OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

A COMPREHENSIVE CONTENT-BASED IMAGE RETRIEVAL SYSTEMS USING HADOOP MAPREDUCE FRAMEWORK

Suresh Kanaparthi[1], USN Raju[2]
Research Scholar, *Department of C S E, National Institute of Technology, Warangal, India.[1]*
Assistant Professor, *Department of C S E, National Institute of Technology, Warangal, India.[2]*
sureshkonline@gmail.com [1], *usn.raju@gmail.com* [2]

Abstract: As we are living in multimedia era, nowadays Internet has been heavily used for viewing, sharing and storing collection of the videos and images. This results in huge amount of data. In user's point of view, different domains are facing many problems with retrieving images that are relevant to their query. To solve these problems different techniques are adopted. Significant work has been done in several issues of performance evaluation such as feature extraction, feature matching, semantic gap reduction and similarity measurements. Based on the requirement of different applications and current technologies, we retrieve, process and store the content based images. But, we are lacking techniques to process, retrieve and store images faster. The Hadoop Map Reduce platform provides a system for large storage and computationally intensive distributed processing when, it is used for Content-based Image Retrieval (CBIR) systems performance can be improved compared to other retrieval systems.

Keywords: Annotation, Content-based Image Retrieval, Feature extraction, Relevance feedback, Similarity measure, Computer Vision, Hadoop HDFS, MapReduce Framework.

I INTRODUCTION

Content-based Image Retrieval is also known as visual image retrieval. It is a technique to search in large scale image databases for a required image specified by the user. Content is considered as image descriptors like shape, color, texture, etc. Since 1970, Image retrieval has been one of the active research areas, with advancements in the communities' database management and computer vision. These communities carried out in different perspectives. The former one uses text-based features whereas the latter one uses visual-based features. In text-based image retrieval, images need to be annotated with text, in order to perform text-based search of images. Several text-based image retrieval methods can be found in [1,2]. Major limitations of text-based approaches include difficulty in annotating large number of images stored (also expensive), subjectivities of human perception i.e., different people may not perceive uniquely for same image. These limitations may lead to unrecoverable mismatches in further retrieval process.

As increased availability of Internet and other digital image technologies in 1990's, large volumes of data have been generated in fields like medical, education, industries etc. Due to this, text-based image retrieval become inefficient. In order to overcome this difficulty, content-based image retrieval (CBIR) was introduced. Unlike text-based retrieval, in CBIR, images would be indexed with their content features such as shape, texture, etc. Based on this innovation large number of applications were developed to serve in the sectors like hospitals, crime investigation, companies etc. Comprehensive surveys of these techniques and systems can be found in [3-5]. The advances in this research direction are mainly contributed by the computer vision community. Many special issues of leading journals have been dedicated to this topic are [6-10]. Usually, CBIR system will be divided into different phases: Feature extraction, feature matching and Semantic Image Retrieval [4]. A Standard content-based image retrieval can be shown as in Fig.1.

The retrieval strategy begins, once user gives a query to the system. The query can be framed either in the form of text (as string) or in the form of image as in example. CBIR system strategy obtains fundamental features like color, texture and shape for query image and then it matches these features with all the images stored in the database after deriving fundamental features for them as well [11]. While matching, the system assigns some index value to grade the images/objects based on to their matching level. Then display the results with respect to grading value [12,13].

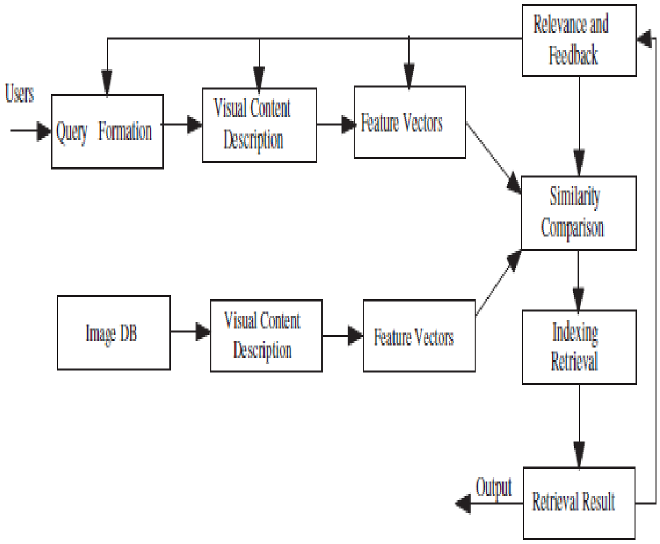

Figure 1 Diagram for Content-based Image Retrieval System

In the retrieval process, most of the CBIR Systems to support the Hadoop Software framework for feature extraction, feature matching, semantic gap, and similarity measurement. But, these CBIR Systems are time consuming, laborious and expensive. Nowadays, many researchers' effort to solve this problem by applying the distributed computing methods. The approaches include different variety of areas like, cluster computing in order to decrease the computational time [14-16] offered a cluster platform which was compatible with running other retrieval approaches practiced in CBIR Systems.

