A Massive Face Recognition Algorithm Based on the Hadoop Technique

Wei Li, Ming-Ming Li

Abstract

This paper is focus on the massive face recognition. We want to recognize a face from a lot of faces at a public location. There are three kinds of techniques used to hit the goal: one is the 3D face detection technique, the other is to detect a face by more than one internal image, and the third is the Hadoop technique. Face detection technique finds a similar face by 3D face features from mass face data. Comparing to 2D face features, 3D detection could avoid the phenomenon of using photos instead of the true face and is more accurate. According to previous studies, it was found that comparison of multiple internal images with the real time photos of the same person can greatly improve the accuracy of a facial recognition algorithm. However, this way could greatly consume processing time of the computer and ultimately makes the system unusable. In this paper we adopt a Hadoop parallel processing architecture to solve the computing ability problem. From simulation results, it is demonstrated that the face recognition velocity adopting the cluster consists of four servers is about 3.5 times of the velocity adopting a single same configuration server. Furthermore, this architecture could be scaled easily when users increase. Our algorithm is an effective and correct method for the massive face recognition.

Keywords: Massive, Face Recognition, Hadoop, Features, Parallel Processing

1. Introduction

Face detection is an important task for several applications on human life. There are some research published and described below. Yang and Huang [1] detected faces by using a hierarchical knowledge-based method. They used three level resolutions in their algorithm. The coarse-to-fine strategy to reduce the computation is advantage in this method.

*Corresponding Author: Ming-Ming Li
(E-mail: limmwork@aliyun.com)

Leung et al. [2] used a local feature detector and random graph matching techniques to create a probabilistic method to locate a face in a scene. They used five features (two eyes, two nostril, and nose/lip junction) to depict a typical face. They defined a facial template and relative distances of any pair of facial features. This method can detect whether the testing object is a human face or not.

Yang and David [3] surveyed the detecting faces in images. They provided a general and complete face detection technique. It is a valuable method for face detection.

For face image comparison, it is a hard work because it needs a lot of computation and it cannot achieve 100% accuracy. If we want to improve the comparison accuracy, the multiple face in different angle must achieve the goal. Zhang Xiaohua and Shan shiguang[4] proposed a multi-angle camera image acquisition system. Usually it need to collect 39 photos of a same person at different angles with different expressions and different ornaments. The more the internal image, the more accurate the face recognition algorithm, and the more the computing time. For this reason, the traditional face recognition technology is confined to a small range, stationary state, and single person face recognition research. Furthermore, if we want to search a people from railway station or bus station, it is more difficult because it is related to problems in real time operation. It is a huge computation. However, the parallel processing technique increases computation ability. For a huge data, it needs massive computation ability; it needs several computers working together to share the computing jobs. Therefore, Hadoop structure is a suitable system for solving the massive face recognition problem [5,6].

The Apach Hadoop project develops open-source software for reliable, scalable, and distributed computing.

The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from a single server to thousands of machines, each offering local computation and storage. Rather than relying on the hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer so to deliver a highly-available service on top of a cluster of computers, which may be prone to failures. HDFS，Mapreduce and Hbase are the three main

modules of Hadoop[7]. Now, Hadoop has become the de facto standard in the field of big data parallel computing.

The main goal of our work is to develop a simple and effective face detection system on the massive video sequence. The remainder of this paper is organized as follows; section 2 describes the face recognition algorithm and its detail process; the Hadoop system is described in the section 3; experiment results are shown in Section 4; finally we conclude this paper in Section 5.

