

## Research Article

# Research on Fast Face Retrieval Optimization Algorithm Based on Fuzzy Clustering

Xiajun Dong , Bin Huang , and Yuncai Zhou 

College of Computer Science, Yangtze University, Jingzhou 434023, Hubei, China

Correspondence should be addressed to Bin Huang; jonsen@yangtzeu.edu.cn

Received 11 November 2021; Accepted 8 December 2021; Published 7 January 2022

Academic Editor: Punit Gupta

Copyright © 2022 Xiajun Dong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Aiming at the problem of long retrieval time for massive face image databases under a given threshold, a fast retrieval algorithm for massive face images based on fuzzy clustering is proposed. The algorithm builds a deep convolutional neural network model. The model can be used to extract features from face photos to obtain a high-dimensional vector to represent the high-level semantic features of face photos. On this basis, the fuzzy clustering algorithm is used to perform fuzzy clustering on the feature vectors of the face database to construct a retrieval pedigree map. When the threshold is passed in for database retrieval of the target face photos, the pedigree map can be quickly retrieved. Experiments on the LFW face dataset and self-collected face dataset show that the model is better than the commonly used K-means model in face recognition accuracy, clustering effect, and retrieval speed and has certain commercial value.

## 1. Introduction

In recent years, due to the rapid development of image recognition technology, face recognition technology is also developing rapidly [1]. The face recognition [2, 3] model based on deep learning keeps refreshing records on the open-source dataset, such as LFW [4], MegaFace [5], and other datasets. Many cameras have implemented deep learning algorithms in the camera in order to realize the real-time face recognition function just using the camera. This type of face recognition camera can easily complete the frame-by-frame analysis of the captured video and extract the face from the video. The face captured by the camera is transmitted to the server for storage. In the field of security [6], the facial recognition cameras deployed by county-level units upload 5 million facial photos to the server every day. Each face photo uploaded to the server needs to be compared with a locally constructed blacklisted face database in real time to determine whether the person is a person on the blacklist controlled by the police.

The face comparison process of the large dataset contained in the above scenario has great social value in the security field. Face comparison under the large face dataset faces the following challenges:

- (1) The face database has a huge number of faces. In most scenarios, its quantity can reach tens of millions or even hundreds of millions.
- (2) There are a lot of face photos collected by the same person, such as person's ID photos, front-grabbing photos, side-grabbing photos, and so on. This will double the size of the face database and make the comparison more difficult.
- (3) The number of samples of each type of face is not balanced, resulting in different comparison times in the same class. Due to the short board effect, the final comparison speed is dragged down by the samples with a large number of samples.

In 2018, Li et al. used the deep feature clustering algorithm to optimize the problem of massive face retrieval [7] and achieved good results because they used the K-means clustering algorithm to cluster the face features. Unbalanced distribution of features will lead to serious missed detection.

In 2019, Dubey improved the discrimination of face image descriptors by using the decoder concept of multi-channel decoded local binary pattern over the multifrequency patterns. In this paper, a frequency decoded local

binary pattern (FDLBP) is proposed with two decoders. This can greatly improve the accuracy of face retrieval, but when the number of faces is huge and the number grows dynamically, it cannot meet the requirements of high real time [8].

This paper is based on the deep learning model for feature extraction of face photos, combined with feature fuzzy clustering algorithm for fuzzy clustering of face feature vectors. Using Caltech 10k Web Faces Dataset as the training set, clustering of face photos on the LFW face dataset and the face dataset collected from the Internet has achieved good results.