A parallel technique to execute similarity comparison and feature extraction of visual features rooted in cluster architecture [14]. The experiments indicated that a parallel computing technique can be used with the intention of considerably improving the performance of a retrieval system. To define a distributed algorithm and program it with cross-platform ability is a crucial mission [17]. In this study, we proposed a parallel processing technique medium of Hadoop MapReduce framework [18-20] for content-based image retrieval system instances. The two contributions discussed in this study are index creation process and similarity measurement process by MapReduce.

II OVERVIEW OF CBIR SYSTEMS

Many industrial and academic organizations have developed content based image retrieval systems which use such general features [21,22]. Most of the Image Retrieval Systems support one or more of the following options [23], such as random browsing, search by sketch, search by example, search by text (Keyword and/or speech), and navigation with customized image categories. Here, we will select some representative systems and highlight their distinct characteristics.

A. QBIC (Query by Image Content)

QBIC System introduced by IBM. It is actually extracts image features like shape, texture and color etc. User can give queries by drawing some picture on the screen. QBIC search is very fast, easily implemented and invariant to small changes in camera angle (sometimes large). Sensitive to illumination changes and different levels of gamma corrections, doesn't account for location of color [4,24]. This is first commercial content-based image retrieval system.

B. VIR Image Engine

Virage Inc. has developed VIR (Visual Image Retrieval) Image engine to design user interface sufficient GUI tools. A basic concept is that of a primitive, which denotes a features type, computation and matching distance. The primitives used by VIR are local and global color values, texture and shape. The Virage engine operates at the image object level of the Virage model. Image analysis, comparison, and management are the three major functional units of the VIR Engine. Application developer invokes these in order to perform image insertion, image query and image re-query [4,24]. VIR takes either sample image or sketch or selected color as an input.

C. Retrieval Ware

Retrieval ware has been developed by Excalibur Technologies Corp. The main focus in Retrieval Ware was on neural nets to retrieve an image. In recent days, search engines using brightness, color layout and aspect ratios as query features [4,24].

D. Netra

Netra designed and developed by Electrical and Computer Engineering Dept. of University of California. Netra, a prototype image retrieval system that uses color, texture, shape and spatial location information is segmented image regions to search and retrieve similar regions from the database [4,24].

E. VisualSEEK

VisualSEEK developed by Advanced Television and Image Lab, Columbia University, New York. VisualSEEK is a hybrid system in that it integrates features-based indexing with spatial Query method. Based on color region, spatial location and shape, visualSEEK retrieves images. In visualSEEK, several feedback facilities are available to refine the search process [4,24].

F. WebSEEK

WebSEEK is a web oriented search engine. It consists of three main modules, i.e. image or video collecting module, subject classification and indexing module, and search, browse, and retrieval module [4,24].

G. Chabot

Chabot is evolution of dept. of Computer Science, University of California. The main idea in Chabot is using both existing texture information and additional texture information for image such as perspective of the picture, location, etc. This system was applied on 11643 images of natural resources in California [24].

H. Photobook

Photobook is one of the developments of Massachusetts Institute of Technology Media lab, Cambridge. For various image contents, Photobook implements one of three different approaches to construct image represents, namely: 2D shapes, faces and texture images. The system can be applied in various applications as in [4,24].

I. FIDS (Flexible Image Database System)

FIDS has Developed by dept. of Computer Science and Engineering University of Washington, USA. FIDS considers the features like color histogram, coefficients of Haar Wavelet decomposition and local binary pattern histogram. After selecting the image, user can choose feature distance measures to combine them into an overall distance measure [24].

J. JACOB (Just A Content Based query system for video databases)

JACOB has developed by Computer Science Artificial Lab, University of Palermo, Italy. JACOB is used to retrieve video clips by using the combination of color, motion and texture features. Moreover, color percentage method and feature of dots applied by Berkeley University in their Digital library project. The user can retrieve videos by specifying various color percentages and can specify dots colors and sizes that are required in the resultant image [24].

III CBIR ATTRIBUTES

CBIR is performed in the following two phases: *Preprocessing*: The image is preprocessed by performing filtering normalization, segmentation and identification. This preprocessing will be used to identify significant objects and regions.

Feature extraction: General features of the image like color, texture and shape will describe the content of the image. CBIR integrates high-tech elements like pattern recognition, human perception information sciences, multimedia, signal and image processing.

Initially CBIR systems were used with manual textual annotation in order to represent image content. The drawback of these systems was it can't be applicable for large volumes of data. In CBIR, feature vectors generated in order to index the images and will be stored in the feature database as indexes. Feature vectors are compared with the query image feature vector to retrieve image [25]. Let $F(w, x)$: $w = 1, 2, 3. . . W$, $x= 1, 2, 3 . . . X$ be a 2D image pixel array. For color image $F(w, x)$ denotes the color value at pixel (w, x) i.e $F(w, x) = \{F_R(w, x), F_G(w, x), F_B(w, x)\}$. For black and white images $F(w, x)$ denotes the gray scale intensity value of pixel (w, x). The problem of retrieval is following for a query image Q, we find image T from the image database, such that distance between corresponding feature vectors is less than specified threshold, i.e. D (*Feature (Q), Feature (T)* < t).

