http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (6): 1143-1151, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.1143.1151

Accelerating Face Recognition for Large Data Applications in Visual Internet of Things

[1]Zhuoxuan Jiang, [1]Yaping Lin and [2]Stan Z. Li
[1]School of Information Science and Engineering, Hunan University, No. 252, Lushan South Road, Changsha, 410082, China
[2]Center for Biometrics and Security Research and National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, 95 Zhongguancun Donglu, Beijing 100080, China

Abstract: This study described a face recognition system for large scale data applications in the framework of Visual Internet of Things (VIoT). The main issue here was the speed in large face matching. In order to solve this problem, this study proposed a general method that could accelerate the computation in several magnitudes based on classical k-means clustering while approximately maintaining the original face recognition rate. Experimental results showed that it can speed up existing face recognition systems about 3 times.

Key words: Large data, face recognition system, visual IoT, clustering, acceleration, k-means

INTRODUCTION

After decades of development, face recognition technologies had made considerable progress and many commercial applications had been developed. Due to the special advantages of face biometric, like friendliness, concealment and low expense, face recognition technology could be widely applied in public security, criminal investigation, human-computer interaction, education, entertainment, etc. (Gao *et al*., 2003).

Nowadays, the concept of Visual Internet of Things (VIoT) was well known and played an imperative role in the entire paradigm of Internet systems (Quack *et al*., 2008). It was a kind of Internet of Things (IoT) but first-tier sensors were specified to image or video sensors rather than generalized sensors. VIoT's feature was to obtain images and videos by various visual sensors and it provided visual labels of people, vehicles or objects, which were analogous to the RFID (Radio Frequency Identification) labels. By net transmission and intelligent processing on the labels with related information, valuable relations between them were established to know more (Quack *et al*., 2008).

Based on VIoT's features, face recognition was one of the most appropriate applications of VIoT. All face recognition technologies used visual sensors like cameras or videos, while face feature was the inherent label of human beings. However, it was not easy to put face recognition on VIoT framework or make it run well, since many face recognition technologies could only work well under constrained situations like low scale, tiny illumination discrimination and approximately frontal posture. Among these constraints, the speed of scanning the large scale database was one of the most critical factors that limit the face recognition's application on regional or national platforms (Yan *et al*., 2011). To solve this problem, a general method was proposed to accelerate the face matching while not reducing the recognition rate. This system worked in the VIoT framework. Extensive experimental results showed it could accelerate the speed at least 3 times than before without sacrificing recognition rate much, which was practical in large scale applications.

RELATED WORK

Concepts of VioT: The IoT referred to uniquely identifiable objects (things) and their virtual representations in an Internet-like structure (Jia *et al*., 2012). In other words, it meant a network world connected by things, which consisted of sense, connection and intelligent information processing. As a special kind of IoT, VIoT focused on combing the visual related technologies to traditional IoT, which consisted of cameras, information transmission net and intelligent image or video analysis algorithms. Due to the limitation of current video analysis algorithms, the objects in VIoT were usually restricted to humans and vehicles. Compared with IoT, the biggest advantage of VIoT was its ability to extract labels from the appearance of objects and without the need of RFID or other explicit labels attached on the objects (Jia *et al*., 2012). Based on some state-of-the-art

Corresponding Author: Yaping Lin, School of Information Science and Engineering, Hunan University, No. 252, Lushan South Road, Changsha, 410082, China Tel: 86-731-88821932 Fax: 86-731-88821977

algorithms in biometrics, computer vision or pattern recognition, VIoT could label objects with name, identity, color, shape or other attributes. By combing these labels or information, relations between objects could be established. In scenarios of large scale regional or national network, it was of great significance to social management and public security.

Large scale face recognition: As known, researches on face recognition were usually categorized as two aspects: the algorithm aspect and the system aspect. With the incremental requirements for large scale data applications, efforts on algorithm and system were both valuable to solve the issue of speeding up feature matching while not sacrificing recognition rate.