## 2. Design of Face Image Retrieval Model Based on Fuzzy Clustering

The model structure of the facial image retrieval algorithm based on fuzzy clustering is shown in Figure 1. The face image retrieval model mainly includes two parts. The first part uses the deep convolutional neural network model [9, 10] to extract features from the face image [2, 3] to obtain a 256-dimensional face feature vector  $v_i\{v_1, v_2, \dots, v_n\}$ . Each component in the feature vector ( $v_i$ ) is a floating-point type data, and it keeps 4 decimal places. The formula for calculating the similarity of the feature vectors of two people's faces is defined as shown in formula (1). If the value of  $\text{Similar}_{ij}$  tends to 1, then the face photos represented by feature vectors  $v_i$  and  $v_j$  are more similar.

$$\begin{aligned} \text{Similar}_{ij} &= \sum v_i \times v_j, \\ \text{Similar}_{ij} &= 1, \text{ IF}(\text{Similar}_{ij} > 1), \\ \text{Similar}_{ij} &= 0, \text{ IF}(\text{Similar}_{ij} < 0). \end{aligned} \quad (1)$$

Algorithms commonly used to calculate image similarity, such as Euclidean distance, cosine distance, hamming distance, and so on, have their own advantages and disadvantages. In formula (1), we use an algorithm that approximates the Euclidean distance to calculate the similarity of two face images. This algorithm is closer to the calculation method of matrix multiplication and can perform  $N:N$  face similarity calculations at the same time. GPU is best at matrix operations, so this algorithm can maximize the performance of GPU.

The second part is to perform fuzzy clustering on the eigenvectors in the face database to be compared to obtain the similarity matrix of the pairwise eigenvectors  $S_{ij}$  and to draw a pedigree map  $P$  for the cluster  $\{C_1, C_2, \dots, C_j\}$ . The next step is to calculate the center point of each cluster in the pedigree graph to get the cluster center point  $\{\mu_1, \mu_2, \dots, \mu_n\}$   $\max(n) = 256$ . After extracting the features of the face image to be searched, perform further retrieval according to the cluster center points in the pedigree diagram.

**2.1. Face Feature Extraction-Deep Convolutional Neural Network Structure Design.** The deep learning model used in this article is mainly based on the convolutional neural

network(CNN), and the structure of the convolutional neural network is fine-tuned to satisfy the effective feature extraction of face photos. Considering that the feature information of face photos will lose part of the information during the global pooling operation, resulting in incomplete expression of face features, the maximum pooling method is used in the structural design. The input of the deep convolutional neural network is a face picture, and the output is a one-dimensional vector of 256 dimensions. This vector is used to represent the characteristics of the face, which we call the feature vector of the face.

The structure design of the face feature extraction model based on the deep convolutional neural network is shown in Figure 2. The network is called FFEDN (Face-Feature Extract Dense Net) in this article, and the Stack Conv\* structure in the model is the structure shown in Figure 3.

**2.2. Design of Face Image Retrieval Process.** The design of the face image retrieval process is shown in Figure 4. The face photos to be searched are processed into pictures with  $W \times H \times C$  pixels in the system. We use  $128 \times 128 \times 3$  pixels in this model. Then, we can get a 256-dimensional vector through the FFEDN model. The next step is to calculate the similarity between this vector and the cluster center point vector. After this step, we can get the most similar clusters. Then, search within this cluster until you get the most similar leaf node. Finally, return all face photos in this cluster in descending order of similarity.

In the public security application scenario, when performing face retrieval, the 100 most similar face images are often screened out, namely, top100. Therefore, fuzzy clustering can avoid the problem of missing the most similar face due to inaccurate determination of K value when using K-means clustering [10].

## 3. Face Image Retrieval

**3.1. Feature Vector Extraction Method for Face Photos.** The face photos to be searched need to be processed into a three-dimensional tensor structure  $F(I): W_I \times H_I \times C_I$ . In this structure,  $C_i$  represents the number of channels of the color picture, and this article takes the three channels of the picture, namely,  $C_i = 3$ ,  $W$  means the picture width,  $H$  means the picture height, and the model will process the picture into a dimension of  $128 \times 128$ , that is,  $W = 128; H = 128$ . That is, the face image needs to be processed as a tensor with  $128 * 128$  pixels and a channel of 3.

The calculation process of tensor in the deep neural network structure shown in Figure 2, and the calculation of the convolutional layer is defined as follows:

$$X_{W,b}^{m(l)} = f \left( b^{m(l)} + \sum_n W^{mn(l)} * X_{W,b}^{m(l-1)} \right), \quad (2)$$

where  $X^{m(l)}$  represents the  $m$ th channel of layer  $l$ ,  $W^{mn(l)}$  and  $b^{m(l)}$  are the corresponding convolution kernel filters and bias terms, and  $*$  is the convolution operator.

