Edge and Texture Feature Extraction Using Canny and Haralick Textures on SPARK Cluster

D. Sudheer, R. SethuMadhavi and P. Balakrishnan

Abstract Image retrieval is the most significant technology forever. Computer vision is improving its trends and methodologies to perform like a human. System can identify any object without a human help by a simple query. To train the system, we need precise algorithms. Content-based image retrieval (CBIR) is the recent emerging trend in computer vision to retrieve relevant images from huge amount of data. This paper used a distributed and parallel processing paradigm to accelerate the retrieval process. SPARK stream processing environment is used on the top of the Hadoop Distributed File System (HDFS). To extract visual content of the images, edge and texture features are used. The distance between query image and database is measured with Mahalanobis distance metric. The performance of the retrieval system has been compared with other distributed image retrieval systems. The proposed methodology using SPARK environment outperforms the existing systems.

Keywords SPARK · HDFS · Map reduce · CBIR · Feature extraction
Texture analysis

1 Introduction

CBIR is the advanced trend in the computer vision. CBIR is not only used to retrieve the relevant images from the large-scale databases, but also to recognize objects in medical picture archiving communication systems (PACS), biomedical applications, satellite image retrieval systems, and many other applications. Currently, CBIR has achieved sophisticated accuracy using feature extraction systems based on color, shape, texture. But CBIR is lagged in efficient storage and processing paradigm due to increase in usage of smart devices. Since CBIR systems have to adapt to the commodity hardware [1]. Processing multimedia data is tedious process in terms of speed and accuracy. This paper used open-source distributed and parallel frame-

D. Sudheer (✉) · R. SethuMadhavi · P. Balakrishnan
SCOPE, VIT University, Vellore, Tamil Nadu, India
e-mail: 2498sudheer@gmail.com

A. J. Kulkarni et al. (eds.), *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, Advances in Intelligent Systems and Computing 828, https://doi.org/10.1007/978-981-13-1610-4_56

work Hadoop released by Apache foundation. To achieve more reliable and fastest response, SPARK is used on the top of the Hadoop. To achieve accurate retrieval result, Canny edge algorithm and Haralick textures are used in spatial domain.

This paper is organized as follows: Sect. 2 describes Big Data processing framework, Section 3 describes the related work done so far in the application, Sect. 4 describes the proposed methodology, Sect. 5 describes the experimental results and discussion, and Sect. 6 describes the conclusion of the work.

2 Big Data Processing Framework

The multimedia data is growing through social media (Flickr—75 million public images per day, Instagram—60 million images per day, Facebook—136,000 images per min, Google—57,988 query per second). Storage of these huge amounts of data very complex task using traditional systems. To overcome this issue, Google has published a white paper on Google file system. This is a distributed block-level file system. The data will be decentralized to several nodes. Each node will have numerous blocks [2]. Each block size can change by user requirement. Later, Google also published another white paper on MapReduce. MapReduce is parallel processing paradigm. Here, data locality is introduced. Instead of transforming the data through the network, code will be sent to the distributed blocks that exist in the several nodes. These two approaches were combined together and free leased as an open-source framework and named as Apache Hadoop. Hadoop is not suitable for a large amount of small files. Hadoop is designed as block-oriented file system, where default block size is 128 MB [3]. When the file size is less than block size, then it will create a separate block to that file and use the remaining space to other block. MapReduce will divide the process into two tasks as Map and Reduce, in which Map tasks are created based on the number of blocks that contain our data. So if system contains more number of blocks, then more number of Map tasks should be initiated by the Hadoop Master. This process will create a bottle neck problem to the Hadoop cluster. Hadoop introduced sequence file format to process all small files as a single large file [4]. Data sharing is not much faster in Hadoop due to replication and serialization process [5].

To simplify this problem, Hadoop introduced SPARK to processing. It is designed for fast cluster computations. SPARK will execute the scripts on the top of the Hadoop, and it extends the MapReduce mode to streaming process [6]. SPARK is also introduced by Apache foundation. SPARK was developed in 2009. The main feature of SPARK is in-memory cluster computation. SPARK not only supports MapReduce, but it also supports SQL, machine learning, etc. SPARK will create the data as Resilient Distributed Dataset. The RDDs will be divided into logical partitions which are able to process in several nodes. RDD is a read-only record; it can be created through deterministic operations. RDDs are fault-tolerant. It can reuse the intermediate results to accelerate the data sharing process.

