Oracle Character Image Retrieval by Combining Deep Neural Networks and Clustering Technology

LIU Guoying, WANG Yangguang

***Abstract*—In this paper, we study the description and retrieval of oracle character images. A great challenge is that there is a big difference between the variants of the same oracle character in character patterns, bringing deep negative impact on the retrieval of oracle character images. To solve this problem, in this paper, an image retrieval method is proposed by combining deep neural networks (DNN) and clustering technology. Firstly, for making full use of the powerful image representation ability of DNN, the deep convolutional neural network is used to extract character-level image features. Then, in order to solve the problem of retrieving variants of oracle characters, the Kmeans++ clustering algorithm is used to divide the oracle character dataset into different subsets of variants. Finally, based on the newly divided data set, a fully connected neural network is trained to get the variant-level features, which are used for the retrieval of oracle character images. Compared to the traditional methods and the DNN-based methods, this method achieves the optimal results on the data set of oracle character images.**

***Index Terms*—Oracle character, image retrieval, deep learning, clustering technology.**

I. Introduction

CONTENT-BASED image retrieval (CBIR) presents the technologies allowing to organize digital images by their visual features, which is a hot topic at present and attracts a lot of attention. It retrieves the most visually similar images to a given query image from a database of images. CBIR algorithms can be classified into traditional ones and deep neural networks based ones. In the traditional algorithms, many hand-crafted image features, such as SIFT[1], SURF[2], GIST[3], LBP[4], and HOG[5], are used for image retrieval. These features are usually coded by BoW[6], FV[7], SPM[8], and VLAD[9] to get compact and accurate image representations for image retrieval.

In recent years, with the development of deep learning, a variety of deep neural networks have been proposed and successfully applied to image classification and recognition [10][11][12][13]. Inspired by the successes, many researchers attempted to build CBIR systems by resorting to deep learning. In [14], J. Wan et al. made a comprehensive study about the CBIR systems based on deep learning. They investigated a framework of deep learning for CBIR by applying convolutional neural networks (CNN) for learning feature representations from image data. From the experimental results, they revealed that CNN-based image features are helpful for capturing high semantic information in raw pixels and consistently outperforming conventional hand-crafted features. In [15], J. Y. Ng et al. found that intermediate layers or higher layers with finer scales would produce better results for image retrieval, compared to the last layer. Y. Gong et al. [16] proposed the multi-scale orderless pooling (MOP-CNN) to improve the invariance of CNN activations, resulting in a generic representation for instance-level retrieval. Z. Liu et al. [17] proposed FashionNet to learn local and global clothing features, which were successfully applied to in-shop and consumer-to-shop clothing retrieval. A. Gordo et al. [18] leveraged a deep architecture trained for the task of image retrieval by producing a global and compact fixed-length representation for each image. Q. Zheng et al. [19] proposed an end-to-end image retrieval system by redesigning the famous VGGNet [11] and making use of the differential learning method. X. Wang et al. [20] presented a multi-similarity loss for deep metric learning, and established a General Pair Weighting (GPW) framework which obtained satisfactory performance on four image retrieval benchmarks. All of these methods mentioned above have achieved better results in their realms.

Oracle characters are the earliest and the most holonomic characters of ancient Chinese characters that have been discovered in China. Recording the political, economic, social life and other aspects of the Shang dynasty (about 1600 B.C. - 1046 B.C.), they play an important role in the study of the development of Chinese characters and ancient China history. These ancient characters engraved on the oracle bones provide very useful information for uncovering the truth of major events in ancient China. The same characters on different oracle slices can be easily found by image retrieval, which undoubtedly establishes a specific relationship between different oracle bone slices, and provides more abundant clues for the basic topics in this field, such as the interpretation of oracle characters and the joint methods for different oracle bone slices. However, the heteromorphism leads to many variants of the same oracle character, which brings great difficulties to image retrieval. Fig.1 shows four variants of the oracle character “right”. It is obvious that their character patterns change very much.

Clustering analysis is a technique of grouping a set of data into many subsets consisting of similar patterns. In the past decades, many famous clustering algorithms such as K-Means[21], FCM[22][23], GMM[24], Meanshift[25][26], and BIRCH[27] have been proposed. They employ different strategies to merge data with similar characteristics into a group. Namely, the intra-class difference can be greatly decreased by applying clustering analysis. Such a basic

Manuscript received August 02, 2019; revised February 01, 2020. This work was supported by the joint fund of National Natural Science Foundation of China (NSFC) and Henan Province of China under Grant U1804153, and partly supported by the Program of Innovative Research Team (in Science and Technology) in University of Henan Province of China under Grant 17IRTSTHN012.

