91.56%	97.16%

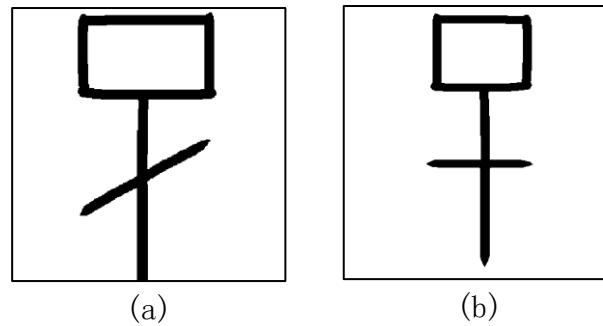


Fig. 6. Two OBIs with serious misclassification: (a) The OBI “子”(child); (b) The OBI “巳” (the sixth of the twelve Earthly Branches, which is an ordering system used throughout East Asia).

5. Conclusion and Future Work

In this paper, we propose an OBI recognition method based on the convolutional neural network. The method employs 5 convolutional layers with small kernel size (3×3) to extract image features and 2 fully connected layers to get the final description of OBI instances. In order to testify its performance, we collect an OBI image dataset with 44868 OBI images including 5491 different OBIs. The Stochastic Gradient Descent technique is employed to train the CNN-based model. The top-1 and top-3 testing accuracy has reached 91.56% and 97.16%, respectively, which shows its superiority to the hierarchical-representation based method proposed in [10].

Although our method has made some progress in the field of OBI recognition, there are still some issues needed to be further studied. For example, some OBIs have too few samples to train an accurate classifier which can avoid overfitting, resulting in serious misclassification with other OBIs. The study of all of this issue is left as our future work.

Acknowledgment

This work is supported by the National Social Science Fund of China (NSSFC: 16@ZH017A3), the National Science Fund of China (NSFC: U1504612), the Changjiang Scholars and Innovative Research Team in University of China, the key program of Henan Education Department (18B520001), and the National language committee scientific research planning project (YB135-50).

References

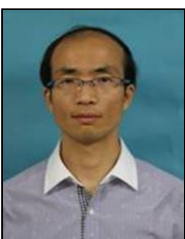
- [1] Li, F., & Zhou, X. (1996). Recognition of Jia Gu Wen based on Graph theory. *Journal of Electronics*, 18(suppl.), 41-47.
- [2] Zhou, X., Li, F., Hua, X., & Wei, J. (1996). A method of Jia Gu Wen recognition based on a two-level classification. *Journal of Fudan University (Natural Science)*, 35(5), 481-486.
- [3] Meng, L., & Izumi, T. (2017). A combined recognition system for oracle bone inscriptions. *Int. J. Mechatron. Syst.*, 7(4), 235-244.
- [4] Li, Q., Yang, Y., & Wang, A. (2011). Recognition of inscriptions on bones or tortoise shells based on graph isomorphism. *Computer Engineering and Applications*, 47(8), 112-114.
- [5] Lv, X., Li, M., Cai, K., Wang, X., & Tang, Y. (2010). A graphic-based method for Chinese oracle-bone classification. *Journal of Beijing Information Science and Technology University*, 25(Z2), 92-96.
- [6] Meng, L. (2017). Two-stage recognition for oracle bone inscriptions. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10485, 672-682.

- [7] Meng, L., & Izumi, T. (2017). A combined recognition system for oracle bone inscriptions. *Int. J. Mechatron. Syst.*, 7(4), 235-244.
- [8] Meng, L. (2017). Recognition of oracle bone inscriptions by extracting line features on image processing. *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods* (pp. 606-611).
- [9] Feng, G., Gu, S., & Yang, Y. (2013). Feature extraction method of Oracle-bone inscriptions based on mathematical morphology. *Journal of Chinese Information Processing*, 27(2), 79-85.
- [10] Guo, C., Wang, E., Roman, R., Chao, H., & Rui, Y. (2016). Building hierarchical representations for oracle character and sketch recognition. *IEEE Trans. Image Process.*, 25(1), 104-118.
- [11] Zhong, Z., Jin, L., & Xie, Z. (2015). High performance offline handwritten Chinese character recognition using GoogLeNet and directional feature maps. *Proceedings of the 13th International Conference on Document Frontiers in Handwriting Recognition* (pp. 846-850).
- [12] Xiao, X., Jin, L., Yang, Y., Yang, W., Sun, J., & Chang, T. (2017). Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition. *Pattern Recognit.*, 72, 72-81.
- [13] Zhang, X. Y., Bengio, Y., & Liu, C. L. (2017). Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognit.*, 61, 348-360.
- [14] Cun, Y. L., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., & Hubbard, W. (1990). Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.*, 2(2), 396-404.
- [15] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *Computer Science*, 1-15.



Guoying Liu was born on July 16, 1979. He received the Ph.D. degrees in geoscience and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China, in 2009. His research interests mainly focus on machine learning and computer vision, in particular, pattern recognition in digital images.

From 2014 to 2015, he worked as a post-doctoral researcher with the University of New Brunswick, Fredericton, Canada. He is currently a professor with Department of Computer and Information Engineering, Henan Key Laboratory of Oracle Bone Inscriptions Information Processing, and Collaborative Innovation Center of International Dissemination of Chinese Language of Henan Province, Anyang Normal University, China.



Feng Gao was born on June 2, 1976. He received the Ms. degree in computer science and technology from the Department of Computer, Shanxi Normal University, China, in 2007. His research interests mainly focus on pattern recognition in OBI image processing.

He is currently a lecture with Department of Computer and Information Engineering, Henan Key Laboratory of Oracle Bone Inscriptions Information Processing, and Collaborative Innovation Center of International Dissemination of Chinese Language of Henan Province, Anyang Normal University, China.