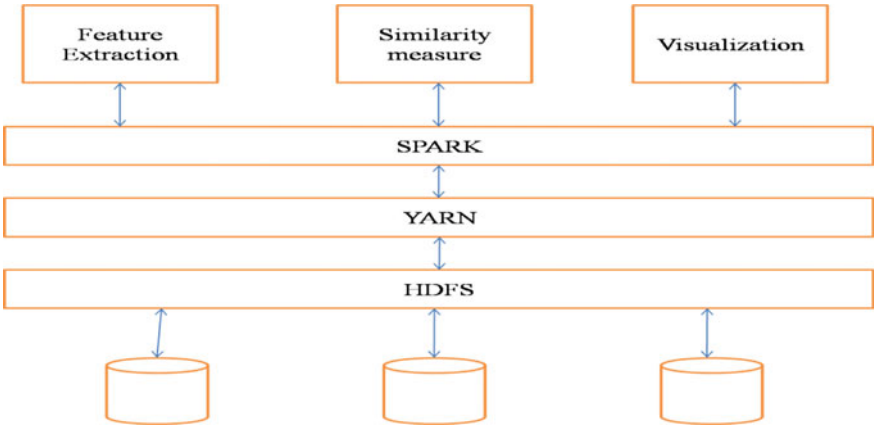

Fig. 1 Architecture of the proposed system

Image database was collected from the signal processing laboratory Web server. Ultrasound images are collected from the online Web server and stored in the HDFS [7, 8]. Feature extraction and similarity measure were done using Python and OpenCV library. The experiment was run on the SPARK integrated with Hadoop-2.6.0 single-node cluster environment. The proposed architecture is shown in below figure.

2.1 Architecture

See Fig. 1.

3 Related Work

CBIR has evolved with robust feature extraction algorithms so far. Color histogram-based features are primary and easy to represent any image. The color maps can be represented as binary, or gray scale, RGB, HSV, etc.; the extraction of color histogram is done based on the intensity levels of pixels in an image. The distribution of intensities can be represented as mathematically as follows:

$$h_c = \frac{1}{MxN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \delta\left(f_{ij} - C\right), \quad \forall c \in C$$

$$\text{where, } \delta(x) = \left\{ \begin{array}{l} 1, \text{if}(x = 0) \\ 0, \text{if}(x != 0) \end{array} \right\}$$

The color histogram features were computed, and the similarity between vectors is computed using distance metrics [9]. Color features alone cannot obtain the exact required image due to high variations in the intensity levels. So texture-based features are introduced in 1973 by Haralick. The gray-level co-occurrence matrix (GLCM) will be computed from the image, and the statistical features of the GLCM will be extracted as texture of the image [10]. The important statistical features that can be extracted from GLCM were shown below:

$$\text{Energy} = \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} M^2(i, j)}$$

In the above formula, i, j denotes the spatial position of an image. Calculating the extent of pixel pair repetition is called as energy. This energy will become large if the image pixels are same.

$$\text{Entropy} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} M(i, j)(-\ln(M(i, j)))$$

Entropy is used to categorize the texture of an image. Randomness of the input image can be known using entropy. When the entire co-occurrence matrix values are same, then the value of entropy will also reaches utmost.

$$\text{Contrast} = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i - j)^2 M(i, j)$$

The intensity of a pixel and its neighbor of an image is measured using contrast. A contrast is nothing but the divergence in the color and the brightness of object which is explained by visual perception [11].

Many applications have proved experimentally that hybrid features will give more accuracy. However, the accuracy might have satisfied in traditional approach, but when dealing with real-time and large-scale datasets, it is complex to process with existing CBIR [9]. Distributed and parallelized CBIR systems have been developed and proved the response time is very less when compared to traditional systems [12].