L. Guoying is with the Department of Computer and Information Engineering, Anyang Normal University, Anyang 455000, China. He is also with Key Lab of Oracle Bone Inscriptions Information Processing of Ministry of Education, Anyang Normal University, Anyang 455000, China. Phone: 86-0372-3300039; fax: 86-0372-2900001; e-mail: guoying.liu@aynu.edu.cn

W. Yangguang is with the Department of Computer and Information Engineering, Anyang Normal University, Anyang 455000, China. ECBIR-CNNfeatures-valid-mail:1843127373@qq.com

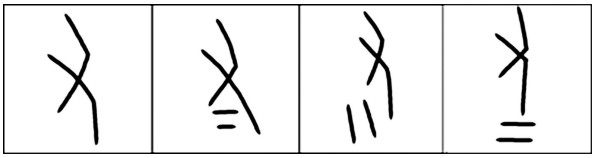

Fig. 1. Four variants of the oracle character "right".

merit impels their successful application to many fields, e.g., image segmentation[28], speaker recognition[29], and target tracking[30]. By resorting to clustering analysis, the intra-class differences between samples of the same oracle character can be reduced, which will improve the performance of image retrieval.

In this study, we combine the advantages of deep neural networks and clustering technology to address the problem of oracle character image retrieval. Firstly, the deep CNN models are used to obtain character-level features of oracle character images. Then, the clustering technology is used to divide image features of the same oracle character into different subsets, each of which is composed of image features belonging to the same variant character. Finally, the deep fully connected network is further exploited to obtain variant-level image features, which will be used for image retrieval. The main contributions of this work include two aspects: 1) deep neural networks are used to obtain two levels of oracle character image features for image retrieval, and 2) the combination of deep neural networks and clustering analysis is used to address the difficulties of oracle character variant retrieval.

The rest of this paper is organized as follows. In Section II, we build the framework of our oracle character image retrieval algorithm. Sections III discusses the details of our algorithm. Extensive experimental results are given in Section IV to show the performance of the algorithm. And Section V concludes this study.

II. New framework of oracle character image retrieval

In order to retrieve oracle character images accurately, we try to build image features that can distinguish different variants of the same oracle character. The proposed image retrieval framework is shown in Fig.2, where the Deep Convolutional Neural Network (DCNN) model and the Deep Fully Connected Network (DFCN) model are concatenated to generate two levels of image features, i.e., the former generates character-level features and the latter generates variant-level ones. The core of this algorithm is depicted in a red dotted rectangle, which is also our main contribution. It consists of three modules connected in turn: DCNN, Kmeans++, and DFCN. They works in different manner during the training stage and the retrieval stage.

During the training stage (modules connected by black solid arrows in the figure), all training samples are input into DCNN to get character-level image features, which are further divided into subsets by the Kmeans++ algorithm[31], each of which corresponds to a variant of oracle characters. Then the new division is employed as the training data for DFCN, whose outputs describe images on the variant-level level and form the database of image features for retrieval.

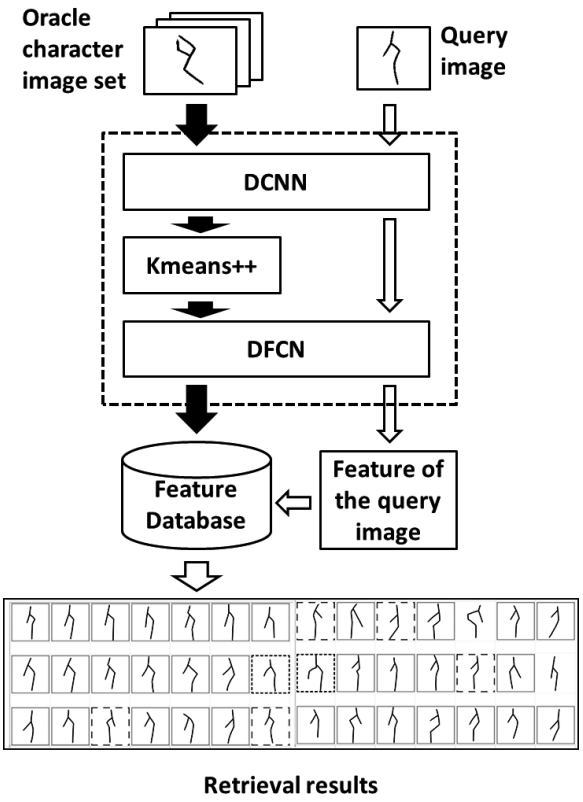

Fig. 2. Framework of the proposed image retrieval algorithm.

During the retrieval stage (modules connected by black hollow arrows in the figure), the clustering algorithm K-means++ does not work any more. The query image is firstly input into DCNN to acquire its character-level features, and then these features are directly input into DFCN to obtain the variant-level features, which are employed as the final description of the query image and used to search in the database of image features.

III. The proposed algorithm

In this section, the main modules of the proposed algorithm are introduced in details.

A. Character-level features obtained by DCNN

Our oracle character image set consists of 41957 gray images with the same size of 64×64, involving 5491 different oracle characters. All of these images are obtained by historians in a way of carefully converting scanned oracle bone images with pen-based input devices. Therefore, all images have very clear strokes and without any background disturbance.