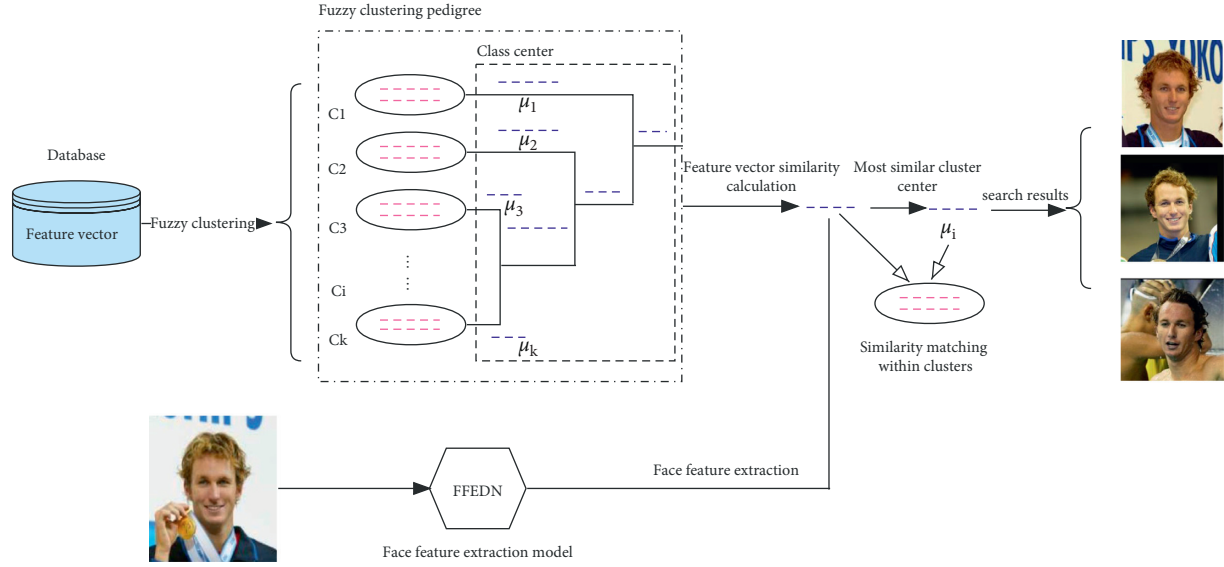


FIGURE 1: Face image retrieval model structure.

In order to avoid inaccurate feature expression caused by overfitting, this model structure uses a local maximum pooling (max-pooling) method to process the output of the convolutional layer.

$$X_{(i,j)}^{m(l)} = \max_{\forall (p,q) \in \Omega(i,j)} \{X_{(p,q)}^{m(l)}\}, \quad (3)$$

where  $\Omega(i,j)$  is the specific area of index  $(i,j)$  and  $(p,q)$  is the index of the specific position on  $\Omega$ . The last layer of the network is a fully connected layer, and finally a 256-dimensional 1D vector is obtained to represent the characteristics of the face.

**3.2. Fuzzy Clustering of Facial Feature Vector.** According to formula (1), calculate the correlation coefficient matrix ( $r_{ij}^*$ ) between the feature vectors of two people's faces.

Use formula (4) to perform matrix transformation on the correlation coefficient matrix, so that the value in  $r_{ij}^*$  is converted to the interval  $[0,1]$ , thereby forming a fuzzy matrix  $R = (r_{ij})$ .

$$r_{ij} = \frac{1 + r_{ij}^*}{2}. \quad (4)$$

Perform a convolution operation on the fuzzy matrix:  $R \rightarrow R^2 \rightarrow R^3 \rightarrow \dots \rightarrow R^n$ . After a finite number of convolution operations, make  $R^n \bullet R = R^n$ , where " $\bullet$ " represents the convolution operator. When  $R^n \bullet R = R^n$ ,  $R$  is the final correlation coefficient matrix. Sort the correlation coefficient values  $\lambda_{ij}$  in the correlation coefficient matrix in descending order, intercept the values  $\lambda_{ij} > 0.75$  to construct the pedigree diagram, and finally obtain the pedigree diagram. The threshold 0.75 is a commonly used value for face recognition, that is, when  $\text{Similar}_{ij} > 0.75$ , it can be considered that the two persons are the same person or people with high similarity.