4 Methodology

The proposed methodology used edge and texture features. The workflow of proposed methodology is shown in Fig. 2. Canny algorithm is used to extract the edge features. Canny uses the first-order derivatives of the image. To remove the noise, Canny applies the Gaussian filter first. The gradients are extracted by applying first-order derivatives [13]. Intensity thresholding is used to remove the false edges.

4.1 Canny Edge Descriptor Algorithm

Step 1. To remove the noise and to make an image smooth, Gaussian filter is to be applied.
Step 2. Image's intensity gradients are to be found
Step 3. Non-maximum suppression is to be applied to get exonerate from the false response of edge detection.
Step 4. To find the potential edges, apply double threshold.
Step 5. Those edges that are not strongly connected to the other edges are to be detected.

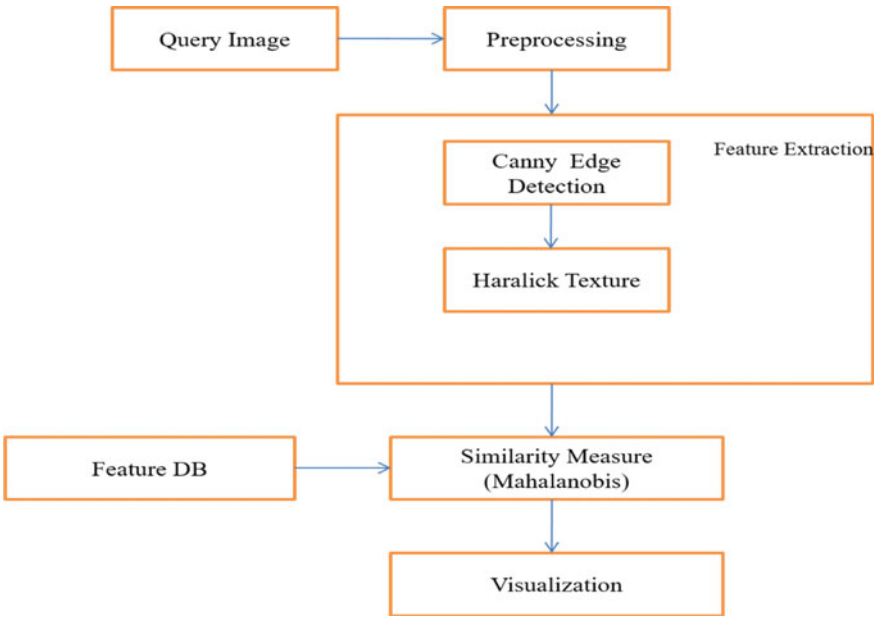

Fig. 2 Workflow of the system

4.2 Workflow

See Fig. 2.

4.3 Haralick Texture Features

To create the "texture" data, Haralick analysis is the one among the various techniques. Haralick descriptors are used to extract the texture features; those can be evaluated from the gray-level co-occurrence matrices (GLCMs). GLCM is a 2D histogram, which apprehends the co-occurrences of the two-pixel intensities, each other at certain offset in a region from where texture is calculated. Features like energy, homogeneity, contrast, and dissimilarity are extracted from GLCM [10]. The GLCM can construct in three directions (horizontal, vertical, and diagonal).

5 Results and Discussion

Figures 3 and 4 show the retrieval results of the proposed system. Figure 1 retrieved MedPix dataset; it contains scan images of various body parts. Figure 4 contains SPLab dataset; it contains ultrasound images. The performance of the retrieval system is mathematically defined as [14];

$$\text{Precision} = \frac{\text{Number of relevant images retrieved}}{\text{Total Number of Retrieved Images}}$$

The proposed retrieval system obtained 75% precision for the input datasets. Another performance measure is considered as speed. The speed of the system is compared between single-node and multimode cluster by varying multiple dataset sizes. The performance graph was shown below.

Figures 5 and 6 show the relationship between time and data size on single-node and multi-node cluster, respectively. The X-axis denotes the time taken to execute the task in milliseconds. The Y-axis denotes number of images in the dataset.