One of the latest methods was the incremental linear discriminant analysis and hashing bases search method proposed by Yan *et al.* (2011). But its experiments showed only on specific data scale and specific values of searching parameters could that method work well, which was not suitable for practical applications. Wu *et al.* (2010) proposed a method of component-based local feature voting to get a small relevant subset firstly and then a global hamming signature was adopted to re-rank. Since local feature voting was of linear complexity, it was still time consuming when data scale was large enough, like more than millions. Hashing based methods, e.g., the Locality Sensitive Hashing (LSH) (Charikar, 2002; Datar *et al.*, 2004) which had been commonly used in image retrieval, were perhaps the most popular ones but (Wu *et al.*, 2010) found that their performance degraded quickly in incremental database. One possible reason for LSH was that the binarization loses too much discriminative information for face recognition. Su *et al.* (2002) used multimodal part face recognition method based on principal component analysis (MMP-PCA) for accelerating the algorithm speed. But experiments showed this method could not guarantee a high level recognition rate. Besides, from the systematic perspective, Intel's MultiMedia extensions (MMX) technology as an extension to the basic Intel Architecture (IA) (Mitall *et al.*, 1997; Peleg and Weiser, 1996) used by Meng *et al.* (2005) was greatly restricted by the constraints on hardware and did not have universality, although its effectiveness was content for speeding up matching by 14 times. The method proposed in this study could not only keep the recognition rate at a high level, but also speed up the feature matching on the majority of universal server machines.

LARGE SCALE FACE RECOGNITION ON VIoT

The concept of VIoT was proposed based on traditional IoT a few years ago. Currently, clear definition and successful applications had not been developed yet. Here, face recognition was regarded as one of the most appropriate technologies that could be applied in the framework of VIoT. In this section, a face recognition system was implemented in accordance with the concepts and architecture of VIoT.

Face recognition system on VioT: A typical face recognition process included image capture, face detection, face tracking, pre-processing, feature extraction, feature matching and result presentation. The architecture of face recognition system on VIoT was shown in Fig. 1.

Now a prototype system based on this architecture named with IoT Face Recognition System (IFRS) had been developed. As shown in Fig. 1, It consisted of three tiers. Visual sensors included Closed Circuit Television (CCTV) camera, smart mobile phone, pad or personal computer. The category of information transmission net included video network, communication network and the internet, which was the second tier. In the third tier of the system, 1-2 servers were responsible for face recognition processing while a database server was for storing persons' information. Although this system was single-function as one form of VIoT yet, it illustrated the essences of VIoT, such as various sensors, multi-way connections and intelligent information processing module (face recognition). This system could be a good foundation for further improvement.

Analysis of large scale problem: When face recognition was applied in large scale situation, e.g., with millions or

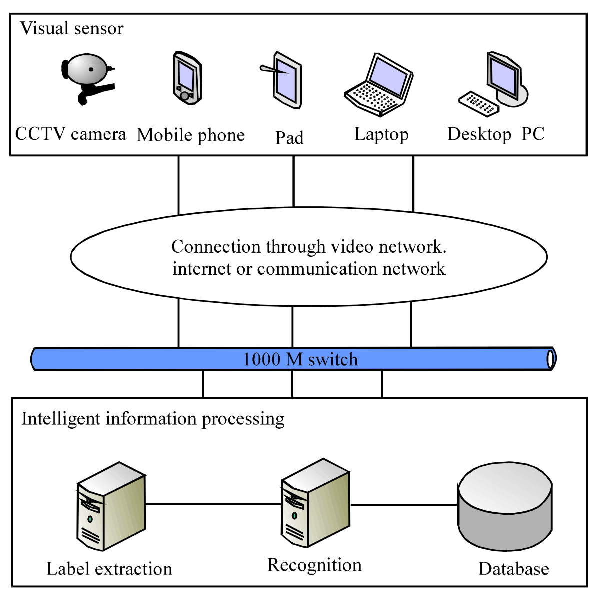

Fig. 1: Architecture of existing VIoT with face recognition application

billions of faces database, the recognition performance degraded badly, both of the response time and recognition rate (Grother *et al*., 2010). From the perspective of engineering, firstly the reasons for large time consuming were analyzed and then a method was proposed to accelerate the recognition process nearly without sacrificing recognition rate.

In a computer system, the consuming time for computation complexity mainly came from two places, CPU load and I/O load, ignoring other tiny data transmission time. CPU load referred to the high cost in application which called for high CPU performance, for instance the parallel requests to a website without I/O procedure, or some computation for complex algorithm. To solve this kind of problem, a simple and effective solution was to duplicate identical servers and add load balance servers to scatter the CPU load. At present the majority commercial websites worked like this, or other more complicated load balance architectures designed with similar principles. I/O load generated system bottleneck mainly because of reading and writing large data on disk frequently. Research showed that the read-write speed of memory was 10^5-10^6 times faster than that of disk, while 100 times faster than disk of data transmission speed (Ito and Tanaka, 2010). If optimizing the architecture by adding duplicated servers simply the same with the CPU load problem, the performance would not be improved. One reason was that the inner speed difference between CPU and I/O determines the slower response time on I/O, the other one was new problem which made the system more complicated and unobservable might emerge from multi-version data synchronization and concurrence risk, especially on super scale database. The general solution was using distributed database, duplicating database servers and bestowing the function of synchronous transaction processing between database servers. Reading operation and writing operation were separated, because writing affairs called for synchronization while reading not. This mechanism was mainly for general websites, such as blog, Bulletin Board System (BBS), etc.