A. Feature Extraction

The basis for Content-based Image Retrieval is feature extraction. In general, features can be classified as text-based and visual features. Text-based features include key words and annotations whereas, visual features include color, texture, shape, fingerprint and human faces. Although, most of the literature is on text-based feature extraction (as in DBMS), we are now focusing on visual feature extraction. Coming to visual features, these can be further divided into general features such as color, texture and shape and domain-specific features such as fingerprint and human faces. Out of these two, the former features used in content-based image retrieval and the later can be utilized in pattern recognition. Depending upon pixel values, general features of the given image and other stored images in the databases will be extracted. This extraction can be done on whole image or certain segments (or regions), acquired from image objects segmentation. Further, these features will be stored by CBIR System into separate database called feature database (or Image signature) [12]. Eakins mentioned three levels of queries in CBIR [21]. The similarity feature, which is used for comparing the various features is the Euclidean Distance (*ED*). The formula of Euclidean Distance is,

$$ED = \sum_{i=1}^{n} | x_i - y_i |$$

the minimum distance value signifies an exact match with the query [26]. Euclidean Distance is not always the best metric. In general, image features used for content based image retrieval were i) Color ii) Texture iii) Shape.

A.1) Color

In image retrieval, most widely used visual feature is color. Some studies on color spaces and color perception can be referred in [27-29]. Different color spaces are used for various applications. These color spaces can be broadly categorized into uniform color spaces and non-uniform color spaces. In uniform color spaces, the Euclidean Distance between two points in the color space is calculated from a color difference perceived by human observer. Some of the uniform color spaces are MTM, CIE L* a* b* and CIE L* u* v*. Some of the non-uniform color spaces are HSV, HCV, HSL, YCrCb and Hue-min-max-difference (HMMD). Although we have several non-uniform color spaces, HSV is most frequently used color space to represent color, due to the property that it is perceptual to user. In image retrieval system, query image color histogram is obtained and then mapped with other images color histograms which are stored in the database. Most similar images with query image will be retrieved. The amount of pixels of each color embedded in the color can be represented by color histogram. User can specify input as either an amount of color pixels like blue 40% or by providing sample image from which calculation of color histogram can be done. In 1991, a matching technique called histogram intersection was proposed by Swain and Ballard. To represent color histogram as color sets, Smith and Chang [30,31] implemented conversion from RGB color space into a non-uniform color space, namely HSV and then quantized the converted color space into N bins [4,32]. Furthermore, to reduce quantization effects color moments technique was proposed by Stricker and Orengo as an alternative to color histogram [33,34]. In this, moments can be used to characterize color distribution.

A.2) Shape

In the Shape based image retrieval, the main focus is on measuring similarity in shapes based upon their features. For image content description, shape is one of important visual features. Since measuring similarity between shapes is complex task, this can be decomposed into two stages namely, feature extraction and measuring similarity among extracted features. Broad classification of shape descriptors is region-based and contour-based methods. In the former methods, the entire area of an object is used for shape description, in later methods only contour of an object is used for shape description [21]. These two methods can be represented by using Fourier Descriptor and Moment variants. Fourier Transformed boundary is used in Fourier Descriptor as the shape feature. Region-based moments are used in Moment variants as the shape feature. For example, image or a sketch can be used to provide queries to the system [4].

A.3) Texture

One of the prime regional descriptors that will be used in retrieval process is Texture. Although Texture alone can't be sufficient to find similar images by separating textured images and then be combined with shape kind of other visual attribute in order to perform effective image retrieval. Texture is mainly used in multimedia databases to classify images. Texture classification can be done either by using structural approach or by using statistical approach [35]. In the former approach texture is represented as cellular, which are time consuming. In the latter approach, texture is represented as complicated pictorial patterns which are effective than earlier ones.

Popular co-occurrence matrix representation for texture feature is proposed by Haralick et.al in 1970 [36]. By using co-occurrence matrix, calculation of Contrast, coarseness, directionality and randomness became easier. Later, Wavelet sub-bands representation for texture feature is proposed (mean and variance). Further, to take advantages of above two representations, these are combined by Thyaparajan et.al. [37]. In this, wavelet transform is combined with co-occurrence matrix. This Wavelet Transform with K L expansion was used in the new technique proposed by Gross et.al to perform texture analysis [38].

B. Semantic Gap

Due to mismatch between human perception and low level features, retrieval systems which uses low-level features are unpredictable and unsatisfactory. This mismatch leads to Semantic Gap [39]. For example, user query "a wonderful sport car" which consists of high level features builds a distance with low level features which is considered as Semantic Gap. In one way, by adding as metadata as possible to the images we can minimize the Semantic Gap but it is not possible in most of cases. In other way, by adding Semantics from gathered feedback provided by user we can minimize the Semantic Gap [21].

C. Feature Matching

In this process, comparison is performed between image signature of the query image and image signature of stored images. Ranks are assigned based on the distance threshold calculated in the earlier comparison. One of the possible distance measures is Euclidean Distance. Feature matching can be done in one way is by regions generated from segmentation and in other way is by an entire image. In recent years, various distance measures also proposed like city-block distance and the Minkowshky distance [22].