As shown in Fig. 3, we design a seven-layer (only accounting for the convolutional layers and the fully connected layers) DCNN model for character-level feature extraction. The network consists of five convolutional layers and two fully connected layers. Each of the first three convolutional layers are followed by a max-pooling layer, followed by tow consecutive convolutional layers, and then another max-pooling layer. After the last max-pooling layer, there is a fully connected layer with 1024 neurons, and then is another

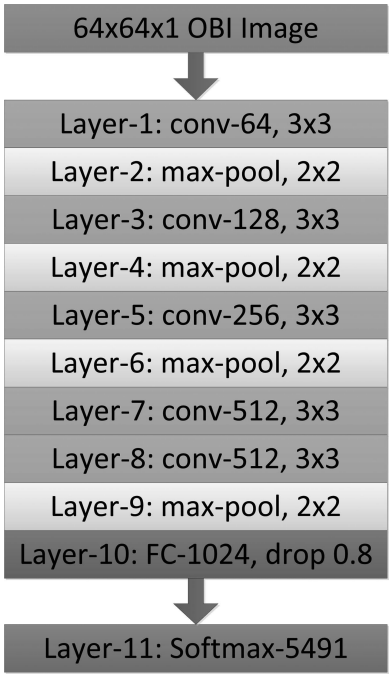

Fig. 3. Architecture of the DCNN model for extracting character-lever features.

fully connected layer with 5491 neurons, which is used to perform character classification.

In order to make the network converge easily and minimize the risk of overfitting, the rectified linear unit (ReLU) is used for all convolutional layers and the first fully connected layer (FC1). For the convolutional layers, all filters are 3×3, and the stride of the sliding window is 1. The max-pooling operation is carried out over a 3×3 window with the stride of 2. The dropout rate 0.8 is adopted for FC1 to increase the generalization ability of the model. In this study, the outputs of FC1 are employed as the character-level image features.

B. Kmeans++ clustering for dividing character-level features

As shown in Fig. 1, character patterns vary greatly between different variants of the same oracle character. Therefore, character-level features obtained by DCNN are not accurate enough to describe the patterns of oracle character variants. In other words, we have to find a new way to reduce the intra-class differences to improve the accuracy of oracle character descriptions. An easy way to solve such a problem is resorting to the clustering analysis, which divides data set into smaller subset, each of which consists of images only belonging to one kind of variants. Among all clustering algorithms, KMeans++[31] is one of the most frequently used techniques. It is an unsupervised algorithm that divides similar data into the same cluster and dissimilar ones into different clusters. In this way, the intra-class differences can be greatly reduced.

Kmeans++ improves Kmeans in the determination of initial cluster centers. It iteratively selects samples that are farther away from existing cluster centers as a new cluster center with a higher probability, resulting into more decentralized clustering centers. In each iterative step, the probability of a sample being selected as a new center is defined as

$$p(x) = \frac{D(x)^2}{\sum_{i=1}^{n} D(x_i)^2}, \tag{1}$$

where $D(x)$ refers to the shortest distance from a sample x to all existing centers and n is the number of samples.

In this study, the cluster number K indicates the variant number of a certain oracle character. In order to get the the optimal K, only characters consisting of more than 10 samples are grouped by Kmeans++. Specifically, we extract the character-level features by DCNN described above for all images belonging to the same character, and then iteratively set K as an integral value in $[2, \sqrt{n}]$ and calculate the clustering results. The value with the optimal Silhouette coefficient (SC) [32] is determined as the optimal cluster number K, and the corresponding clustering result is employed as the division of the set of images. The SC can be calculated as below:

$$SC_K = \frac{1}{n} \sum_{i=1}^{n} S_i, \tag{2}$$

where SC_K is the SC with K clusters, and S_i is the SC of the i^{th} sample:

$$S_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}, \tag{3}$$

where a_i is the average distance of sample i to others in the same cluster, and b_i is the minimum value of average distances to different clusters. A higher SC_K indicates a better clustering result.

By the clustering analysis of Kmeans++, images of an oracle character will be divided into several subsets, each of which corresponds to a variant of this character. An example is shown in Fig.4. Different linetypes indicate different clusters, representing different variants. After such a processing for each oracle character, we get a new variant-level dataset with 7997 different classes.

C. Variant-level features extracted by DFCN

In order to get variant-level image features, in this study, we design a four-layer DFCN (as shown in Fig.5) based on the variant-level dataset obtained by Kmeans++. The first three fully connected layers have 2014, 1024, and 2048 neurons, respectively. The ReLU function is used as the activation function for these layers. Each of the first three fully connected layers is followed by a dropout layer with a rate of 0.5 to reduce the computation and improve the generalization of DFCN. The last fully connected layer contains 7997 neurons, and it is used to classify different variants. In this study, the outputs of the first fully connected layer (FC1) of DFCN are exploited as the the variant-level image features.

In order to eliminate the magnitude difference between different levels of data, the L2 regularization is applied to all fully connected layers and convolutional layers of both DCNN and DFCN.