As one special kind of data type, feature matching process was a traversal of all face features each time, which was totally different from traditional data retrieval on distributed systems or website applications by index searching in distributed database. An original solution on one machine for speeding up was to discard relation database's negative impact on speed in matching process and put all features into memory for avoiding I/O load. In many cases face feature vector was an array of some kind of data type. Specifically, it was an array of short data type (2 bytes, 16 bits) with the length of 600 in the system. With an additional identity number, integer type with a length of 4 bytes, a typical feature structure was a length of 1204 bytes. The detail of a feature structure was shown in Fig. 2 with two parts. The left part was the user's ID number occupying 4 bytes within the memory, while the right part was the user's feature data which was the outcome of face extraction from the user's image. The face feature was an array of short data type with a length of 600 occupying 1200 bytes within the memory. So the overall space cost of one user in memory was 1204 bytes for each.

For purpose of accelerating single machine's processing ability, relational database was not used in the matching process due to I/O blocks, which was only involved in querying the detailed personal information through the recognized IDs. All the features and IDs were in memory of a single machine. As a result, the features of 7,000,000 face images would occupy about 8 GB RAM. But memory was limited and there must be an upper bound to the amount of features. Although when the amount of features just reached the upper bound and the system could work not badly, an experimental result showed that the system performance declined as the scale increases.

Figure 3 revealed the relationship between feature scale and consuming time of the feature matching process on a server (Intel(R) Xeon(R) CPU E5606 @ 2.13 GHz and

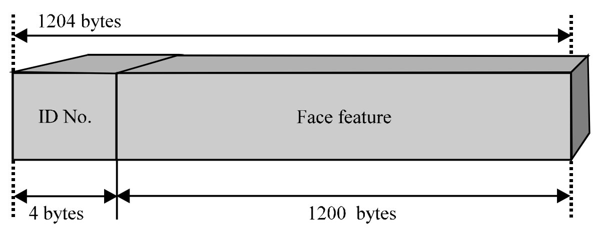

Fig. 2: Data structure of one face feature stored in memory

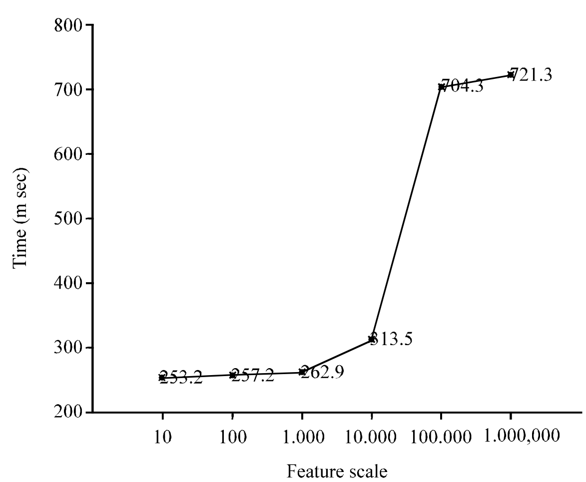

Fig. 3: Time cost for retrieving one face on a single machine under different scales

16.0 GB memory). If the feature scale was lower than 10,000, system's performance was acceptable when searching one person with 300 msec. But when more than 100,000 faces were registered in the system, response time was up to more than 700 msec for searching one person and about 10 sec for ten persons, which was unacceptable in practical applications. So putting data to memory in one machine was still not a final solution and it was the challenge that how to construct a distributed database system for only scanning partial data as little as possible to save time. It seemed practical to partition the overall database to several sub-databases for parallel computing, but this way also needed to scan all the data and how to find out the n most similar faces globally would become a new complicated problem.

As all known, the purpose of matching process was to find out the n most similar faces of input face from feature database. Obviously the Euclidean distance could be the appropriate criteria for computing similarity between features, since face feature was a vector in mathematical point of view. So if the overall features were split to several clusters, meanwhile the Euclidean distances between features were as close as possible inside each cluster and as far as possible outside clusters, the matching process would be cut to handle only features in one cluster, rather than overall features. In this way, k-means clustering algorithm was the perceived and reasonable solution to classify face features. If so, at each matching process, the n most similar faces recognized from one cluster could be regarded as the final result. The matching time would be reduced greatly because only features in one cluster were involved.