**3.3. Building a Pedigree Map and Searching within Clusters in the Pedigree Map.** In Section 3.2, after the fuzzy matrix is calculated, the fuzzy matrix is sorted in reverse order

according to the value  $\lambda_{ij}$  of the fuzzy matrix, and the same values are classified into the same cluster according to the result of the sorting, thereby constructing a clustering pedigree diagram. Next, calculate the center point of the eigenvectors of each pedigree graph, and the calculation formula (5) is as follows:

$$\text{circle} = \frac{\sum_{i=1}^n r(i)}{256} \quad n = 1, 2, \dots, 256. \quad (5)$$

In the face retrieval process, set the given similarity threshold to threshold = 0.75. First calculate the similarity with the nodes in the pedigree graph, and the similarity calculation formula is shown in formula (1). Select nodes with similarity  $> 0.75$  to perform intranode branch search, and the judgment condition is still whether the similarity is greater than the threshold. Finally, the same similarity calculation method in the cluster can be used to calculate the face similarity, and the final results are returned in descending order.

## 4. Experimental Process and Result Analysis

The CPU of the server used in the experiment is Intel(R) Core(TM) Xeon-5600 series. The memory of the server is 128 G. The GPU uses NVIDIA Ti-Tan. The operating system uses Centos 7.4. This experiment is based on the TensorFlow framework.

**4.1. Face Recognition Dataset.** In this paper, the Caltech Web Faces Dataset is used to train the model, and the LFW dataset and some datasets collected from the Internet are used as the model's test dataset. The total dataset contains 1000 samples and about 600,000 face photos. Some of the datasets are shown in Figure 5.

**4.1.1. Training Dataset.** The training set used in this article is Caltech 10 k Web Faces Dataset, which was released in 2007

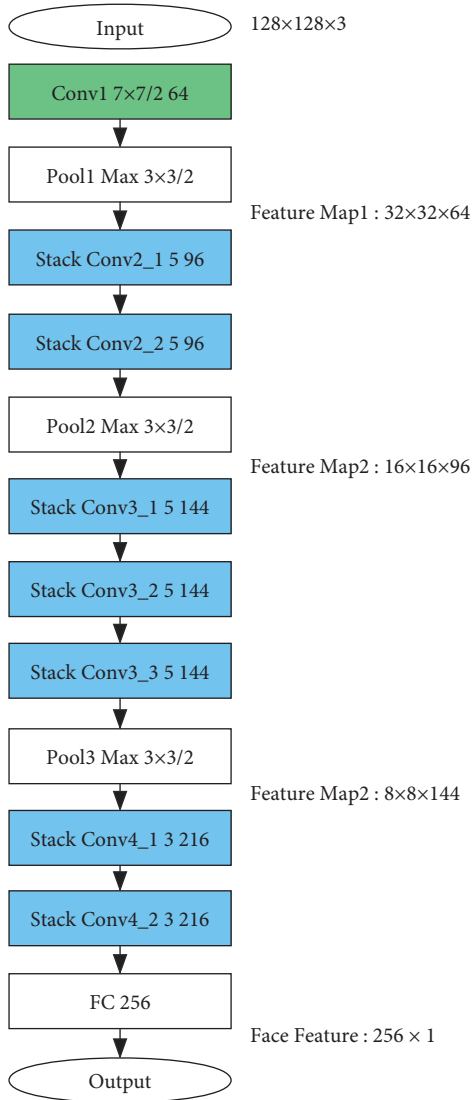


FIGURE 2: Deep convolution facial feature extraction neural network structure diagram.