2. Face Recognition

2.1 Object Extraction from the Video Stream

The first stage of the face recognition is object extraction. Once the individual face is obtained, the face identify can be execute correctly. Figure 1 shows the the flow chart of the object extraction from the video stream. Since the video input video is a color stream, we transfer the color image into grayscale image for reducing the computer computation. RGB2YIQ stage is used to map the color image into YIQ component image. Since Y component has 90% energy of the image, it was used for the following task. Next, a background removal stage was used to extract the object from a video. A media filter was used to filter out the noise. An exclusive bitmapping (EB) technique was used to convert the image from grayscale to binary even on differently exposed environments. Usually, the object image extraction will cause some distortion, so image processing (IP) techniques like the dilation, opening, and closing techniques were used to remedy the drawback. Finally, a small area discard stage was used to delete those objects belonging to noises because their areas are so small.

In addition, by the person structure; we easily extract the human face location by a scale method of object. We compare the face location with the previous frame to make sure the object is brand new or repeated. Once the objects were all examined, we refreshed the previous frame and recorded the new objects. Finally, we normalized the object and run the next frame. The details flowchart is described in Figure 1.

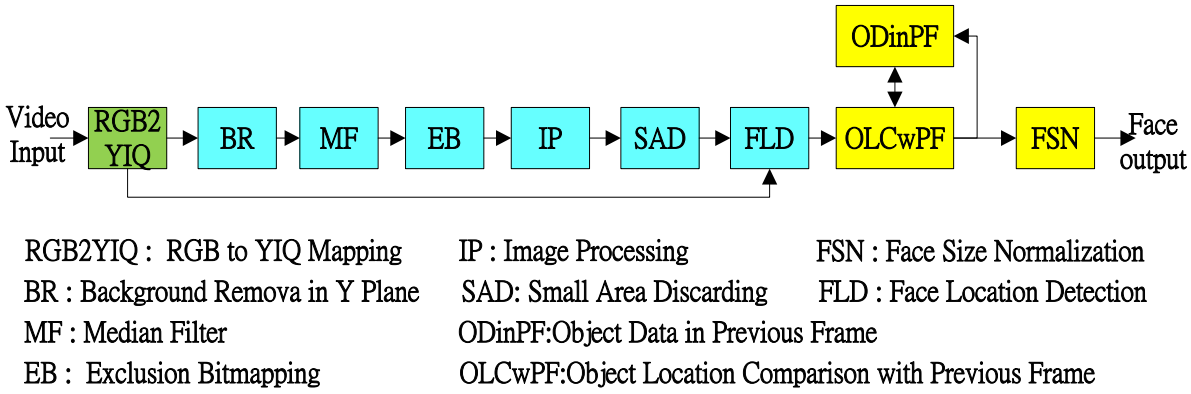

Figure 1: The flow chart of the object extraction from the video stream

2.2 Three-D Face Recognition Algorithm

The 3D face recognition is better than 2D because 3D method has lots of features. In this paper, we used the skin color and principle component analysis techniques to achieve the face recognition. At first, we got the left and right images that included face components. Next, the color image normalization stage was used for further skin color detection. In order to obtain the face area, the several stages including erosion, dilation, labeling and padding, were used to hit the goal. However, the size normalization stage is necessary to make the face image standard. Finally, 3D face recognition can be completed by using following three steps: (1) merging the left and right image form a 3D image; (2) executing the principle component analysis to get the features; (3) verifying the features by comparing the input image and database. Figure 2 shows the details of the system.