In data mining, k-means clustering was a classical method of cluster analysis which aims to partition n observations into k clusters in which each observation belonged to the cluster with the nearest mean. This resulted into a partitioning of the data space into Voronoi cells (Xie and Jiang, 2010). The clusters partitioned by k-means owned characteristics that every observation in the same cluster had high similarity while observations between clusters had low similarity. This was just satisfied of face feature's characteristic that usually features of highest similarity would be found out in once matching process.

Given a set of observations (x_1, x_2, ..., x_n), where each observation was a d-dimensional real vector, k-means clustering aimed to partition the n observations into k sets ($k \le n$) S = {S_1, S_2, ..., S_k} so as to minimize the with-in-cluster sum of squares (WCSS):

$$\mathrm{Argmin}_S \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \quad (1)$$

where, μ_i was the mean of points in S_i.

An iterative refinement technique was used to build our face clusters. Given an initial set of k means $m_1^{(1)}$, ..., $m_k^{(1)}$, the algorithm proceeded by alternating between two steps: (MacKay, 2003).

Assignment step: Assign each observation to the cluster with the closest mean (i.e. partition the observations according to the Voronoi diagram generated by the means):

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\| \le \left\| x_p - m_j^{(t)} \right\| \forall 1 \le j \le k \right\} \quad (2)$$

where, each x_p went into exactly one $S_i^{(t)}$, even if it could go in two of them.

Update step: Calculate the new means to be the centroid of the observations in the cluster:

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j \quad (3)$$

Commonly used initialization methods were Forgy and Random Partition (Hamerly and Elkan, 2002). The Forgy method randomly chose k observations from the data set and used them as the initial means. The Random Partition method first randomly assigned a cluster to each observation and then proceeded to the Update step, thus computing the initial means to be the centroid of the cluster's randomly assigned points. The Forgy method tended to spread the initial means out, while Random Partition placed all of them close to the center of the data set (Hamerly and Elkan, 2002). And because face features' means should be widely spread for distinguishing clusters, Forgy method was chosen.

Proposed method and system: Since each face feature was a 1200 dimension vector of short data type (16 bits) in recognition algorithm of VIoT, clustering process was actually a mathematical computation to determine which faces should be partitioned into the same cluster. So in order to satisfy that process, a robust system was designed including three parts, pre-process module, retrieval module and data management module, respectively. Pre-process module initialized all the face data and made it prepared for accepting retrieval requests from retrieval module, while data management module responded to requests of inserting, updating and deleting face data.

Pre-process module was to partition the original large-scale face data to several clusters. The parameters, e.g., the number of clusters and the maximum face amount of each cluster, depended on real situation and machine's ability and in the next section this method's performances would be given on different parameters. This module was also responsible for making the overall database prepared for accepting requests from retrieval module, which always ran before or at the first time when the overall system was launched. As shown in Fig. 4, pre-processing module consisted of four steps. The first step was to set relevant parameters of the system. Then for every picture, this module did face detection and feature extraction, calculating the distances between picture and each mean to determine which cluster was the closest one to the picture and assigned the picture to the specific machine representing that cluster repetitively. After all the steps were completed and all the pictures in training data were initialized, the whole system was prepared well for requests.

Retrieval module and data management module both generated requests to the large database but were for different kinds of purposes. Retrieval module was for face retrieval process including face detection, image pre-processing, feature extraction, computation of which cluster the input face belonged to, finding the n most similar faces from the specific cluster and returning results to requestors. Data management module was designed for responding modification requests and reconstruction of database. Under most of situations, once the database was initialized, structure of clusters should not be changed frequently due to its high cost of reconstruction. Also the recognition results highly depended on the clusters' structure, so a management module was needed for handling the alteration like inserting one face, modifying one face or deleting one face. The approximate process was shown in Fig. 5.

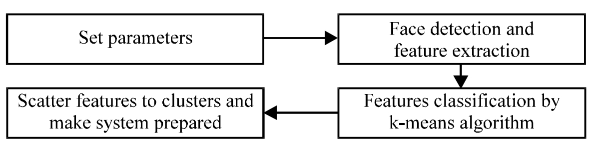

Fig. 4: Procedure of pre-process module in a large scale face recognition application

According to Fig. 5, there were some public steps among retrieval module and data management module. For searching one person or modifying the database, steps of image capture, face detection, feature extraction and computing Euclidean distance with each mean were essential. Then, if the request was for searching, it turned to retrieval module and found the targets with feature matching one by one namely nearest neighbor method, while if it was for modifying, it turned to data management module and did reconstruction of the overall database or modification of one feature directly, depending on the judgment whether the number of change times was larger than a threshold.