D. Feature matching

In CBIR systems, many methods are used to measure the distance between features, e.g., the Euclidean distance, the Cityblock distance, and the Chebychev distance. Because we

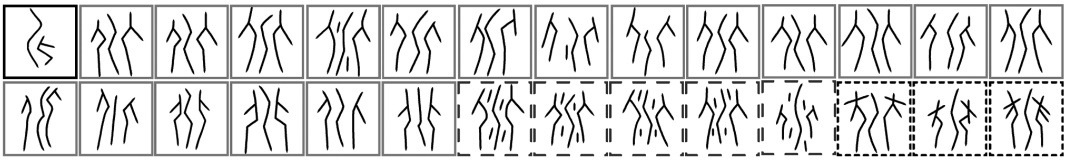

Fig. 4. The clustering result of the oracle character with Id 999082. Different linetype boxes indicate different clusters.

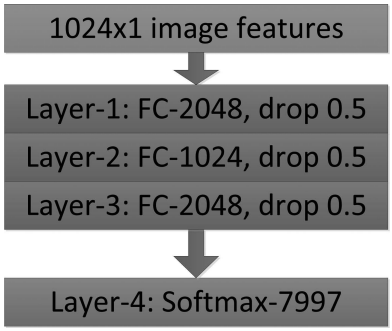

Fig. 5. Architecture of the DFCN model for extracting variant-level features.

do not focus on finding the optimal distance function, in this study, the simplest method Euclidean distance is employed. Let $F_i = [F_{i,1}, \ldots, F_{i,m}]$ be the feature vector of the query image and $F_j = [F_{j,1}, \ldots, F_{j,m}]$ be the feature vector of an image in the dataset, the Euclidean distance can be calculated by

$$d(F_i, F_j) = \sqrt{[\sum_{h=1}^{m}(F_{i,h} - F_{j,h})^2]}, \tag{4}$$

where m is the length of feature vector. In this study, because the output of FC1 layer of DFCN is employed as the variant-level feature vector, we set $m = 2048$.

E. The proposed image retrieval algorithm

The proposed algorithm includes two parts: training part and query part.

In order to prevent from overfitting as well as improve the robustness, random cropping, random rotation, and random scaling are applied to model training. Let $I = \{I_1, \cdots, I_n\}$ be the image set and $C = \{C_1, \cdots, C_n\}$ be the corresponding set of oracle character label, where n is the sample number and $C_i \in \{1, 2, \cdots, 5491\}$ indicates the oracle character label of I_i .The training algorithm can be described as algorithm 1.

After gaining the image feature dataset F, the image retrieval algorithm can be described as algorithm 2.

Algorithm 1 Training DCNN and DFCN.

Input: The image set I, and the oracle character label set C

Output: Variant-level feature dataset F

1: Training the DCNN model. The cross-entropy loss function is employed, and the adaptive moment estimation algorithm is used to learn the parameters of the model. Set the learning rate, the number of epoches, and the batch size, respectively. After the convergence, save the output of FC1, which is the character-level description denoted as $f = \{f_1, f_2, \cdots, f_n\}$.
2: **for** $k = 1 \rightarrow 5491$ **do**
3: Execute Kmeans++ on image features $f^k = \{f_i^k | f_i^k \in f, C_i^k = k\}$ that all features have the same oracle character label k.
4: For each image feature, save its cluster Id.
5: **end for**
6: Training the DFCN model. Based on the feature set f and the cluster Id set $L = \{l_1, l_2, \cdots, l_n\}$ (where l_i is the cluster Id of f_i obtained by Kmeans++), using the same loss function and optimizing algorithm as used for DCNN, train the DFCN model. Set the learning rate, the number of epoches, and the batch size, respectively. After the convergence, save the outputs of FC1, denoted as $F = \{F_1, F_2, \cdots, F_n\}$, which forms the feature database used for image retrieval and also provides the final output of training.

Algorithm 2 Image querying from the database

Input: The query image I and the minimum number r of retrieved images.

Output: The set of retrieved images $Q = \{q_1, q_2, \cdots, q_r\}$.

1: Enter image I into DCNN, calculate the character-level image feature f_I.
2: Enter f_I into DFCN, calculate the variant-level image feature F_I.
3: For each image feature F_i in the dataset F, calculate the distance $d(F_i, F_I)$.
4: Find the smallest r distances, and get the corresponding image set $Q = \{q_1, q_2, \cdots, q_r\}$.

IV. Experimental results

A. Oracle character image set

The dataset used in this study is a set of oracle character images, some examples are shown in Fig.6. Each row shows ten samples of the same oracle character coded by the oracle character Id. For instance, the first row shows images with oracle character Id "00100", which is the code for the oracle character "person". The dataset consists of 41957 images, involving 5491 different oracle characters. All images are created by experts in a way of tracing inscriptions in the scanned oracle-bone images by pen-based input devices. Therefore, the resulting character images are very clear and without any background disturbance. As mentioned earlier, each oracle character may have many different variants, therefore, images of different oracle characters are grouped by Kmeans++ respectively, resulting in a new division of the dataset that has 7997 different variants of oracle characters. Some techniques, such as random cropping, random rotation, and random scaling, are employed to augment the dataset.

