ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(17), DOI: 10.17485/ijst/2016/v9i17/93014, May 2016* ISSN (Online) : 0974-5645

Two Dimensional Medical Images Diagnosis using MapReduce

Jyoti S. Patil* **and G. Pradeepani**

Department of CSE, K L University, Vaddeswaram, Guntur District - 522 501, Andhra Pradesh, India;
jyoti1199@gmail.com, pradeepini_cse@kluniversity.in

Abstract

Objectives: Due to advanced camera capturing techniques used in medical domain efficient management and quick diagnosis of massively generated 2D/3D medical data has become challenging tasks for doctors. **Methods:** In this paper, we propose an idea for analyzing medical images using Hadoop's MapReduce Framework. HDFS is used for storing feature library of existing medical images and parallelism in indexing; matching and retrieval processes are achieved by MapReduce. The Map function is used to match feature vector of the query image with feature vectors present in feature library, while the Reduce function is used to aggregate and sort results from all the mappers. **Findings:** As a result of parallelism Hadoop based medical image retrieval system will take very less time for image retrieval as compared to the traditional image retrieval systems. Existing Content based medical image retrieval system uses only image processing methods, but to handle Large scale image retrieval and indexing processes we are using MapReduce. It results in the elimination of manual processes of diagnosis and leads to automated detection and diagnosis. **Application:** This model will definitely help doctors in real time decision making and understanding of particular stage of disease. Also real time treatment is suggested after identification of a particular type of disease.

Keywords: Bit Bucket Histogram, Content Based Medical Image Retrieval (CBMIR), Hadoop, HDFS, MapReduce, Medical Image Diagnosis, Sobel Edge Detection

1. Introduction

With the tremendous growth of image generation by 2D/3D graphics hardware technologies, improved cameras and moving satellite, real time analysis of images has become major part of modern image processing systems. In the medical imaging field data retrieval is at very high level like similar-type-of- disease or disease-relevant-diagnosis needs to be analyzed at real time for taking decision in Operation Theater or in diagnosis. The task of the image retrieval and analysis framework is to identify important features and provide indexing of Data as well. For handling these challenges of image analysis one of the most popular programming paradigms on large scale data is Map-reduce. This model can be implemented as a series of Map-reduce operation each consisting of Map Phase and Reduce Phase to process large number of medical data[1].

The Content-Based Medical Image Retrieval (CBMIR)[2] has the advantages of fast retrieval speed and high precision. In CBMIR technology visual features of images are extracted and compared it for image retrieval. It calculates distance between the features of query image and contents of feature library. When the features in the library are huge then efficiency of single-node retrieval from the traditional method is difficult to meet real time requirements of the images, which results in a poor stability and extensibility.

In the traditional text based image retrieval key words are used to retrieve the required area of images[3]. But main disadvantage of this method is that each image should be marked manually which will increase the work load.

* *Author for correspondence*

Another limitation of this method is that image cannot be completely described by using only words. Also each person may understand same image in different way, so the understandings of images are varies from person to person.

In this paper, we are proposing a medical image retrieval system in which analysis is based on MapReduce to achieve parallelism and subsequently reduce retrieval time. Hadoop[4] is an open source framework which is used to handle big data. Hadoop Distributed File System (HDFS)[4] for parallel distributed file storage system and MapReduce[4] for parallel processing are used in this application. Hadoop technology provides scalability, fault tolerance, high availability and parallelism[4]. With such features we can say that Hadoop is best suited to handle the problem of matching and retrieval of large medical image data. Medical Image Retrieval using MapReduce[1] works on the basis of medical term known as Evidence-Based-Practice (EBP)[5], in which doctors treat patients on the basis of treatment given to similar cases in past. Doctors need to refer the records of previous cases to diagnose diseases more efficiently, for this doctors need accurate matching data from the database of previous cases, and that too in less amount of time. Our application has provided best tool to doctors for analyzing diseases.

2. Proposed System

We are proposing a Hadoop MapReduce based medical image retrieval and analysis system to manage large amount of medical image data[1]. We are considering prototype system with a master node and three slave nodes. When user gives query image to the system, it will go to master node and master node will generate a MapReduce task to construct feature vector of query image. After this a new MapReduce task will be generated by master node to match feature vector of query image with feature vectors in feature library stored at HDFS. The result of previous MapReduce task will be given to the third MapReduce task which will retrieve required images[1].