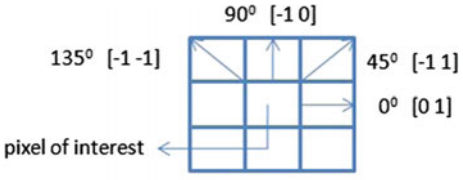

Fig. 3 GLCM directions

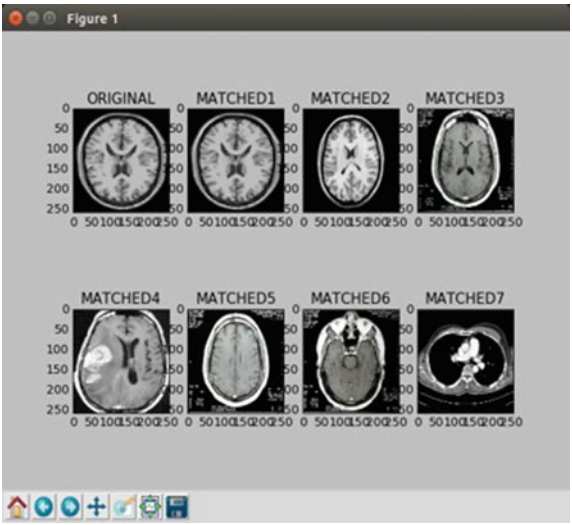

Fig. 4 Result for dataset1

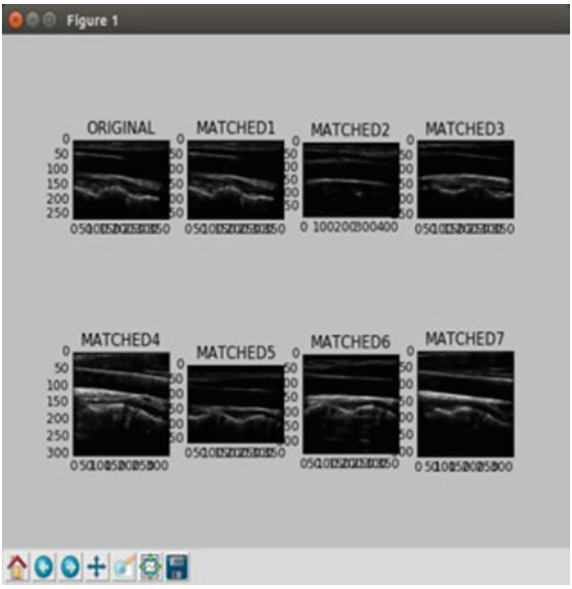

Fig. 5 Result for dataset2

The experimental results had proved that our proposed methodology obtained better precision and faster than existing systems (Fig. 7).

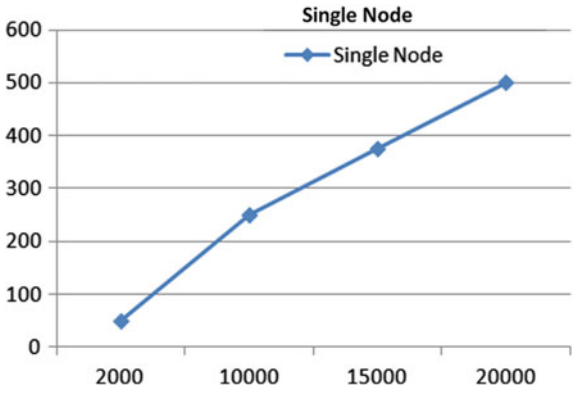

Fig. 6 Time graph for single-node cluster

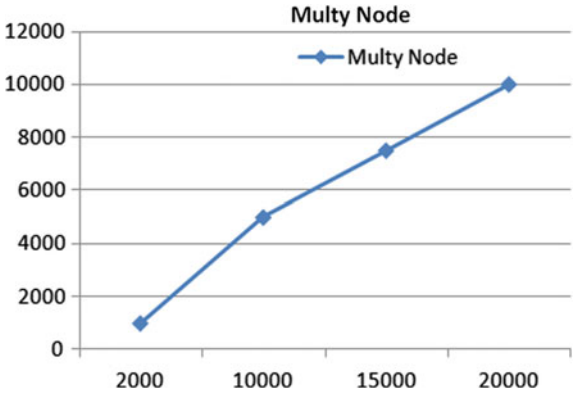

Fig. 7 Time graph for multi-node cluster

6 Conclusion

Image retrieval is the growing trend. Many technologies are emerging largely nowadays for this sake. Features like mean, standard deviation, and variance are extracted from LBPs. Canny edge detection is used for detecting edges. Gray-level co-occurrence matrices (GLCMs) are used to extract texture features. Mahalanobis distance is used to measure the similarity between query images and database. SPARK performs well for the entire system. Thus, we can able to get the related images from the database to the query image successfully.