D. Performance Measures

Although there are many procedures for measuring the performance of retrieval systems, precision and recall are most standard performance evaluation measures [13].

Precision (P) is defined as the ratio of the number of relevant images over the total number of relevant images available in the databased [33].

Precision = Number of Relevant Images Retrieved / Total Number of Images Retrieved

Recall (R) is defined as the number of retrieved relevant images over the total number of relevant images available in the database [33].

Recall = Number of Relevant Images Retrieved / Number of Relevant images in the database.

Low precision means most relevant images are retrieved, whereas low recall means many of the relevant images are missed.

IV DISTRIBUTED PROCESSING ON HADOOP AND IT'S COMPONENTS

Hadoop is the free and also open source java based software to process large collection of data sets in a distributed computing environment. It was formulated by Apache software foundation as part of the project Apache. By using Hadoop, it is possible to process or deploy applications which requires large number of nodes with very large volumes of data. Due to its 'distributed' nature. Hadoop can achieve rapid data rates between computation nodes, and in case of failure of any node, system can be operated without interruption. Even if some significant number of computation nodes or Inoperative. Hadoop will not lead to catastrophic system failure. Hadoop framework is scalable, as it can grow from small number of servers to thousands with high degree of fault tolerance [40].

In CBIR Systems point of view we knew how to retrieve the image and videos. Hadoop Software Framework can be used for storing, distributing, processing, retrieving and replicating the big image data sets. In that software there are two basic components, 1. Hadoop Distributed File System (HDFS) and 2. MapReduce (MR).

HDFS: It holds very large amount of data and provides easier access. To store such huge data, the files are stored across multiple machines. These files are stored in redundant fashion to rescue the system from possible data losses in case of failure. HDFS also makes applications available to parallel processing. HDFS consists of two elements namely, NameNode and DataNode as shown in Fig. 2.

NameNode: Name Node is a service, responsible to manage the metadata that is the location information of the data stored in HDFS. Whenever a file is placed in the cluster then corresponding entry of its location is maintained by the NameNode. Each HDFS cluster have a single NameNode and also determines the mapping of blocks to DataNode.

Secondary NameNode(SNN): It is important note that the SecondaryNameNode is not a failure node for the NameNode, although the name is misleading. The SecondaryNameNode is responsible for performing periodic housekeeping functions for the NameNode. It only creates check points of the file system present in the NameNode.

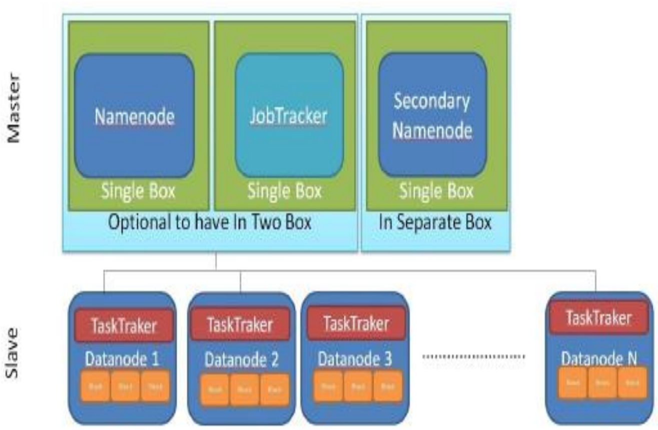

Figure 2 Elements of Map-Reduce and HDFS

DataNode: DataNode is responsible for storing the files in the HDFS. It manages file blocks with in the node. It sends the information to the NameNode about the files and blocks stored in that node and responds to the NameNode for the all operations of file system.

MapReduce: MapReduce programmed is used to retrieve the semantic images and videos, performing distributed processing by single-master and multiple-slave servers. Hadoop can run MapReduce programs written in various languages like Java, Ruby, Pythan and C++. MapReduce Consists of two elements namely, JobTracker and TaskTracker.

JobTracker: JobTracker is responsible for assigning task to TaskTracker in DataNode where data is present, if that not possible it will atleast try to assign tasks to another TaskTracker with in the same rack where data present. If node fails for any reason, then Job tracker assigns the task to another Task tracker where the replica of the data exists across the data nodes. Although this leads the successful completion of job with in cluster.

TaskTracker: TaskTracker is a daemon that takes like Map, Reduce and Shuffle from the JobTracker and and executes. It keeps on sending heartbeat message to the JobTracker to notify that it is alive. Along with the heartbeat it also sends the free slots available within it to process tasks. It also monitors the Map and Reduce Tasks and records their progress and sends this information to JobTracker.