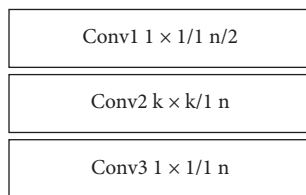


FIGURE 3: Model convolutional layer structure diagram.

and is obtained by crawling Google search engine with keywords. This dataset provides the center coordinates of the front eyes, nose, and mouth in each picture. The dataset has 10,524 pictures with different resolutions. Before the formal model training, we perform basic processing on the picture: face alignment, face cropping, and face cropping. The size after face cropping is  $128 \times 128$  pixels.

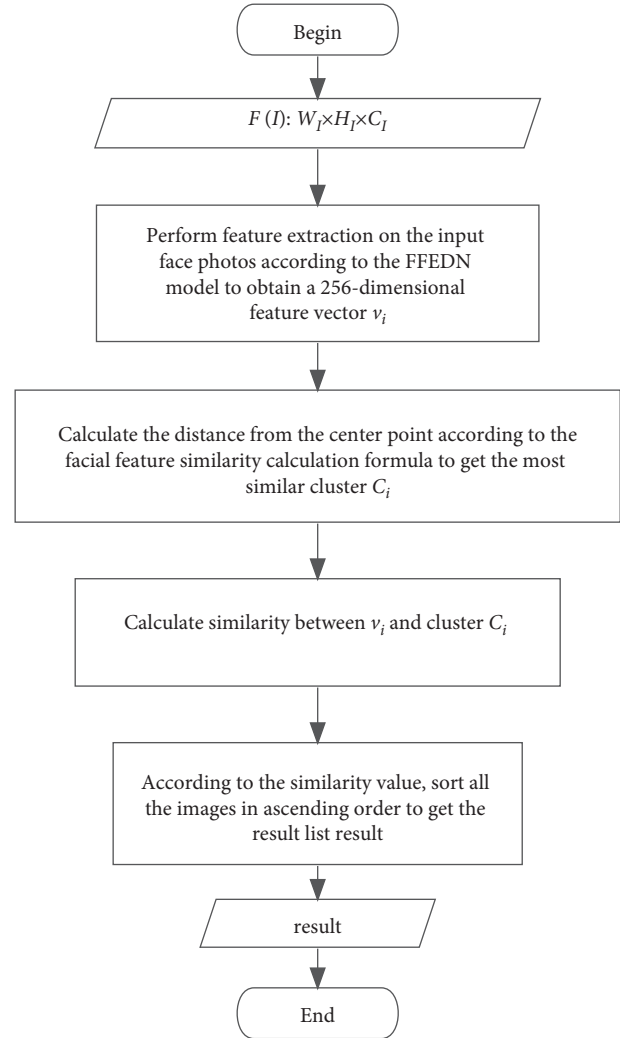


FIGURE 4: Face image retrieval process design.

**4.1.2. Test Dataset.** The test set LFW (Labeled Faces in the Wild dataset) is used in this article. This dataset contains 13,233 face photos of 5749 different people. At the same time, a search engine was used to collect face image datasets of 100 celebrities, with about 200 photos of each celebrity, and a total of 16,846 face photos were collected. The image collection process is as follows:

- (1) Determine the list of names of people to be crawled.
- (2) Use the Java HTTP component framework to call the search engine to search for pictures based on the names of the characters and download the pictures and save them in the designated folder.
- (3) Invoke its own face comparison algorithm and cloud face recognition SDK [2] to compare the collected face photos to identify as the same person as possible, and at the same time, delete poor quality pictures, such as severely occluded and blurred pictures.
- (4) Rotate, align, and crop the collected pictures, and the final picture size will be  $128 \times 128$  pixels.