The three modules were designed as multi-tier mode based on Hypertext Transport Protocol (HTTP) thanks to its good manageability and stability. The whole system was much similar to IFRS, but improvement was done on

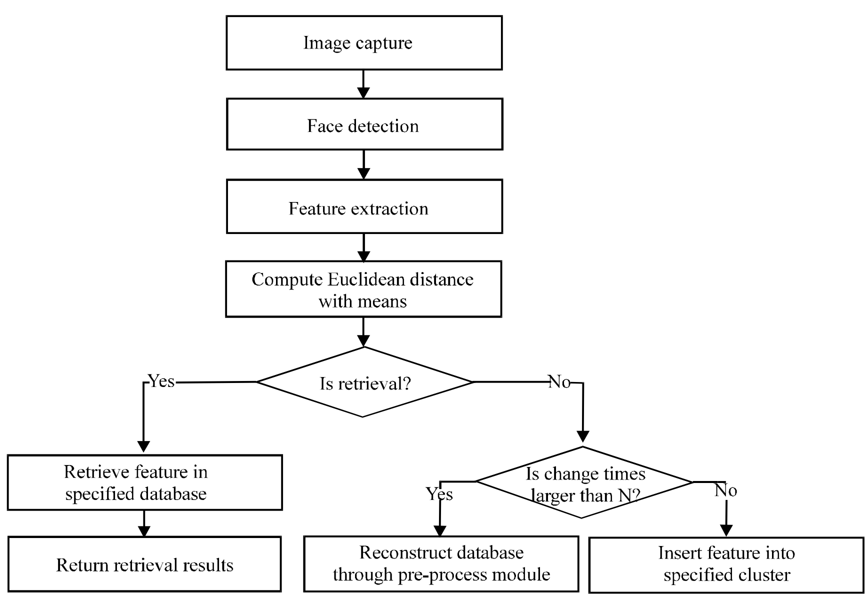

Fig. 5: Procedure of retrieval module and data management module in a large scale face recognition application

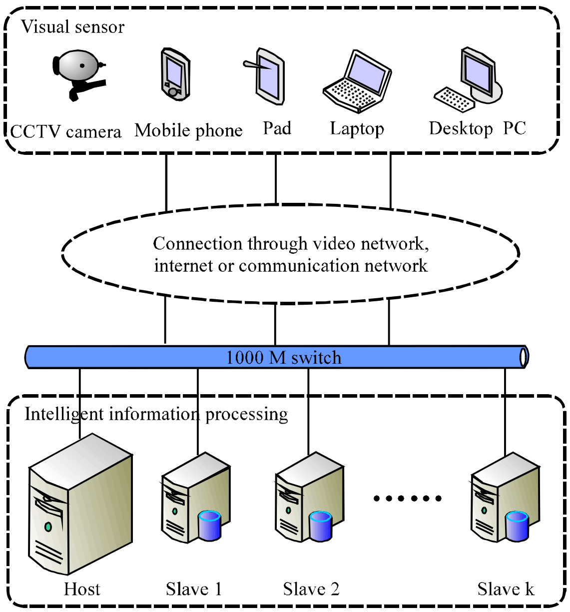

Fig. 6: Architecture of a large scale recognition application on VIoT

the intelligent information processing. As a website, it would not be told about how users accessed this face recognition service through browser or other visual sensors. Here the background server architecture was focused on, tier of intelligent information processing, on VIoT and made it a special kind of distributed database system effectively, host-slave mode. The main system architecture was shown in Fig. 6.

The mainly difference between Fig. 6 was the background servers which consisted of distributed machines based on k-means algorithm principle, rather than independent machines with simplex function. The all server machines were also connected by 1000 M network switch, so requests' elay time could be almost ignored. When initializing the database in pre-process module, the main face database (MFDB) was split to k sub-databases (SFDB) by k-means algorithm. The k means together with ids mapping between face id and cluster id were stored on the host machine. Then the overall sub-databases were prepared for accepting requests for feature matching from retrieval module. Let $SFDB_i$ donated the ith sub-database and the relationship between MFDB and $SFDB_i$ could be defined as follow:

$$\text{MFDB}=\sum_{i=1}^{k}\text{SFDB}_i, \text{SFDB}_i \subseteq \text{MFDB}, i=1,\ldots,k \tag{4}$$

where, k means the number of means, also clusters.