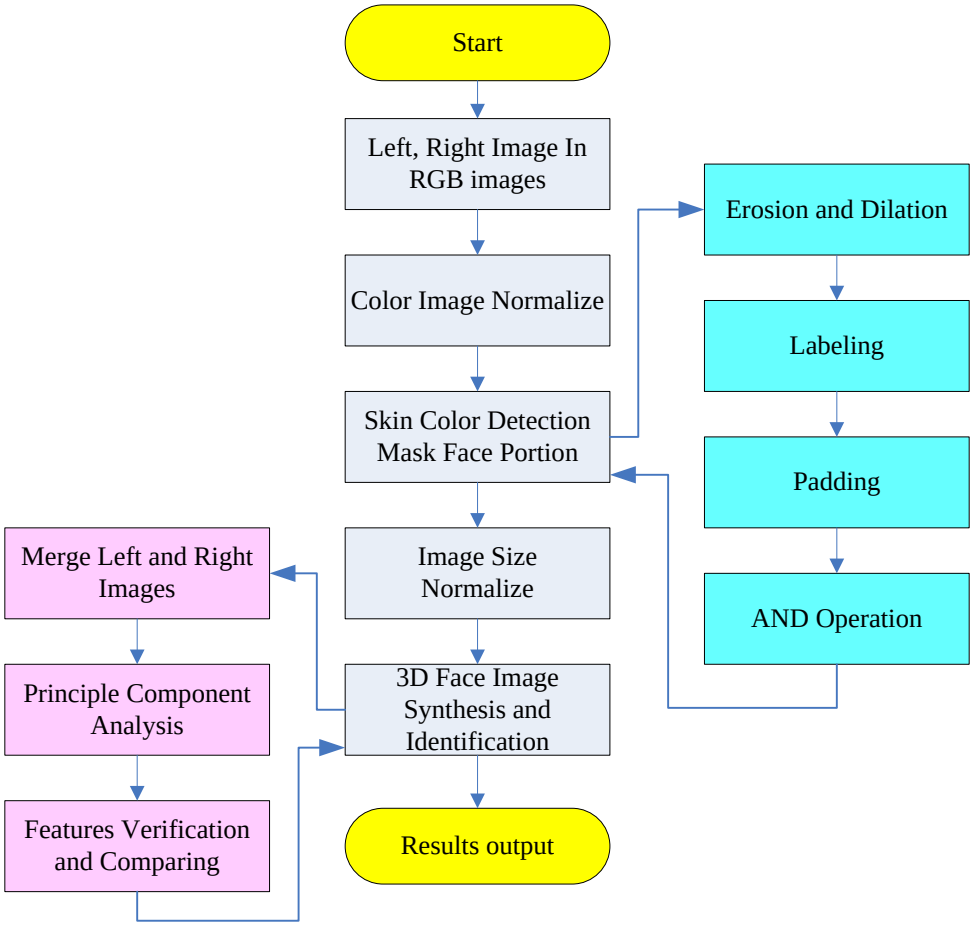

Figure 2: The flow chart of the 3D face rewcognition

2.3 The Features and Decision Function

The face features in this paper have 15 numbers as shown in Figure 3. It is divided into three parts: eye feature, nose feature and mouth feature. The important points of the face are denoted in the face image listed as A to J. The fifteen features are the distance between two points as shown in the Figure 3. For example the distance A and B express the feature, the distance between two eyes. The feature IJ denote the width of the mouth. The feature EK is the distance between the centre of two eyes and mouth. The detail feature point vector definition is shown in Table 1. The distance between the feature points are obtained from the European Curie using the mathematical formula (1) - (3).

Table 1: Feature point vector definition table

$\overline{AB}$	Two eye spacing	$\overline{EH}$	Two eye center-to -nose distance	$\overline{EK}$	Two eye center to mouth distance
$\overline{CD}$	Two eye edge distance	$\overline{EG}$	Right nose to the two eyes center distance	$\overline{EJ}$	Right from the mouth to the two eye center distance
$\overline{FG}$	Nose width	$\overline{DG}$	Distance from the right nose to the outer edge of the right eye	$\overline{DJ}$	Right from the mouth to the outer edge of the right eye distance
$\overline{IJ}$	Mouth width	$\overline{EF}$	Left nose to the two eye center distance	$\overline{EI}$	Left mouth to the center of two eyes distance
$\overline{HK}$	Distance from the nose to the mouth	$\overline{CF}$	Distance from the left nose to the outer edge of the left eye	$\overline{CI}$	Distance from the outer edge of the left eye to the left mouth

$$d = (y, y_i^k) = \|y - \alpha_i^k\| \quad (1)$$

$$d(y,k) = \min[d(y,\alpha_1^k), d(y,\alpha_2^k), \cdots\cdots, d(y,\alpha_N^k)] \quad (2)$$