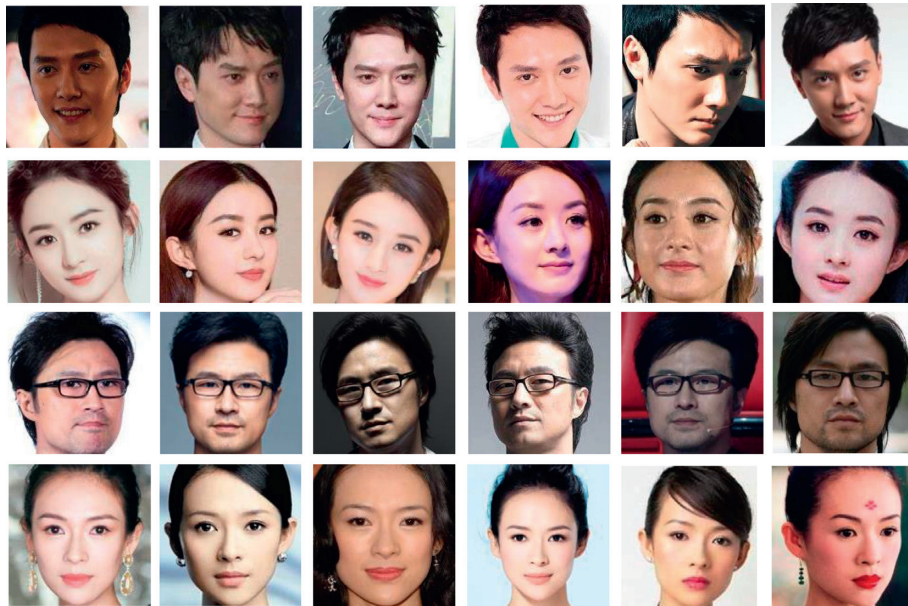




(1) Caltech Web Faces partial face data



(2) LFW partial face data



(3) Collect some facial data from the Internet

FIGURE 5: Training data used in the experiment (some face image data). (a) Caltech web faces partial face data. (b) LFW partial face data. (c) Some facial data from the Internet.

TABLE 1: Comparison of results between K-means clustering algorithm and fuzzy clustering algorithm based on LFW test dataset.

Algorithm/evaluation method	LFW			
	$F_1$ measurement	Cluster/node	LSR	Time/s
<b>K-means</b>	0.328	5749	0.232	367.86
<b>Based on fuzzy clustering</b>	0.824	5869	0.046	168.45

TABLE 2: Comparison of the results of K-means clustering algorithm and fuzzy clustering algorithm on the star test dataset.

Algorithm/evaluation method	Self-collected star dataset			
	$F_1$ measurement	Cluster/node	LSR	Time/s
<b>K-means</b>	0.828	100	0.02	321.16
<b>Based on fuzzy clustering</b>	0.964	100	0.00	134.28

In summary, the test dataset has two sets of data and 5949 face photos of 30079 different people, of which 4,069 people contain only one photo. Before the test, the face photos in all test groups were also aligned and cropped.

**4.2. Clustering Algorithm Evaluation Method.** In order to better evaluate the clustering ability and retrieval effect of the face clustering algorithm, this paper uses the  $F_1$  metric to evaluate the clustering results, as shown in the following formula:

$$F_1 = \frac{2 \times P_{\text{pair}} \times R_{\text{pair}}}{P_{\text{pair}} + R_{\text{pair}}}, \quad (6)$$

where  $P_{\text{pair}}$  is called the pairwise precision and  $R_{\text{pair}}$  is called pairwise recall (pairwise recall). When each sample is clustered into a single cluster, a high accuracy rate will be obtained, but the recall rate will be very low. On the contrary, when all samples are clustered into a cluster, a very low accuracy will be obtained. Therefore, the  $F_1$  metric combines the accuracy rate and the recall rate. Only when the accuracy rate and the recall rate are both high, the  $F_1$  metric can get a higher value.

In the actual application process of face retrieval, it is often necessary to return face pictures whose similarity is higher than a certain threshold in descending order of similarity for practical application. Under normal circumstances, the full database of faces is compared for similarity, and it is enough to return face photos greater than the threshold, but after clustering, due to different clustering effects, there will be no face photos with high similarity to the retrieval target. The clusters are not retrieved, so this article defines a new index to comprehensively evaluate the retrieval effect after clustering. The index is named LSR (lost search ratio), and the index formula is expressed as follows:

$$LSR = \frac{S_p}{R_p}, \quad (7)$$

where  $S_p$  is the number of retrieved faces whose similarity is greater than the threshold and  $R_p$  is the total number of faces in the library whose similarity is greater than the threshold. In this paper, the average value of the missed detection rate LSR obtained from 100 random searches is taken as the evaluation index.