V IMAGE PORCESSING ON HADOOP

Coming to the issue of image processing on Hadoop, there are some challenges [40]. Hadoop is best suited for huge chunks of files, where each file may extend to Giga-Bytes. However, dataset of images in most cases consist of very small but numerous images. Hadoop is not very efficient in handling such a situation because this may involve moving around lots of les throughout the network, leading to significant overhead, this is where Hadoop *sequentialFiles* enter [40]. *SequentialFiles* are e client data structures built-in Hadoop, that can handle binary data (like images very well). For any kind of image processing in Hadoop, all the images first need to loaded onto the HDFS (Hadoop Distributed System). After loading, all the images are concentrated to form *SequentialFiles*. This results in few numbers of large les, which Hadoop can efficiently handle. As part of our preliminary work on Hadoop, an efficient module to load the imaged onto HDFS was developed on Hadoop. This module concentrates all the image les into a Sequence Files and prepares them for further processing.

Although we will be using *sequentialFiles* for our purpose on Hadoop, we started our exploration with Hadoop module developed by a group of students from University of Virginia, called Hadoop Image Processing Interface (HIPI). HIPI was reported to be faster (on average) than *SequentialFiles* for image processing. However, in 2012, Hadoop API underwent a major change. The entire Hadoop API was changed from mapped to map reduce. This resulted in the major version change from Hadoop from Hadoop 1 to

Hadoop 2 (Current stable version is Hadoop 2.7.2). Unfortunately, HIPI is reported only till Hadoop 1 and has remained untested for Hadoop 2. This led to shifting our focus to *SequentialFiles* which perform well in most cases, with an additional bene t of being a part of the Hadoop core module, unlike HIPI.

Image Pre-processing techniques are used before actual retrieval process. These include convert the image into gray-scale image, extract the low-level features and similarity measurement of the image.

A. MapReduce Functionalities

In MapReduce, it is possible to perform parallel distributed processing by writing programs involving the following three steps: Map, Shuffle and Reduce as in Fig. 3. shows an example of the flow when Map and Reduce processes are performed. Because MapReduce automatically performs inter process and maintains load balancing of the processes.

A.1) Mapping Phase

In this phase, image pixels information in the form of Key-Value pair < *Key, Value*> as the input and generates one or multiple pairs, List < *{Key}', {Value}'*> as the intermediate output. Example for 4x4 pixel of RGB image is shown in Fig. 4 and Fig.5.

A.2) Shuffle Phase

Map phase produces the intermediate Key-Value pair(s), they are efficiently and automatically grouped based on Key by the Hadoop system in preparation for the Reduce phase as shown in Fig. 5.

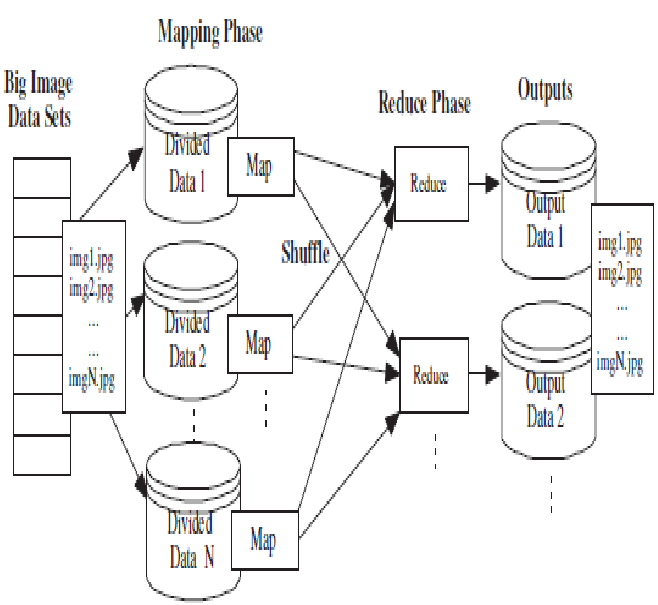

Figure 3 Performing the Map and Reduce Phases [40]

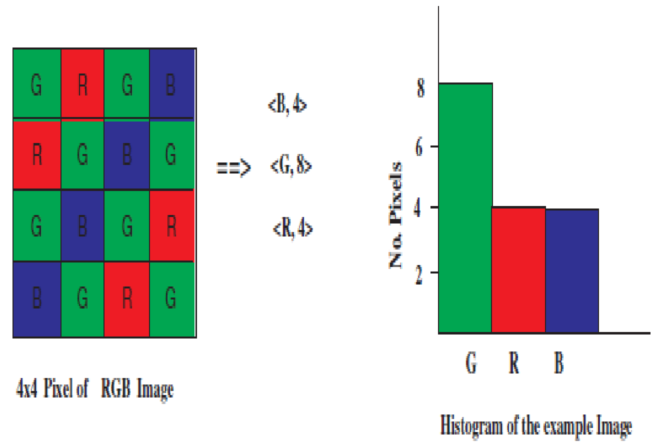

Figure 4 4X4 array pixels of RGB image and Histogram [41]