Decision Function:

$$ID_y = \arg\ min_k\ d(y,k) \quad (3)$$

Symbol Description of Equation (1) -(3):

Suppose the feature sample of test image is y, database consists of images which are divided into C categories, and each category has N pieces of images which have been trained; these characteristic parameters are $\{\alpha_i^1, \alpha_i^2, \cdots\cdots, \alpha_i^N\}$, and $1 \le k \le C,\ 1 \le i \le N$,so the Euclidean distance between the tested image y and the database image α_i^k are obtained by Formula (1); and the minimum Euclidean distance between characteristic parameters of test image y and category k are obtained by Formula(2); we use IDy to delegate the people of test image, and the decision function to get IDy are shown in Formula (3).

Euclidean distance：This method is to compare the sample image and the training image to find consistent image; that is to get the shortest distance between the sample image vector and the trained image vector.

In this article, before performing this Euclidean distance method, at first comparison between each sample image and the training image is attained to improve the recognition rate, and identification methods are used to compare every small characteristic feature between test image and trained image; when the difference of one item is greater than the threshold value, the test image will be determined to conform so to be eliminated by the following formula.

Any small term feature vector difference

$\geq$ Sth(Threshold value) $\Rightarrow$ Eliminate

Any small term feature vector difference

$<$ Sth(Threshold value) $\Rightarrow$ Retain

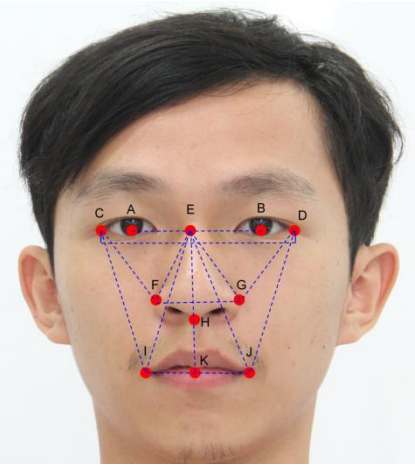

Figure 3: The sketch of the face feature

There are two steps for the decision function; one is the error of the feature that the input image and database is under the threshold. The other is that the error of the features is the minimum. The first step is to check the features if one of features is over the threshold so that it will die out. The second step is to calculate the features error that is the error by comparing the input image with the database. Once the accumulation error is the minimum, it is the answer.

3. Hadoop System

Since the 3D face recognition from a video stream is computation consumption, a big data processing technique is needed to achieve the goal. For our case, the Hadoop technique is suitable for massive face recognition.

3.1 Brief Introduction of Hadoop

Hadoop has a distributed storage and computing platform suitable for massive data, such as above PB level massive data. In Hadoop, a computing task will be assigned to multiple virtual machines to handle.

As the de facto standard of Cloud Computing, Hadoop is a large ecosystem. There are many technologies and products in this system, such as HDFS, Mapreduce, Hbase, Zookeeper, Hive, Floom, Sqoop etc. The core module is HDFS, Mapreduce, and Hbase. Figure 3 is the Hadoop ecosystem diagram. We only introduce the HDFS, MapReduce and Hbase briefly, and these three modules will be used in the study.

All the Hadoop modules are built on the distributed file system, which is HDFS-Hadoop Distributed File System. Currently, the most widely known distributed file system should be the Network File System (NFS). HDFS are different from NFS on many levels, especially in the data security and system scalability and cost advantage. HDFS is designed based on the master-slave structure. Only one master node is responsible for recording the metadata. The metadata only stores file structure, but actual file data. The actual file data is stored in the Data Node. By default a file stored will be 3 copies in different data node of HDFS; if one node is shutdown, the file can still be accessed from other two data nodes. The probability of 3 data nodes being shut down at the same time is very little, so the data stored in HDFS is very safe. Hadoop was designed with consideration to the scalability. As the amount of data increase, we just add a data node into the cluster. Hadoop cluster is built on the assumption that every node in the cluster will be damaged. If one node is damaged, the data and computing task will be migrated to other node. Single node damage will not affect the whole cluster.