2.1 Proposed Architecture

Proposed system consists of a Hadoop cluster containing one master node and more than one slave nodes (Multinode Hadoop cluster). As described in Figure 1 of Proposed Architecture when user will upload a query image to the system, Hadoop's master node will accept that image and processing will be done. After processing this image will be stored in the HDFS in the form of feature vectors. JobTracker[4] on master node will create a MapReduce job for doing the above task (i.e. extracting features from query image and storing it into HDFS). JobTracker will also divide the MapReduce job into many tasks, and then these tasks are allocated among slave nodes (Map phase).

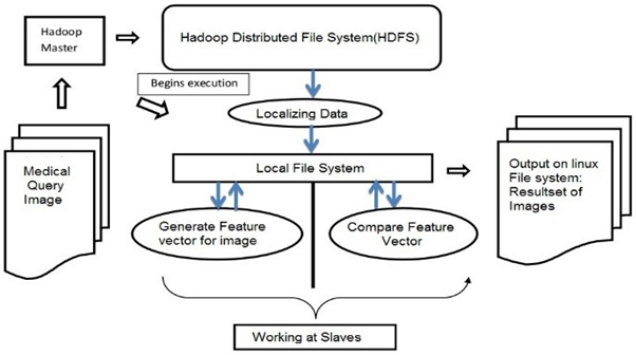

Figure 1. Proposed Architecture.

Task tracker[4] on slave nodes will perform the actual work. JobTracker creates a MapReduce job and divides this job into tasks. These tasks are then given to the TaskTracker. TaskTracker will perform the mapping of feature vector of query image with feature vectors in feature library and the result is written in a temporary directory, reducer combines results from all the mappers and generate an aggregated final result which contains distance of each vector in library with respect to feature vector of query image. Distances are in ascending order i.e. smallest will come first.

2.2 Design Steps

Figure 2 shows all the design steps used in this model which are elaborated further in this section.

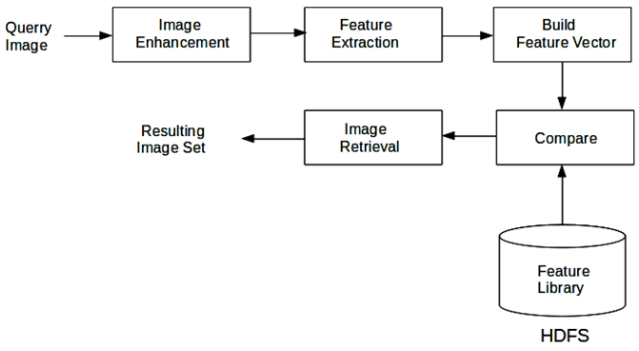

Figure 2. Design Steps.

2.2.1 Step I: Image Enhancement

Query image uploaded by user may not be suitable for the system, as it may contain noise or it can be dimensionally incompatible with the system. So in this case image enhancement is necessary. Image enhancement[6] is the process of adjusting features of the image in order to make it suitable for the system. There are various methods for image enhancement such as histogram equalization, contrast enhancement, image resize, and gray level slicing, negative of the image etc. Figure 3 depicts an original image and resulting image after adding enhancement method.

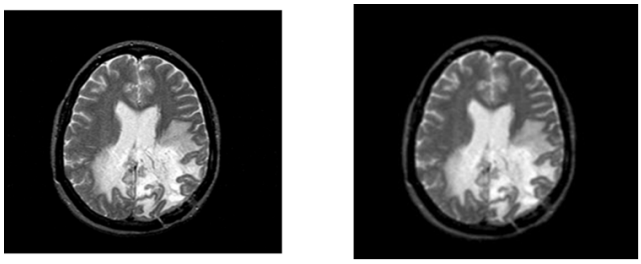

Figure 3. Effect of enhancement.

2.2.2 Step 2: Feature Extraction

Feature extraction[7] is a process of extracting relevant feature from the set of features of an image. The extracted features are expected to contain relevant data for further processing. Since it is very complex to process the entire image (such as a comparison), it is feasible to process selected features of an image.

2.2.2.1 Sobel Edge Detector Algorithm

A spatial edge detection filter that detects edges by finding the gradient of an image. Which means the dramatic difference between the values of pixels is calculated. Two Template Matrices are used in Applying Sobel Algorithm[8]. Figure 4 shows the results of sample image after applying the Edge Detection algorithm.

$$\text{FOR X - AXIS(Gx)} \begin{matrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{matrix}$$

$$\text{FOR Y - AXIS(Gy)} \begin{matrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}$$

Algorithm steps.
Input: A Sample Image.
Output: Detected Edges.