References

1. Palle A, Kulkarni RB (2016) Classification of medical MRI brain images based on Hadoop. In: Proceedings of the second international conference on information and communication technology for competitive strategies—ICTCS '16, 1–4. https://doi.org/10.1145/2905055.2905154

2. Alham NK, Li M, Liu Y, Hammoud S (2011) A MapReduce-based distributed SVM algorithm for automatic image annotation. Comput Math Appl 62(7):2801–2811. https://doi.org/10.1016/j.camwa.2011.07.046
3. Sudheer D, Lakshmi AR (2015) Performance evaluation of Hadoop distributed file system. Pseudo Distrib Mode Fully Distrib Mode (9):81–86
4. Vesset D, Olofson CW, Nadkarni A, Zaidi A, Mcdonough B, Schubmehl D, … Carnelley P (2015) IDC FutureScape: worldwide big data and analytics 2016 predictions. IDC, 1–13
5. Yamamoto M (2012) Parallel image database processing with Mapreduce and performance evaluation in pseudo distributed mode. Int J Electron Commer Stud 3(2):211–228. https://doi.org/10.7903/ijecs.1092
6. Raju USN, George S, Praneeth VS, Deo R, Jain P (2015) Content based image retrieval on Hadoop framework. IEEE Int Congr Big Data 2015:661–664. https://doi.org/10.1109/BigDataCongress.2015.103
7. Hsieh LC, Wu GL, Hsu YM, Hsu W (2014) Online image search result grouping with MapReduce-based image clustering and graph construction for large-scale photos. J Vis Commun Image Represent 25(2):384–395. https://doi.org/10.1016/j.jvcir.2013.12.010
8. Costantini L, Nicolussi R (2015) Performances evaluation of a novel Hadoop and spark based system of image retrieval for huge collections. Adv Multimedia. https://doi.org/10.1155/2015/629783
9. Gu C, Gao Y (2012) A content-based image retrieval system based on Hadoop and Lucene. In: 2012 second international conference on cloud and green computing, pp 684–687. https://doi.org/10.1109/CGC.2012.33
10. Wang X-Y, Yu Y-J, Yang H-Y (2011) An effective image retrieval scheme using color, texture and shape features. Comput Stand Interfaces 33(1):59–68. https://doi.org/10.1016/j.csi.2010.03.004, Arabi PM, Joshi G, Vamsha Deepa N (2016) Performance evaluation of GLCM and pixel intensity matrix for skin texture analysis. Perspect Sci 8:203–206. https://doi.org/10.1016/j.pisc.2016.03.018
11. Yin D, Liu D (2013) Content-based image retrial based on Hadoop. Math Prob Eng. https://doi.org/10.1155/2013/684615
12. Chen M, Mao S, Liu Y (2014) Big data: a survey. Mobile Netw Appl 19(2):171–209. https://doi.org/10.1007/s11036-013-0489-0
13. Moise D, Shestakov D, Gudmundsson G, Amsaleg L (2013) Indexing and searching 100 M images with map-reduce. ICMR 2013:17–24. https://doi.org/10.1145/2461466.2461470
14. Sakr NA, ELdesouky AI, Arafat H (2016) An efficient fast-response content-based image retrieval framework for big data. Comput Electr Eng 54:522–538. https://doi.org/10.1016/j.compeleceng.2016.04.015
15. Bolón-Canedo V, Sánchez-Maroño N, Alonso-Betanzos A (2015) Recent advances and emerging challenges of feature selection in the context of big data. Knowl-Based Syst 86(May):33–45. https://doi.org/10.1016/j.knosys.2015.05.014