MapReduce is a parallel distributed data application framework or library. It allows ordinary programmers to program distributed applications without knowing the underlying details of distributed processing architecture. Like HDFS, its architecture is based on a master-slave mode. Master is a special node to coordinate activities between multiple job nodes. The following is how it works. The master receives the input data to be processed. The input data is divided into smaller blocks, and all the blocks are processed paralleling on multiple distributed job nodes. This step is called "mapping." Working node returns the results to the master, and master starts another task to aggregate these result, which is called "reducing". The reducing task also run on multiple distributed job nodes in paralleling.

Hbase is a nosql database running on Hadoop, which is a distributed and scalable big data ware houses. HBase can use HDFS distributed processing mode and benefit from Hadoop's MapReduce program model. This means that it can store many large tables that have billions of rows and columns of millions in a set. In addition to Haoop advantage, Hbase itself is very powerful, and able to integrate key/value of the store model for bringing real-time query capability, and the ability to batch process through MapReduce or offline.

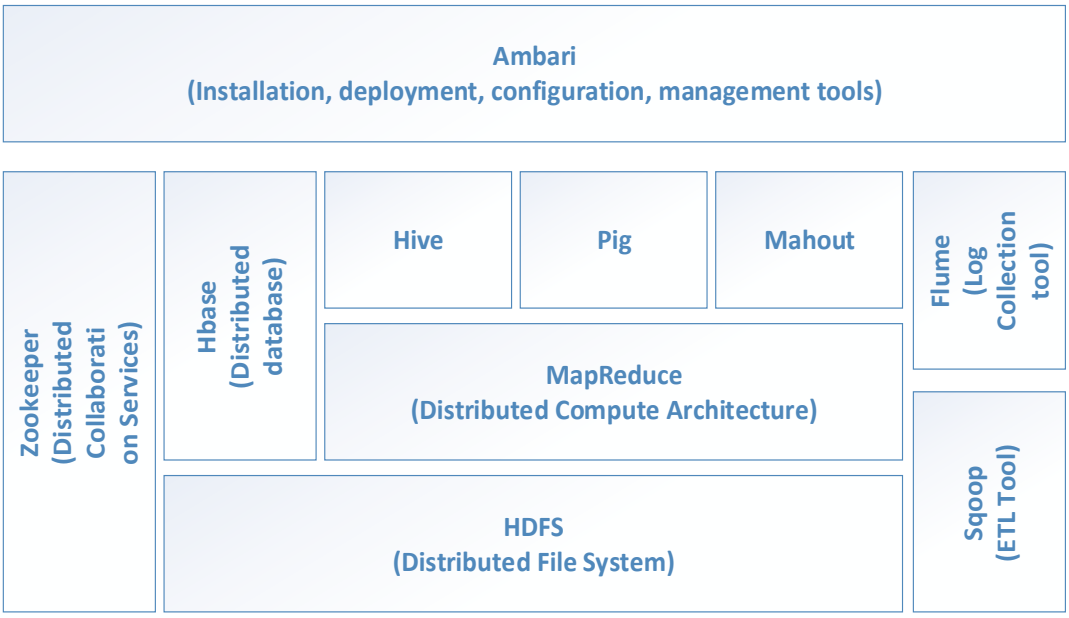

Figure 4: Hadoop ecosystem module

3.2 Massive Face Recognition System Structure

A massive face recognition system was used to obtain man information from lot of peoples, like the personal identification in a railway station. For this case, to find a people from massive peoples, it is a big data problem which needs a technique to reduce the complexity and finish the job.