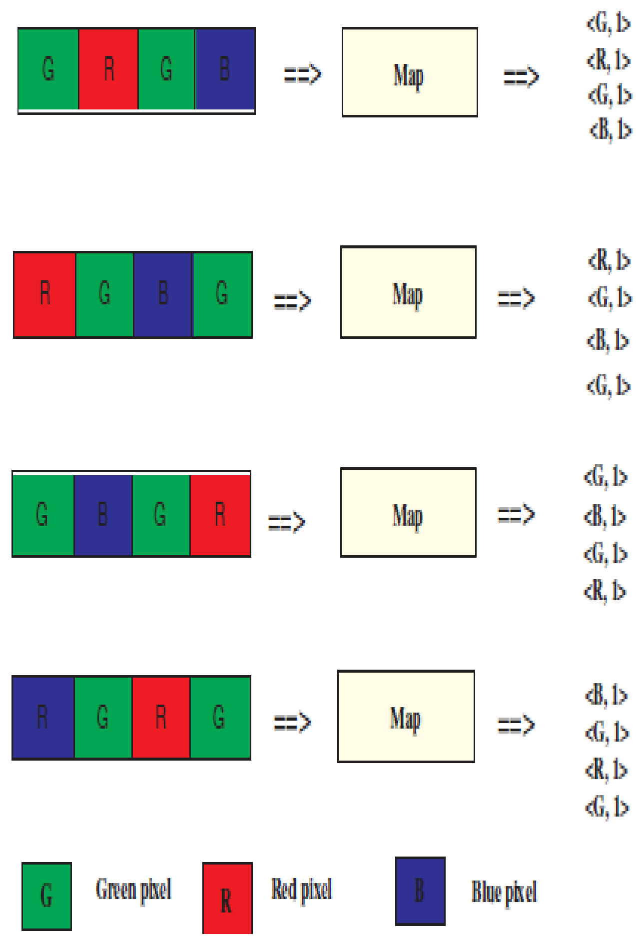

Figure 5 Data Input and output from Map phase [41]

List < *Key',Value'*> => { < *Key'',List(Value'')*> }

The combined output is then essentially a map with unique keys created during the mapping process, with each associated value being a list of values from the mapping phase as shown.

$$
\begin{array}{lcll}
<G,1> & & \left.\begin{array}{l}<B,1>\\<B,1>\\<B,1>\\<B,1>\end{array}\right\} & \Rightarrow \; <B,4> \\
<R,1> & & & \\
<G,1> & & & \\
<B,1> & & & \\
<R,1> & & \left.\begin{array}{l}<G,1>\\<G,1>\\<G,1>\\<G,1>\\<G,1>\\<G,1>\\<G,1>\\<G,1>\end{array}\right\} & \Rightarrow \; <G,8> \\
<G,1> & & & \\
<B,1> & & & \\
<G,1> & \Rightarrow & & \\
<G,1> & & & \\
<B,1> & & & \\
<G,1> & & & \\
<R,1> & & & \\
<B,1> & & \left.\begin{array}{l}<R,1>\\<R,1>\\<R,1>\\<R,1>\end{array}\right\} & \Rightarrow \; <R,4> \\
<G,1> & & & \\
<R,1> & & & \\
<G,1> & & &
\end{array}
$$

A.3) Reduce Phase

The output of the shuffle phase, Map is given as input to the Reduce phase, with keys being all the unique keys found in the Mapping operation and the values being the collected values for each key from the Mapping process. The output of the Reduce phase can be any arbitrary Value and can be represented by Reduce [41].

Reduce: {<*Key''*, *List (Value'')*>} => List(*Key'''*,*Value'''*) or List(*Value'''*). The input a < *K'*, *LISTV'*> pair, where "*LIST*" is a list of all V' values that are associated with given Key *K'*. The Reduce function produces an additional Key-Value pair as the output. Input data and output from Reduce Phase

$$
\begin{array}{lcl}
<B(1,1,1,1)> & \boxed{\text{Reduce}} & <B,4> \\
<G(1,1,1,1,1,1,1,1)> & \boxed{\text{Reduce}} & <G,8> \\
<R(1,1,1,1)> & \boxed{\text{Reduce}} & <R,4>
\end{array}
$$

By Combining Multiple Map and Reduce process, we can accomplish complex tasks which cannot be done via a single Map and Reduce execution.

In the above section CBIR system to increase the performance of calculation by using MapReduce Framework. The functions of CBIR Systems such feature extraction and similarity measurement are applied by using MapReduce scheme. In insertion module (Feature extraction), images are divided by NameNode, determines the mapping of blocks to DataNodes [41]. Usually one per DataNode in a cluster, DataNode manages the storage that is attached to the nodes on which they run. Another module is query processing module (Similarity measurement), it executes the vector feature of images from database images and query images. The block size determines base as a size of vector feature, depends an image descriptor as shown in Fig. 6

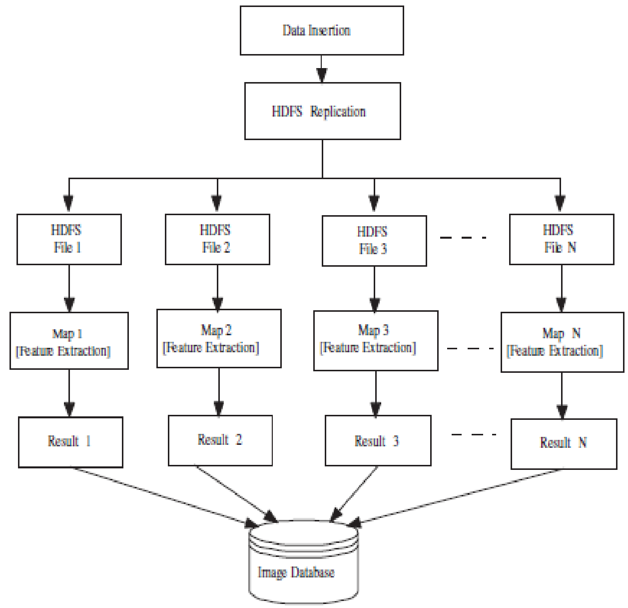

Figure 6 A distributed Processing Framework using MapReduce for Data Insertion Module in CBIR System [41]