**4.3. Analysis of Face Clustering Algorithm.** In this paper, the two algorithms based on K-means clustering and fuzzy clustering, which are widely used, are compared experimentally, and the clustering effect and retrieval effect are comprehensively evaluated. The results are shown in Table 1. On the LFW dataset, the LSR value based on fuzzy clustering is 0.046. It can be seen that compared to the K-means algorithm, fuzzy clustering can more accurately retrieve similar faces greater than the threshold. At the same time, it can be seen from the result that fuzzy clustering has faster clustering speed. Judging from the distribution of fuzzy cluster clusters/nodes, its clustering is also very close to the true cluster value.

Since the number of face photos of the same person in the LFW dataset is skewed, it can be seen from the previous description of the test set that there are 4,069 people with only one photo in the face database. Therefore, in order to make the test results more reliable, this article uses the collected star dataset to test the model again. The test results are shown in Table 2. Table 2 shows that the performance of the K-means algorithm on the star data set is better than that on the LFW data set. It can be seen from the results in Table 2 that if the data samples in the face database are balanced, the performance based on fuzzy clustering is still better than the performance of the algorithm based on K-means clustering.

Experiments show that, when performing face retrieval under a given threshold, the clustering algorithm based on fuzzy clustering is more suitable for scenes with large amounts of data, multiple categories, and skewed samples.

## 5. Conclusion

Aiming at the problem of clustering and fast retrieval of face data with large amount of data, multiple categories, and inclined samples, this paper proposes a face database clustering method based on fuzzy clustering and a face retrieval method under a given threshold. First, deep learning is used to extract the feature of the face photos in the face database to obtain the feature vector of the face photo, then fuzzy clustering is performed to obtain the pedigree map, and the center point feature vector of the node is calculated and stored in the pedigree map. When searching, the score can be compared according to the similarity of the

nodes in the pedigree graph, and only the threshold with the comparison score within the threshold range can be used, greatly reducing the number of comparisons.

This paper conducts experiments on LFW and self-collected star datasets. By analyzing the clustering effect, the LSR indicator of the missed detection rate, and the clustering time indicator, it shows that the fuzzy clustering algorithm can be used in large-scale datasets and face samples. In the case of fast clustering, given a specific threshold for face retrieval, there is a greater advantage.

## Data Availability

The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

## Conflicts of Interest

The authors declare that they have no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## References

- [1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems*, pp. 1097–1105, ACM Press, New York, 2012.
- [2] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proceedings of the British Machine Vision Conference*, September 2015.
- [3] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: a unified embedding for face recognition and clustering," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, June 2015.
- [4] G. B. Huang, M. Matter, and T. Berg, "Labeled faces in the wild: a database for studying face recognition in unconstrained environments," <https://vis-www.cs.umass.edu/lfw/lfw.pdf>.
- [5] I. Kemelmacher-Shlizerman, S. M. Seitz, D. Miller, and E. Brossard, *The Mega Face Benchmark: 1 Million Faces for Recognition at Scale*, CVPR, Seattle, Washington, 2016.
- [6] X. F. Li, "Development status and changes of the face recognition market," *China Security*, vol. Z1, pp. 39–42, 2020.
- [7] Z. Li, Y. Zhong, B. Zhang, and D. Cao, "Mass face image retrieval based on deep feature clustering," *Journal of Harbin Institute of Technology*, vol. 50, no. 11, pp. 101–109, 2018.
- [8] S. R. Dubey, "Face retrieval using frequency decoded local descriptor," *Multimedia Tools and Applications*, vol. 78, no. 12, pp. 16411–16431, 2019.
- [9] A. Krizhevsky, I. Sutskever, and G. Hintong, "ImageNet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, p. 1097, Curran Associates Inc., Long Beach, 2012.
- [10] Z. Simonyank, "Very deep convolutional networks for large-scale image recognition," ArXiv Preprint ArXiv: 1409.1556, 2014.