Hadoop structure is an effective parallel processing technique which can solve big data problem. Figure 5 shows the whole system structure used in the massive face recognition. It is obvious that there are two module blocks; one is the subscriber verification, and the other is the subscriber increase.

The task of the subscriber increase module is to receive the subscriber information. The subscriber increase module passes through the camera to get the face image and use it to add new subscriber. For accuracy, the photographs include different light environments, different backgrounds, different angles and facial expression.The people's information is stored in HBase, and the photographs are stored in HDFS for map function to compare face recognition. The subscriber information is recorded in the database including the name, id, age and photograph.

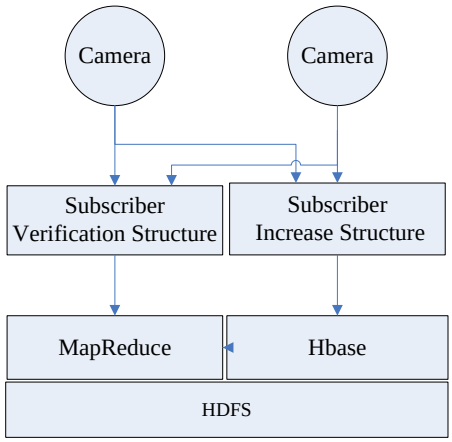

Figure 5: The implement system

The task of the subscriber verification module is search and identifies the input face image with the MapReduce. It make sure the exactly people by comparing all the image, and pick up the one is the least features error. This action need immense computation, therefore, it is need the Map and reduce techniques under parallel process to accelerate operation speed.

3.3 The implement system

The implement system of the Hadoop structure is shown in Figure 6. It includes two processes; Mapping and reducing. The nodes of the Hadoop structure can be failure. If a node is on failure state then the task need to be reassigned. If the task include some side effect then the share state need to restart. For example, the nodes communicate with the outside node, then the share state must be hold until the system restart.

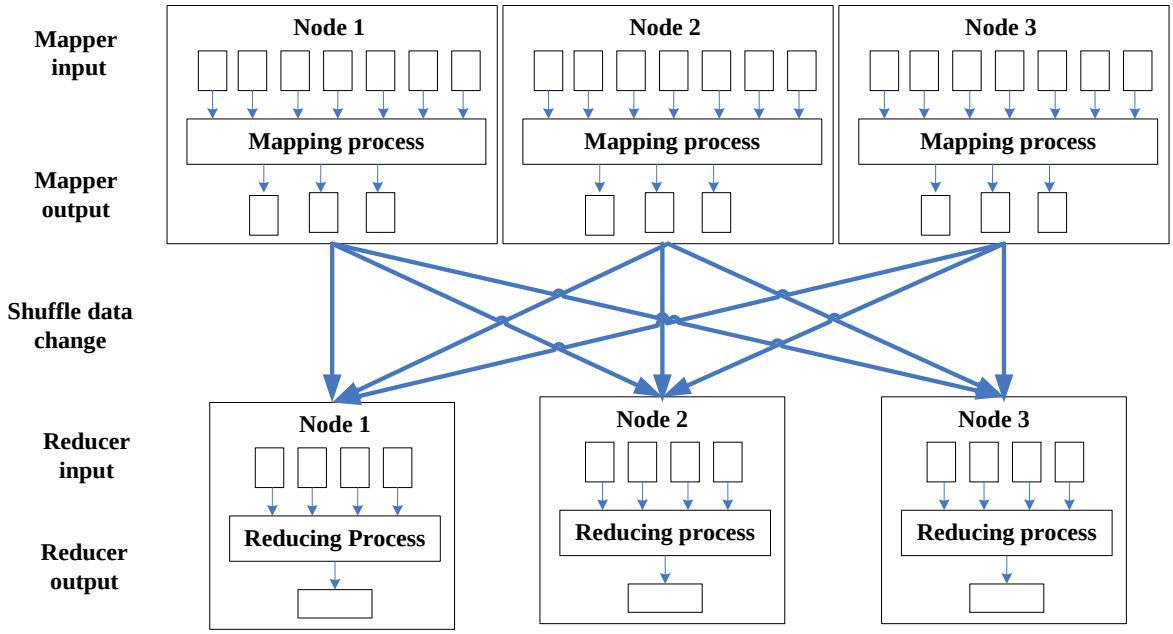

Figure 6: The implement system of the Map/Reduce stream

In Hadoop structure, just only inheriting the MapReduceBase class, it can offer map and reduce