VI CONCLUSION

In this paper, we focused on Distributed processing on similarity measurement and feature extraction using Hadoop MapReduce. We observed that the retrieval process is very fast, compared to previous CBIR systems. We explain about how image retrieved functions can be processed by using MapReduce Scheme. In Hadoop, parallel distributed processing is also possible. This can be done by using MapReduce functions. In the future, semantic gap and Multidimensional indexing implementation and testing in terms of computation time using Hadoop software framework will be proposed.

REFERENCES

[1] S.-K. Chang and A. Hsu, “Image information systems: where do we go from here?," IEEE transactions on Knowl- edge and Data Engineering, vol. 4, no. 5, pp. 431-442, 1992.
[2] H. Tamura and N. Yokoya, “Image database systems: A survey," Pattern recognition, vol. 17, no. 1, pp. 29-43, 1984.
[3] B. Furht, S. W. Smoliar, and H. Zhang, Video and im- age processing in multimedia systems, vol. 326. Springer Science & Business Media, 2012.
[4] Y. Rui, T. S. Huang, and S.-F. Chang, “Image retrieval: Current techniques, promising directions, and open is- sues," Journal of visual communication and image representation, vol. 10, no. 1, pp. 39-62, 1999.

[5] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, “Content-based image retrieval at the end of the early years," IEEE Transactions on pattern analysis and machine intelligence, vol. 22, no. 12, pp. 1349-1380, 2000.
[6] C.Wang, L. Zhang, and H.-J. Zhang, “Learning to reduce the semantic gap in web image retrieval and annotation," in Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, pp. 355-362, ACM, 2008.
[7] V. N. Gudivada and V. V. Raghavan, “Content based image retrieval systems," Computer, vol. 28, no. 9, pp. 18-22, 1995.
[8] Y. Ang, A. Narasimhalu, and S. Al Hawamdeh, “Image information retrieval system," p. Chapter IV:2, 1997.
[9] B. Schatz and H. Chen, “Building large-scale digital libraries," Computer, vol. 29, no. 5, pp. 22-26, 1996.
[10] A. Pentland and R. Picard, “Special issue on digital libraries," IEEE Trans. Patt. Recog. and Mach. Intell, vol. 18, no. 7-8, pp. 673-733, 1996.
[11] H. Kaur and K. Jyoti, “Survey of techniques of high level semantic based image retrieval," IJRCCT, vol. 2, no. 1, pp. 015-019, 2013.
[12] S. B. P. V. R. G. D. D. Dhobale, B. S. Patil, “Semantic understanding of image content," International Journal of Computer Science Issues, 2011.
[13] R. Datta, D. Joshi, J. Li, and J. Z. Wang, “Image retrieval: Ideas, inuences, and trends of the new age," ACM Computing Surveys (Csur), vol. 40, no. 2, p. 5, 2008.
[14] Y. Lu, P. Gao, R. Lv, Z. Su, and W. Yu, “Study of content-based image retrieval using parallel computing technique," in Proceedings of the 2007 Asian technology information program's (ATIP's) 3rd workshop on High performance computing in China: solution approaches to impediments for high performance computing, pp. 186-191, ACM, 2007.
[15] O. Kao, G. Steinert, and F. Drews, “Scheduling aspects for image retrieval in cluster-based image databases," in Cluster Computing and the Grid, 2001. Proceedings. First IEEE/ACM International Symposium on, pp. 329- 336, IEEE, 2001.
[16] Y. Ling and Y. Ouyang, “Image semantic information retrieval based on parallel computing," in Computing, Communication, Control, and Management, 2008. CCCM'08. ISECS International Colloquium on, vol. 1, pp. 255-259, IEEE, 2008.
[17] Y. Gao, J. Fan, H. Luo, and S. Satoh, “A novel approach for filtering junk images from google search results," Advances in Multimedia Modeling, pp. 1-12, 2008.
[18] T. White, Hadoop: The de_nitive guide. " O'Reilly Media, Inc.", 2012.
[19] J. Venner, Pro hadoop. Apress, 2009.
[20] W. Premchaiswadi, A. Tungkatsathan, S. Intarasema, and N. Premchaiswadi, “Improving performance of content-based image retrieval schemes using hadoop mapreduce," in High Performance Computing and Simula- tion (HPCS), 2013 International Conference on, pp. 615- 620, IEEE, 2013.
[21] J. Eakins and M. Graham, “Content-based image retrieval," 1999.
[22] A. Khokher and R. Talwar, “Content-based image retrieval: state-of-the-art and challenges," International Journal of Advanced Engineering Sciences and Technologies, vol. 9, no. 2, pp. 207-211, 2011.
[23] S.-F. Chang, A. Eleftheriadis, and R. McClintock, “Next generation content representation, creation, and searching for new-media applications in education," Proceedings of the IEEE, vol. 86, no. 5, pp. 884-904, 1998.
[24] R. C. Veltkamp and M. Tanase, “Content-based image retrieval systems: A survey," 2001.
[25] R. S. Choras, “Image feature extraction techniques and their applications for cbir and biometrics systems," International journal of biology and biomedical engineering, vol. 1, no. 1, pp. 6-16, 2007.
[26] A. Sandhu and A. Kochhar, “Content based image retrieval using texture, color and shape for image analysis," International Journal of Computers & Technology, vol. 3,\ no. 1c, pp. 149-152, 2012.
[27] C. S. McCamy, H. Marcus, J. Davidson, et al., “A color-rendition chart," J. App. Photog. Eng, vol. 2, no. 3, pp. 95-99, 1976.
[28] M. Miyahara, “Mathematical transform of (r, g, b) color data to munsell (h, s, v) color data, spie visual commun," Image Process, vol. 1001, 1988.
[29] T. V. Papathomas, T. E. Conway, I. J. Cox, J. Ghosn, M. L. Miller, T. P. Minka, and P. N. Yianilos, “Psychophysical studies of the performance of an image database retrieval system.," in Human Vision and Elec- tronic Imaging, pp. 591-602, 1998.
[30] J. R. Smith and S.-F. Chang, “Single color extraction and image query," in Image processing, 1995. Proceedings., International conference on, vol. 3, pp. 528-531, IEEE, 1995.
[31] J. R. Smith and S.-F. Chang, “Tools and techniques for color image retrieval.," in Storage and Retrieval for Image and Video Databases (SPIE), vol. 2670, pp. 2-7, 1996.
[32] A. Chadha, S. Mallik, and R. Johar, “Comparative study and optimization of feature-extraction techniques for content based image retrieval," arXiv preprint arXiv:1208.6335, 2012.
[33] N. Goel and P. Sehgal, “A refined hybrid image retrieval system using text and color," International Journal of Computer Science Issues, vol. 9, no. 4, pp. 48-56, 2012.