- The input image is first converted to gray scaled image.
- Traverse through entire image pixel by pixel.
- For each and every pixel in the image we will take a window of 3×3 pixels and multiply with the given template for matrix.
- Calculate the G using formula (7).

$$|G| = \sqrt{G_X^2 + G_Y^2}$$

- Apply these given templates to filter window.

$$\begin{matrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & a_9 \end{matrix}$$

From above matrix values of a1... a9 are grey levels of each pixel in the filter window

$$\text{Value of X} = -1\times a_1 + 1\times a_3 - 2\times a_4 + 2\times a_6 - 1\times a_7 + 1\times a_9$$

$$\text{Value of Y} = 1\times a_1 + 2\times a_2 + 1\times a_3 - 1\times a_7 - 2\times a_8 - 1\times a_9$$

$$\text{Sobel Gradient} = \sqrt{(X * X + Y * Y)}$$

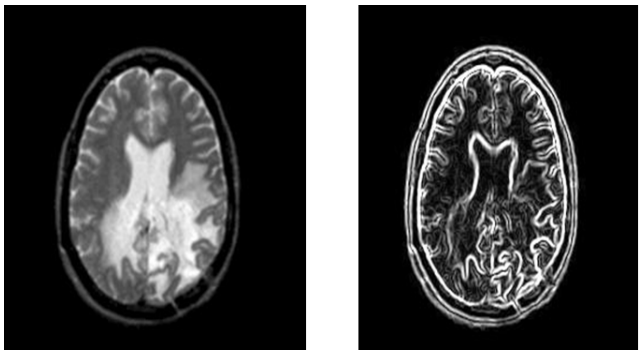

Figure 4. Edge Detection algorithm.

2.2.3 Step 3: Build Feature Vector

Matrix containing information about important characteristics of an image is called as a feature vector. In order to compare two images, actually we compare their feature vectors. Feature vector is built using features

extracted during the feature extraction phase. This feature vector can be used for indexing, matching or retrieving required images. This feature vector is a unique identity for an image, because feature vector generated for each image is different. Methods such as Histogram[9], Bin bucket Algorithm is used for building feature vector. This algorithm is used to generate feature vector of an image. For color images, color value can take values between 0-255. So we get total 256 different color values. In order to generate feature vector we use bucketing technique. In this technique we divide color values of pixels among the buckets. Let's say, for 8 buckets we module color value of 8 and we assign that value to the bucket which having value equal to reminder modulo operation.

i.e. color value of particular pixel is 223

223 % 8 = 7 so we assign 233 to bucket no. 7

Generated feature vector for single image should look as follows

[654, 54, 354, 354, 87,735,354,325]

2.2.4 Step 4: Compare

In this phase feature vector of query image is compared with existing feature vectors in feature library. Feature library is a set of feature vector of existing images, and this library is stored in HDFS. While comparing two feature vectors, we calculate the distance between two features. We have used methods such as Euclidean distance and the Hamming distance for calculating the matching distance. According to the calculated distance, existing images are sorted in ascending order (Ranking). Global features and Local features are also matched using histogram equalization techniques.

2.2.5 Step 5: Image Retrieval

According to the ranking created in comparison phase, top ten images (as per requirement) are considered as the best match for the query image because they are having the minimum distance with query image. Such images are retrieved and this image set is given to the user. Based on threshold severity the stage of disease can be easily diagnosed[3].

3. Results and Discussions

For adding each new image into HDFS first text files according to clinical pathology is created ,then the feature vector of a sample is generated for each medical image one for each category and stored in respective text files. Euclidean distance between the new image and the first feature vector stored in text files is calculated. Sample brain tumor images are loaded into HDFS. Figure 5 shows the results of vector generations.

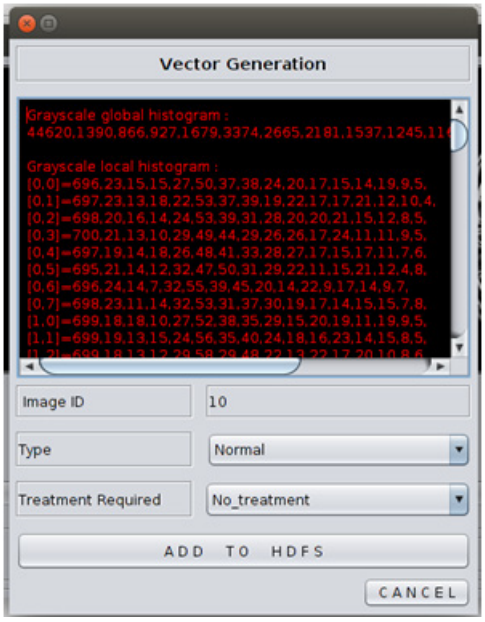

Figure 5. Feature vector generation in hadoop.