classes. Besides, the Hadoop structure can automate operations in distribution state when it is at register Job. For example, we have 90 pictures to be compared with 2 million subscribers each of which has 20 pictures in the system. If each picture size is 2048 by 2048, then the size of every image is equal to 2048*2048=12.5MB. Therefore, we must carry on 2000000*20*90=3600000000 times of comparing 12.5MB image. This is a huge computation task.

Hadoop adopts parallel operation technique. It uses the multiple servers of the cluster to compare with several pictures. This can reduce the time for identification. For example, we set 4 servers in a cluster; let the task of the Map be 25, and the task of the Reduce is 4. If the waiting subscriber is assigned to the 25 mappers, then the jobs of each server mapper is 1/25 of the whole jobs. Considering the task schedule consumption, the task of a server is about 1/3~ 1/4 of all the computing jobs in one server with same hardware configuration.

The process of the Map is listed below. It uses the subscriber id and picture number as the input key, and the subscriber picture as the input value. Simultaneously, use the subscriber id and picture number used as the output key. The comparison results between the subscriber picture and input image are used as the output value. The input of the Reduce process is the subscriber id and picture number. The target is to obtain a maximum value of comparison results by using input key and front half subscriber id. On the other hand, the Reduce process uses the subscriber id and picture number as output key, and the similarity of the test image and subscriber as the output value.

According to the Reduce output, the system sorts the similarity of all the subscribers to get the most similarity subscriber id and picture number.

When the subscribers increase, the system just adds the cluster class to support more operation. This is the advantage of the big data techniques.

3.4 Algorithm Improvement

In the algorithm talked above, we suppose the subscriber images are stored in HDFS to get the feature of every image in the mapping process. If there is only one comparison, this is no problem. However, a real system will be used many times, and every time the real time image will compare with the 40000000 internal subscriber images. There is a lot of repetition in the feature value calculation. Because the feature value is independent of the real time image. Therefore, we calculate the feature value of every subscriber image in the process of getting image, and store the image into the HDFS, but store the feature into the HBase.

Table 2: The Personal INFO Table

rowkey	10 Byte row key. From 0000000000 to 9999999999，can surport 10billion subscriber totally	
Basic	No	Identify No
	Name	Name
	Birthday	Birthday
	Sex	Sex
Contact	Email	Email
	Mobile	Mobile
	Line	Line
	Wechat	Wechat
	Tel	Tel
	Zipcode	ZipCode

Table 3: Image Info Table

rowkey	12 Byte row key. From 000000000000 to 999999999999，surport 10billion subscriber,99 images maximum
$\overline{AB}$	Two eye spacing
$\overline{CD}$	Two eye edge distance
$\overline{FG}$	Nose width
$\overline{IJ}$	Mouth width
$\overline{HK}$	Distance from the nose to the mouth
$\overline{EH}$	Two eye center-to -nose distance
$\overline{EG}$	Right nose to the two eyes center distance
$\overline{DG}$	from right nose to outer edge of right eye
$\overline{EF}$	Left nose to the two eye center distance
$\overline{CF}$	from left nose to the outer edge of left eye
$\overline{EK}$	Two eye center to mouth distance
$\overline{EJ}$	Right from the mouth to the two eye center
$\overline{DJ}$	from mouth to outer edge of right eye
$\overline{EI}$	Left mouth to the center of two eyes
$\overline{CI}$	From outer edge of left eye to left mouth

In Hbase, the design of row key is very important. We set the subcriber id and image number id as the row key id. In order to support 10 billion peoples in the wold, we use 10 byte as the subscriber id, and 2 byte as the image number id. So the row key of Hbase is 12 byte. In order to identify the subscriber's identity, we also need to store the personal info of every subscriber. The personal info table are used to store the subscriber basic info. The image info table are used to store the images of every subscriber. The structure design of Hbase looks as table 2 and table 3.