[34] H. K. Bhuravarjula and V. V. Kumar, “A novel con- tent based image retrieval using variance color moment," IJCER, vol. 1, no. 3, pp. 93-98, 2012.

[35] Y. Q. Chen, M. S. Nixon, and D. W. Thomas, “Statistical geometrical features for texture classi_cation," Pattern recognition, vol. 28, no. 4, pp. 537-552, 1995.

[36] R. M. Haralick, K. Shanmugam, et al., “Textural features for image classi_cation," IEEE Transactions on systems, man, and cybernetics, no. 6, pp. 610-621, 1973.

[37] K. Thyagarajan, T. Nguyen, and C. E. Persons, “A maximum likelihood approach to texture classification using wavelet transform," in Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, vol. 2, pp. 640-644, IEEE, 1994.

[38] M. H. Gross, R. Koch, L. Lippert, and A. Dreger, “Mul-tiscale image texture analysis in wavelet spaces," in Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference, vol. 3, pp. 412-416, IEEE, 1994.

[39] Y. Mamatha and A. Ananth, “Content based image retrieval of satellite imageries using soft query based color composite techniques”, International Journal of Computer\ Applications, vol. 7, no. 5, pp. 40-45, 2010.

[40] M. Yamamoto and K. Kaneko, “Parallel image database processing with mapreduce and performance evaluation in pseudo distributed mode," International Journal of Electronic Commerce Studies, vol. 3, no. 2, p. 211, 2012.

[41] A. Tungkasthan and W. Premchaiswadi, “A parallel processing framework using mapreduce for content-based image retrieval," in ICT and Knowledge Engineering (ICT&KE), 2013 11th International Conference on, pp. 1- 6, IEEE, 2013.

BIOGRAPHY

Suresh Kanaparthi received the B.Tech. degree in Computer Science and Engineering from JNT University, Hyderabad, India and M.Tech degree in Computer Science and Technology from Andhra University College of Engineering, Vsakhapatnam, India. Currently he is pursuing the Ph.D degree in the Department of Computer Science and Engineering at National Institute of Technology, Warangal, India.

Dr. U.S.N. Raju received B.E. degree in Computer Science and Engineering from Bangalore University, India in 1998, the M.Tech degree in Software Engineering from JNT University Hyderabad, India in 2002, and the Ph.D. degree from JNT University Kakinada, India in 2010. He has two years of experience in IT industry and thirteen years of experience in teaching at various reputed educational institutes. At present, he is working as assistant professor in the Department Computer Science and Engineering at National Institute of Technology, Warangal, India. His current research area includes Computer Vision and Image Processing, Content Based Image Retrieval, Big Image Data Processing and Video Processing. He is a member of IEEE, CSI and a life member of ISTE.