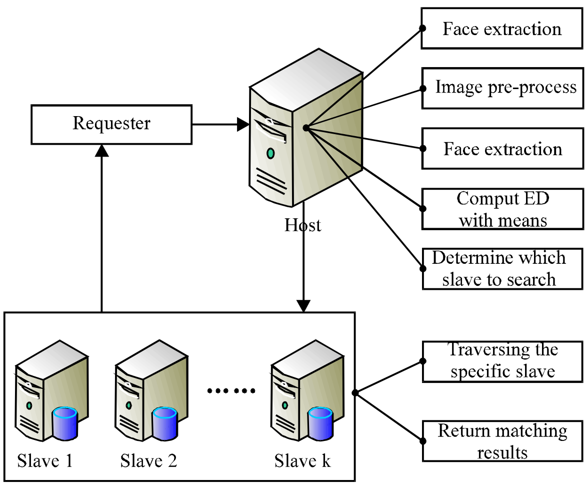

Fig. 7: Overall procedure of a large scale face recognition application

On a general face recognition retrieval process in this system, the host machine received the request with original image data firstly. After face detection, image preprocessing, feature extraction and computing Euclidean distance with means on the host machine, it was determined that which slave the request should be transformed to for retrieving faces. Then the appropriate slave machine received the request and returned the results to requester after feature matching process honestly. The detailed process is shown in Fig. 7.

Improved method: Experimental results showed that although the recognition process was accelerated remarkably with increase of the number of clusters, k-means did not contribute to the recognition rate. Test showed once the classification process found out the right cluster for target face, it could be recognized correctly. So recognition rate's decline was influenced much by classification result.

If only one face image for every different person was mixed into the large scale database, with the increase of clusters, many test samples were classified to wrong cluster and then the following traversal matching was actually wasted efforts. In order to enhance the accuracy of classification, data redundancy in each cluster was an idea. If additional data were added to initialized database artificially, the recognition rate seemed not ugly, but it was not our original intention, since supposing adding every target face into every cluster was meaningless. So some redundant face data was mixed to every target person in the main database before splitting it to sub-databases. Results showed it was one good way to guarantee recognition rate not to decline greatly.

EXPERIMENTAL RESULTS

The overall data source consisted of three parts; the first one was the base large scale database, the second one was target faces for being retrieved out and the last one was used for test as input samples. The first part was standard face image on identity card and it was confidential. The last two were collected from the Facial Recognition Technology (FERET) Database (Phillips *et al*., 2000). The first part and the second part were mixed together to constitute the original data for classification. Then they were scattered in k slave machines by rules of k-means algorithm. For just manifesting the relative promotion in speed and stability in recognition rate, also due to limited right from government, in our experiments 500,000 identity card faces were adopted as large scale database, the first part. About 3,000 pictures were picked up from FERET Gallery (Fa), Probe set fafb (Fb), Probe set fafc (Fc), Duplicate I (dup1) and Duplicate II (dup2) of FERET. Fa was mixed in large database. In tests, the number of k ranges from 1 to 100. When k is equal to 1, it just meant the previous unimproved method and when k was larger than 1, test results showed performance under different situations. Here two indexes were collected for judging the performance. They were face recognition rate and acceleration ratio. The former represented practical value under different ks. The last meant the ratio between the consuming time for running over all test samples when k is equaled to 1 and that when k was larger than 1, which showed the acceleration ability under different ks.

Single picture tests: This experiment included 4 tests for different test data sets like Fb, Fc, dup1 and dup2. In each test, only one face image of each person was mixed in large scale database. The 4 results were similar, so Fig. 8 and 9 showed the Fb result as a represent. Since only one target face was mixed in database, the final recognition result was critically relied on classification process. 1187 pictures were mixed in large database while 1177 pictures of Fb were used for test. And for every test picture, only one picture must be found out from database for judging if it was the right one. The result was follow.

Figure 8 showed the face recognition rate declined so fast before k is equal to 10. After k is equal to 20, the average face recognition rate was about 93% which was very close to that when k is equal to 1. However, the face recognition rate still could not be maintained at higher level because only one face image of every person could not make sure every classification was right result.

Figure 9 showed the acceleration ratio would be steady at about 3.1 after the beginning fast lifting before k is equal to 20. With the increase of k, especially about near to 100, the acceleration ratio might be a little of decrease but still much faster than that when k is equal to 1. Seeing from Fig. 8 and 9, maybe when k is equal to 20 was the best value for cluster number.