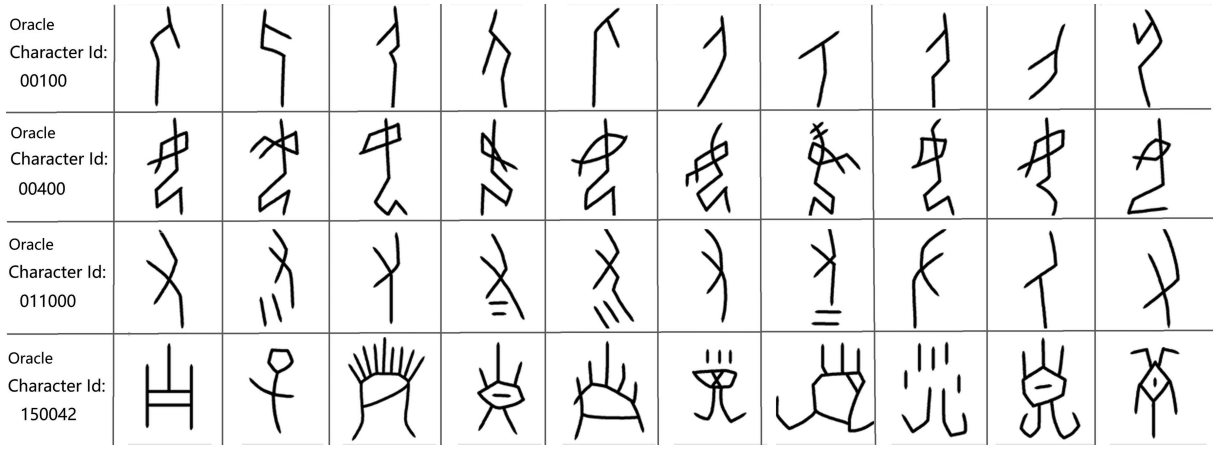

Fig. 6. Some examples of our oracle character image set.

B. Experimental analysis

Both precision and recall are used to evaluate the performance of the proposed method. They are defined as follows.

$$P = \frac{A}{C},\ R = \frac{A}{C}, \tag{5}$$

where P refers to "precision" and R "recall". A is the number of positive samples in a retrieval result, B is the number of retrieved samples, and C is the number of relevant samples in the dataset.

Ten feature extraction methods are compared in this study, including five traditional ones (SIFT[1], SURF[2], GIST[3], LBP[4], HOG[5]) and five deep-learning based ones:

- VGG16-FC6: the output of FC6 of VGG16 [14].
- GPW: a General Pair Weighting (GPW) framework, which unifies existing pair-based metric learning approaches into general pair weighting through gradient analysis for the first time [20].
- DCNN-FC1: the output of FC1 of the DCNN model in this study.
- DFCN-FC1: the output of FC1 of the DFCN model in this study.
- DenseNet: the output of the last global average pooling layer [33].

For the sake of fairness, the Euclidean distance function (Eq. 4) is used to measure the similarity between image features extracted by all methods mentioned above. The precision-recall curves are shown in Fig. 7. First of all, it is easy to be found that the deep-learning based methods (VGG16-FC6, GPW, DenseNet, DCNN-FC1, and DFCN-FC1) have better results than the traditional ones (SIFT, SURF, GIST, LBP, and HOG). It implies that image features extracted by deep neural networks have more powerful abilities to describe images from multiple scales and contain more semantic information to represent the patterns of oracle characters. Secondly, it is very clear that DFCN-FC1 is superior to other deep learning based methods, which verifies the basic idea of this study. By means of clustering analysis, more variant-level semantic information can be discovered, which undoubtedly boosts the retrieval performance. Besides, it is obvious that VGG16-FC6 and DenseNet perform worse than DCNN-FC1, which mainly results from the simpler background of oracle character images. The key point of this study is not to find more abstract global information, but to reduce the serious interference between different variants. From another point of view, such a result further verifies the rationality of the proposed algorithm .

TABLE I
COMPARISON OF QUERY TIME UNDER DIFFERENT METHODS (IN SECONDS)

Algorithm	Average query time of 1000 images	Feature size
DFCN-FC1	0.5975	2048
DCNN-FC1	0.5349	1024
VGG-FC6[14]	0.5920	64
GIST[3]	0.4701	512
HoG[5]	0.6171	2000
LBP[4]	1.0468	3236
SIFT[1]	0.9673	500
SURF[2]	3.3465	1000
GPW[20]	0.6890	512
DenseNewt[33]	0.6396	1920

In order to make a more intuitive comparison between DFCN-FC1 and DCNN-FC1, we show the first 30 retrieval results of two oracle characters in Fig. 8, where each gray solid rectangle indicates a positive sample. It is clear that DFCN-FC1 performs better than DCNN-FC1. Specifically, DFCN-FC1 has the ability to obtain more positive samples within the first several retrieved images, while DCNN-FC1 performs poorer. It is mainly because the combination of deep learning and clustering analysis is capable of providing more accurate features for oracle character images.