Now we could store the feature of images into the Hbase, so we need to change the implementation

of the MapReduce. The change point is : the input of Mapper are from Hbase and the output of reducer are to Hbase. The only change in the code is: the input format should be the: TableInPutFormt and the output format should be the: TableOutPutFormat. TableInputFormat use Hbase as the input of Mapper, and TableOutPutFormat use enable reduce function write data into Hbase.

By calculate the feature value and stored in Hbase previously, we can only compare every value and get conclusion in the Mapper function, the identify speed is increased hugely.

4. The Problems and Performance

4.1 The problems

There are some problems about this system and list it below: (a) due to 3D face recognition have too many features, we hard to decide which is the exactly result. (b) If the subscribers are too big, the system can't make sure to obtain the accuracy result in real time process. For example, there are 10 billion peoples and have 20 pictures of each people then the system had totally more than 200 billion pictures. It is too big and hard to process in real time.

4.2 The performance

The performance of the big data system is based on the parameters and structures of the process. There are some methods to improve the performance and list it in the following:

(a). We can adjust the numbers of the Map and Reduce to improve performance.

(b). Set a suitable path of the NameNode and NameNode Federal technique to solve the NameNode breakdown problem.

(c). Design a suitable Hbase to raise the performance of the system inquiry by means of the problem analysis.

(d). According to the features of the face, Adjust procedure design of the Partitioner, grouping, sort and combiner in the Shuffle of the MapReduce process to raise the system performance.

5. Conclusion

In this article, we use the distribution and parallel techniques of the Hadoop to raise the computation speed for the massive 3D face recognition. From the simulation results, it is demonstrated that the computation speed increases 3.5 times under the simulation condition: four computers, a cluster, and one hundred subscribers. When the subscriber increases, the system only extends the cluster to achieve the high performance face recognition at big subscriber.

Acknowledgements

This work was supported by the Youth Research Fund of Xi'an University of Posts & Telecommunications under Grant Nos:1100405.

References

[1]. G. Yang and T. S. Huang, "Human face Detection in Complex Background," Pattern Recognition, vol. 27, no.1, pp53-63, 1994.

[2]. T. K. Leung, M. C. Burl, and P. Perona, "Finding Face in Cluttered Scenes Using Random Label Graph Matching," Proc. Fifth IEEE int'l Conf. Computer Vision, pp. 637-644, 1995.

[3]. M. H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting Faces in Images: A Survey," IEEE Trans. on Pattern Anal. Mach. Intell., Vol. 24, pp. 34-58, 2002

[4]. Zhang Xiaohua, Shan Shi guang, Cao Bo Gao etc." CAS- PEAL：A Large-Scale Chi nese face database and Some Pri mary Eval uations." JOURNAL OF COMPUTER- AI DED DESI GN SCOMPUTER GRAPHICS,vol.17,no.1,pp9-17,2005.

[5]. Chellappa R，Wilson C L，Sirohey S. Human and machi nerecognition of faces ：A survey. Proceedings of the IEEE，1995,83(5)：704 ~741

[6]. Philli ps P J ，Grot her P J ，Micheals RJ ，et al. Face recognition vendor test 2002 ：Evaluation Report〔OL〕. http ：／／WWW.frvt .org ，2003

[7]. Hadoop. Hadoop Official website.[EB/OL]. http://hadoop. apache.org. 2015-03-22/2016-04-06