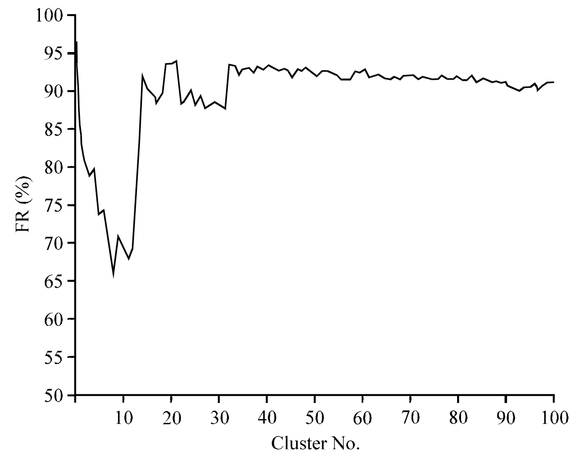

Fig. 8: Results for face recognition rate test under single registered face image situation with k ranging from 1 to 100

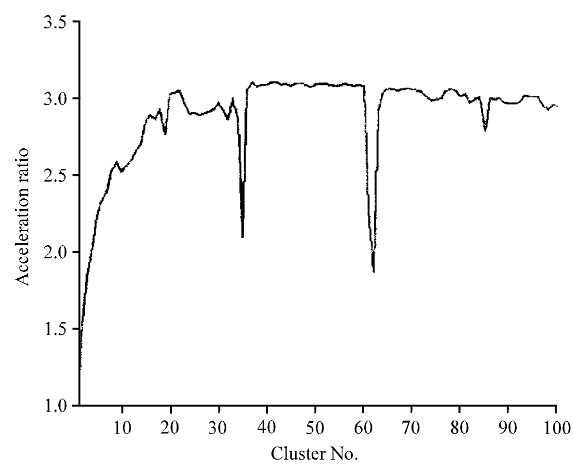

Fig. 9: Results for acceleration ratio test under single registered face image situation with k ranging from 1 to 100

Multi picture tests: Similar to the former experiment, there were 3 tests for different data sets because Fa and dup1 were mixed in large database and Fb, Fc and dup 2 were regarded as test samples. The classification results might be more reasonable. But for ensuring every person in test samples could be found out in database, after excluding images which would not be recognized definitely, 950 pictures were selected for being mixed in large database and 241 pictures were used for test. The result was follow.

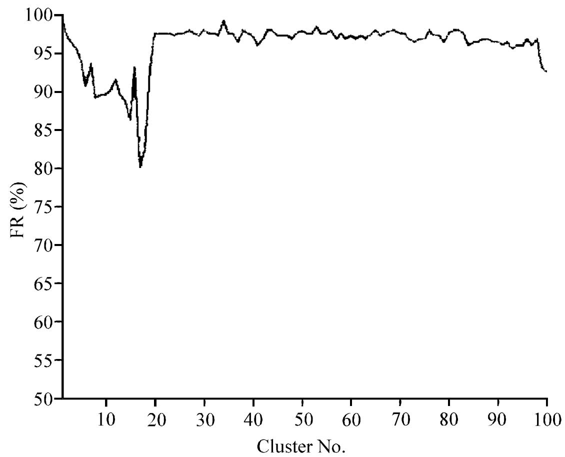

Fig. 10: Results for face recognition rate test under multi registered face images situation with k ranging from 1 to 100

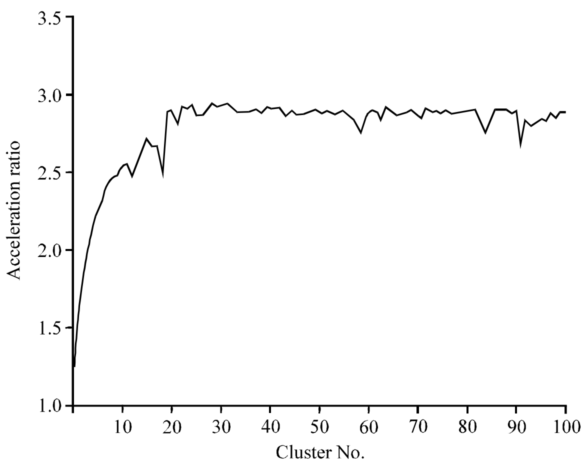

Fig. 11: Results for acceleration ratio test under multi registered face images situation with k ranging from 1 to 100

Figure 10 showed after the beginning dropping and fluctuating, the face recognition rate maintained relatively steady after k equaled 20, the average value was as high as 97%. Multi pictures of one person contributed much to maintain the face recognition rate.

Seeing from Fig. 11, also after k is equal to 20, the acceleration ratio value stayed about 2.9. Maybe when k equaled 20 was also the best point for the overall performance.

It was a tradeoff between recognition rate and time. Under situation of single picture, acceleration ratio might be higher, but face recognition rate was lower than under situation of multi pictures. Absolutely, our method and its improved version had obtained the goal for acceleration of face recognition process on VIoT, compared with when k was 1.