In order to observe the performance of the algorithm in practical applications, we collect and compare the average query time of different algorithms over 1000 query images, as shown in Table I. All algorithms were implemented in the environment of Python and performed on a Windows 10 PC with an Intel(R) i7 3.4G-GHz CPU using 16G RAM. The average query time of our proposed method is 0.5975 seconds, which is faster than all referenced methods except GIST (0.4701 seconds). Such a result indicates that our method has an acceptable retrieval speed.

For the sake of getting a more comprehensive understanding of the proposed method, Fig. 9 shows some results of DFCN-FC1 that have low recall rates. In the figure, "Num" is the number of samples of certain oracle character. Only the first 30 retrieved images are showed in the figure. The symbol "6/30" means there are six positive samples in the first 30 retrieved images. From the figure, one can see

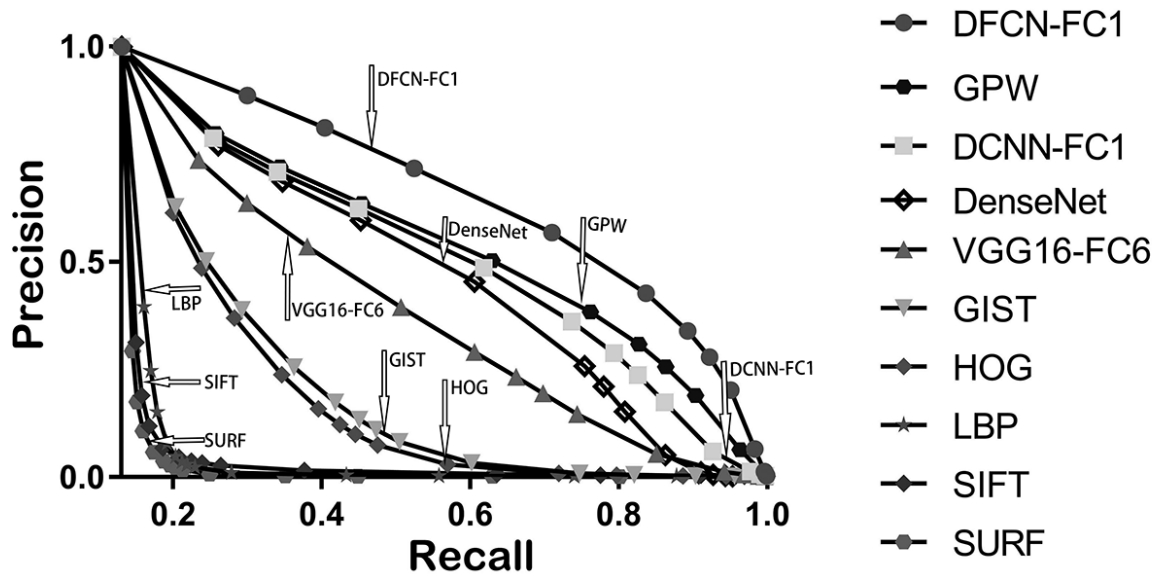

Fig. 7. Comparison between different algorithms.