Each image is enhanced using blurring techniques, its histogram is calculated also edge detection is done by SOBEL algorithm and it gets indexed in feature library stored in HDFS. Figure 6 gives results of edge detection.

Figure 6. After Applying SOBEL Edge Detection Algorithm.

Query image i.e. tumor image[10] of a patient is inputted to the system and relevant matching images are retrieved from the system based on Euclidean distance. MapReduce Job is fired in the background to match and

retrieve relevant images. Figure 7 shows the final output generated from the model and accurate diagnosis for particular query image.

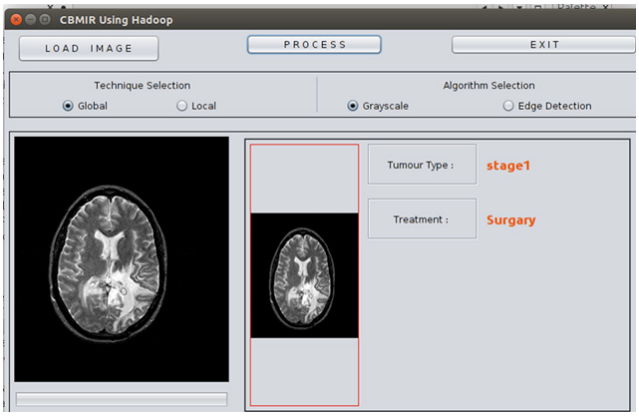

Figure 7. Final Result Obtained and Diagnosis.

Each query image is matched with each indexed file. Based on the matched threshold value of image stage of particular disease can also be analyzed. According to previous cases, what type of treatment can be given is also suggested automatically. For example, if maximum number of images are indicating brain tumor which are having stage1 and suggested treatment as surgery, then it will show final diagnosis as:

- Stage1 detected as brain tumor.
- Suggested Treatment: Surgery.

And final output image should be the image with minimum Euclidean distance.

4. Conclusion

A cluster of one master and three slaves is created; a client machine is used by client to upload query images to the master node. When client uploads the image through client machine, master node accepts it and creates a MapReduce job. Job is divided into tasks and given to the slaves to execute. Because of parallel execution, processing becomes faster and throughput of system performance increases by reducing image retrieval time.

Currently system has given 29 millisecond response times for replication factor 2 which is calculated for 10 images. Also 27 milliseconds for replication factor 3 for same set of images on HDFS. We can achieve greater scalability by deploying it as IaaS or SaaS kind of applications on the available cloud systems. Larger number of medical images can be analyzed and diseases are diagnosed automatically leads to a greater reduction in manual disease diagnosis.

5. References

1. Patil J, Mane S. 3-D Image analysis using MapReduce. Proceedings of IEEE conference on Pervasive Computing (ICPC); India. 2015. p. 521–5.
2. Willy PM, Kufer KH. Content Based Medical Image Retrieval (CBMIR): on intelligent retrieval system for handling multiple organs of interest. Proceedings of 17th IEEE Symposium on Computer Based Medical System; 2004. p. 103–8.
3. Qing-An YAO, Zheng H. Medical images retrieval system based on Hadoop. Journal of Multimedia. 2014 Feb; 9(2):216–22.
4. Tom white, Hadoop: The Definitive Guide, 3rd ed. O'Reilly Publication; 2006.
5. Introduction to Evidence-Based-Practice (EBP). 04 Mar 2016. Available from: http://www.guides.mclibrary.duke.edu/c.php
6. Maini R, Aggarwal H. A comprehensive review of image enhancement techniques. Journal of Computing. 2010 Mar; 2(3):1–6.
7. Deepa A, Sasipraba T. Challenging aspects for facial feature extraction and age estimation. Indian Journal of Science and Technology. 2016; 9(4):1–7.
8. Vincet O, Folorunso O. A descriptive algorithm for Sobel image edge detection. Proceedings of Informing Science and IT Education Conference (InSITE); 2009. p. 1–11.
9. Savvas Chatzichristofis A, Yiannis Boutalis S. FCTH: Fuzzy color and texture histogram: A low level feature for accurate image retrieval. Proceedings of the 9th International Workshop on Image Analysis for Multimedia Interactive Service; 2008. p. 191–6.
10. Baraiya N, Modi H. Comparative study of different methods for brain tumor extraction from MRI images using Image Processing. Indian Journal of Science and Technology. 2016; 9(4):1–6.