CONCLUSION

In this study a high-speed and robust face recognition system was proposed in the framework of VIoT, whose improved version was more practical in actual scenarios. The tradeoff between acceleration ratio and recognition rate could satisfy various requirements from users. According to user's at actual scale of database and machines' performance, parameters of clusters could be adjusted freely.

On the other hand, this proposed method was unrelated to details of face recognition algorithm. In a word, as long as the extracted face feature was a multi-dimensional vector by any face recognition algorithm and when it came to large scale problem, this method was a good choice. So this method could not only be a practical system, also a platform for further academic experiment and teaching.

ACKNOWLEDGMENT

This study was partially supported by the research foundation of Chinese Ministry of Education and China Mobile Communications Corporation (MCM20122061).

REFERENCES

Charikar, M.S., 2002. Similarity estimation techniques from rounding algorithms. Proceedings of the 34th annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montreal, Canada, pp: 380-388.

Datar, M., N. Immorlica, P. Indyk and Y.S. Mirrokni, 2004. Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the 20th Annual Symposium on Computational Geometry, June 8-11, 2004, Brooklyn, NY., USA., pp: 253-262.

Gao, W., S.G. Shan, X.J. Chai and X.W. Fu, 2003. Virtual face image generation for illumination and pose insensitive face recognition. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Volume 3, April 6-10, 2003, Hong Kong, pp: 149-152.

Grother, P.J., G.W. Quinn and P.J. Phillips, 2010. Report on the evaluation of 2D still-image face recognition algorithms. NIST Interagency Report No. 7709. http://biometrics.nist.gov/cs_links/face/mbe/MBE_2D_face_report_NISTIR_7709.pdf

Hamerly, G. and C. Elkan, 2002. Alternatives to the k-means algorithm that find better clusterings. Proceedings of the 11th International Conference on Information and Knowledge Management, November 4-9, 2002, McLean, VA., USA., pp: 600-607.

Ito, N. and S. Tanaka, 2010. Large-Scale Web Service Development Technology. Gijyutsu-Hyoron Co. Ltd., Tokyo, Japan.

Jia, X., Q. Feng, T. Fan and Q. Lei, 2012. RFID technology and its applications in Internet of Things (IoT). Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks, April 21-23, 2012, Yichang, China, pp: 1282-1285.

MacKay, D.J.C., 2003. Information Theory, Inference and Learning Algorithms. In: An Example Inference Task: Clustering, MacKay, D.J.C. (Ed.). Cambridge University Press, Cambridge, UK., ISBN-13: 9780521642989, pp: 284-292.

Meng, K., G.D. Su, C.C. Li, B. Fu and J. Zhou, 2005. A high performance face recognition system based on a huge face database. Proceedings of the 4th International Conference on Machine Learning and Cybernetic, Volume 8, August 18-21, 2005, Guangzhou, China, pp: 5159-5164.

Mitall, M., A. Peleg and U. Weiser, 1997. MMX™ technology architecture overview. Intel Technol. J., 1: 3-15.

Peleg, A. and U. Weiser, 1996. MMX technology extension to the Intel architecture. IEEE Micro, 16: 42-50.

Phillips, P.J., H.J. Moon, S.A. Rizvi and P.J. Rauss, 2000. The FERET evaluation methodology for face recognition algorithms. IEEE Trans. Pattern Anal. Mach. Intell., 22: 1090-1104.

Quack, T., H. Bay and L. Van Gool, 2008. Object recognition for the internet of things. Proceedings of the 1st International Conference of the Internet of Things, March 26-28, 2008, Zurich, Switzerland, pp: 230-246.

Su, G., C. Zhang, R. Ding and C. Du, 2002. MMP-PCA face recognition method. Electron. Lett., 38: 1654-1656.

Wu, Z., Q. Ke, J. Sun and H.Y. Shum, 2010. Scalable face image retrieval with identity-based quantization and multi-reference re-ranking. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, June 13-18, 2010, San Francisco, CA., USA., pp: 3469-3476.

Xie, J. and S. Jiang, 2010. A simple and fast algorithm for global K-means clustering. Proceedings of the 2nd International Workshop on Education Technology and Computer Science, Volume 2, March 6-7, 2010, Wuhan, China, pp: 36-40.

Yan, J., Z. Lei, D. Yi and S.Z. Li, 2011. Towards incremental and large scale face recognition. Proceedings of International Joint Conference on Biometrics, October 11-13, 2011, Washington, DC., USA., pp: 1-6.