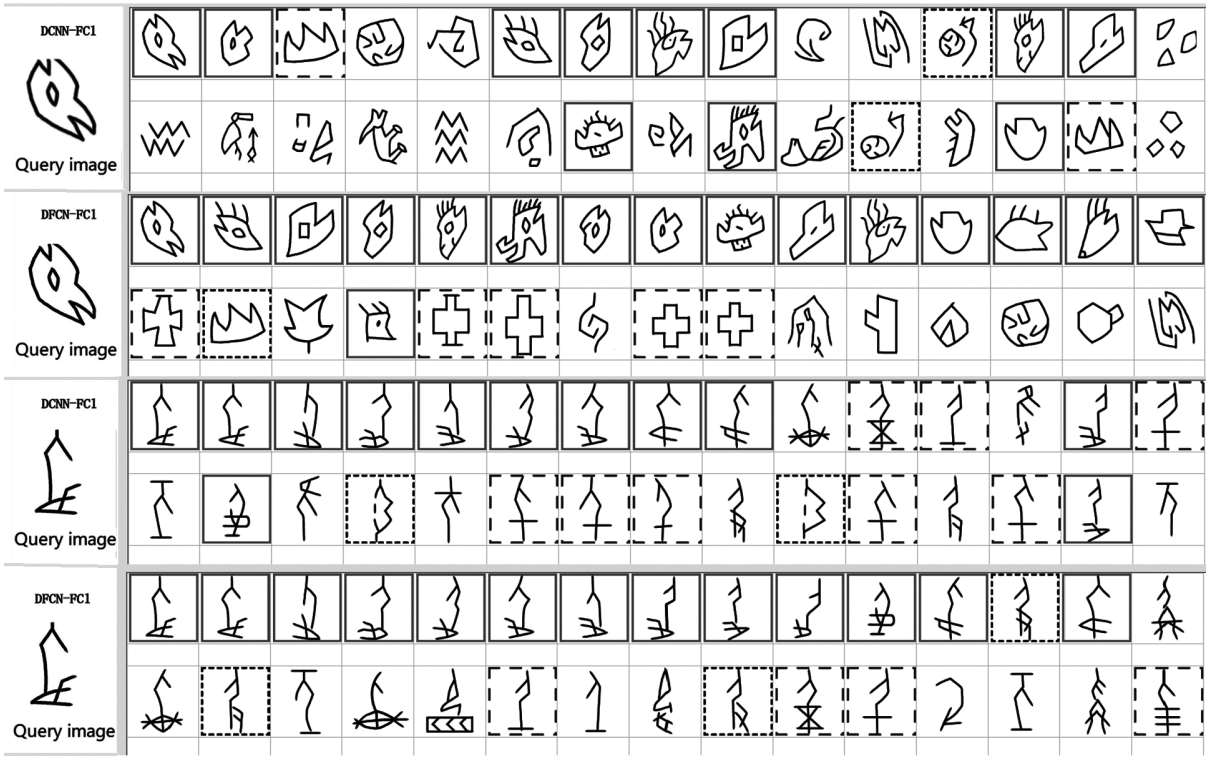

Fig. 8. Comparison between DFCN-FC1 and DCNN-FC1. Gray solid rectangles indicate positive results, black short dashed ones and black long dashed ones indicate negative oracle characters in large quantities, and oracle character images without any rectangles are negative results with only few samples.

that there exists some retrieved images belonging to other oracle characters. The main reason is that different oracle characters may have similar graphic patterns, which have a great negative impact on the recall rate. Such a phenomenon implies that the proposed algorithm still needs to be improved by addressing inter-class similarity.

V. Conclusions

In this study, the retrieval of oracle character images is studied by combining deep neural network and clustering technology. Based on the basic idea of reducing inter-class differences, the clustering technique is embedded between D-CNN and DFCN to get the variant-level division of character-level image features. By using an oracle character image set with 41957 samples involving 5491 different oracle characters, experimental results have clearly verified the superiority of the proposed method to the referenced ones. However, we find that the inter-class similarity has a great negative impact on the recall rate of image retrieval. Therefore, in the future work, we will further address the problems of both inter-

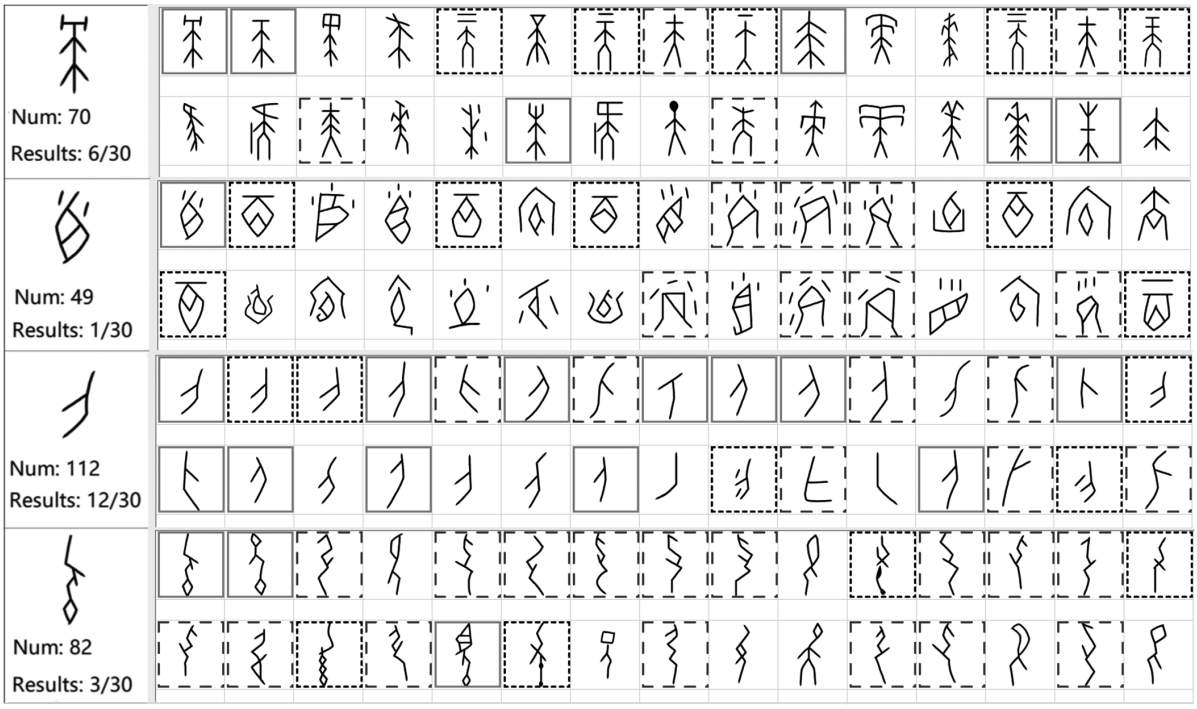

Fig. 9. Some samples provided by DFCN-FC1 that have low recall rates. Gray solid rectangles indicate positive results, black short dashed ones and black long dashed ones indicate negative oracle characters in large quantities, and oracle character images without any rectangles are negative results with only few samples.

class similarity and intra-class differences simultaneously, in order to design a more accurate retrieval algorithm.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94

[2] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314207001555

[3] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, pp. 145–175, 05 2001.

[4] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," pp. 971–987, July 2002.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, San Diego, CA, USA, June 2005, pp. 886–893 vol. 1.

[6] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, Prague 1, Czech Republic, 2004, pp. 1–22.

[7] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek, "Image classification with the fisher vector: Theory and practice," *International Journal of Computer Vision*, vol. 105, no. 3, pp. 222–245, Dec 2013. [Online]. Available: https://doi.org/10.1007/s11263-013-0636-x

[8] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 2, New York, NY, USA, June 2006, pp. 2169–2178.

[9] H. Jgou, M. Douze, C. Schmid, and P. Prez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 3304–3311.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012*, Lake Tahoe, Nevada, United States, 2012, pp. 1106–1114. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR 2015*, San Diego, CA, USA, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, USA, 2016, pp. 2818–2826. [Online]. Available: https://doi.org/10.1109/CVPR.2016.308

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, USA, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[14] J. Wan, D. Wang, S. C. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li, "Deep learning for content-based image retrieval: A comprehensive study," in *Proceedings of the ACM International Conference on Multimedia, MM '14*.

[15] J. Y. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, MA, USA, June 2015, pp. 53–61.

[16] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," in *Computer Vision - ECCV 2014 - 13th European Conference, Part VII*, Zurich, Switzerland, 2014, pp. 392–407. [Online]. Available: https://doi.org/10.1007/978-3-319-10584-0_26

[17] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, Las Vegas, NV, USA, 2016, pp. 1096–1104. [Online]. Available: https://doi.org/10.1109/CVPR.2016.124

[18] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Computer Vision - ECCV 2016 - 14th European Conference, Part VI*, Amsterdam, The Netherlands, 2016, pp. 241–257. [Online]. Available: https://doi.org/10.1007/978-3-319-46466-4_15

[19] Q. Zheng, X. Tian, M. Yang, and H. Wang, "Differential learning: A powerful tool for interactive content-based image retrieval," *Engineering Letters*, vol. 27, no. 1, pp. 202–215, 2019.

[20] X. Wang, X. Han, W. Huang, D. Dong, and M. R. Scott, "Multi-similarity loss with general pair weighting for deep metric learning,"

in *2019 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'2019)*, Long Beach, CA, USA, 2019, pp. 5022–5030. [Online]. Available: http://arxiv.org/abs/1904.06627

[21] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, no. 1, pp. 81–87, Jan 1984.

[22] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," *Journal of Cybernetics*, vol. 3, no. 3, pp. 32–57, 1973.

[23] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Norwell, MA, USA: Kluwer Academic Publishers, 1981.

[24] N. Nguyen, Q. M. J. Wu, and S. Ahuja, "An extension of the standard mixture model for image segmentation," *IEEE Transactions on Neural Networks*, vol. 21, no. 8, pp. 1326–1338, Aug 2010.

[25] D. Comaniciu and P. Meer, "Mean shift analysis and applications," in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 2, Sep. 1999, pp. 1197–1203 vol.2.

[26] M. Fashing and C. Tomasi, "Mean shift is a bound optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 471–474, March 2005.

[27] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, Montreal, Quebec, Canada, 1996, pp. 103–114. [Online]. Available: https://doi.org/10.1145/233269.233324

[28] G. Liu, Y. Zhang, and A. Wang, "Incorporating adaptive local information into fuzzy clustering for image segmentation," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3990–4000, Nov 2015.

[29] C. H. You, K. A. Lee, and H. Li, "Gmm-svm kernel with a bhattacharyya-based distance for speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1300–1312, Aug 2010.

[30] L. Yu, "Moving target tracking based on improved meanshift and kalman filter algorithm," in *2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Wuhan, China, May 2018, pp. 2486–2490.

[31] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, New Orleans, Louisiana, USA, 2007, pp. 1027–1035. [Online]. Available: http://dl.acm.org/citation.cfm?id=1283383.1283494

[32] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.

[33] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, 2017, pp. 2261–2269. [Online]. Available: https://doi.org/10.1109/CVPR.2017.243

LIU Guoying was born in ZhengZhou city, China, on 16 July, 1979. He received his master degree in computer science from Changsha University of Science and Technology, Changsha, China, in 2004, and received his Ph.D. degree in geoscience and remote sensing from Wuhan University, China, in 2009. His research interests mainly focus on the pattern recognition and machine learning, in particular, text image detection and recognition.

He worked as a lecturer in computer science with Changsha University of Science and Technology from 2004 to 2009. After then he became a teacher with Anyang Normal University. From 2014 to 2015, he went to the University of New Brunswick, Fredericton, Canada, to purse his Post-Doctoral position. He is currently a professor in computer science with Anyang Normal University, China.

WANG Yangguang was born in Shangqiu city, China, on 29 April, 1998. He is currently an undergraduate student pursuing his bachelor degree in computer science. He was funded by the science and technology innovation foundation for the university or college students of China under Grant S201810479003.