Private Content Based Image Retrieval Using Hadoop

Dissertation

Submitted in partial fulfillment of the requirement for the degree of Master of Technology in Computer Engineering

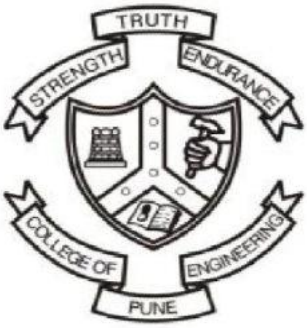
By

Swapnil P. Dravyakar

MIS No: 121222008

Under the guidance of

Prof. Sunil B. Mane



Department of Computer Engineering and Information Technology

College of Engineering, Pune

Pune – 411005

June, 2014

DEPARTMENT OF COMPUTER ENGINEERING AND

INFORMATION TECHNOLOGY,

COLLEGE OF ENGINEERING, PUNE

CERTIFICATE

This is to certify that the dissertation titled

Private Content Based Image Retrieval

Using Hadoop

has been successfully completed

By

Swapnil P. Dravyakar

MIS No: 121222008

and is approved for the partial fulfillment of the requirements for the degree of

Master of Technology, Computer Engineering

Prof. Sunil. B. Mane
Project Guide,
Department of Computer Engineering
and Information Technology,
College of Engineering, Pune,
Shivaji Nagar, Pune-411005.

Dr. J. V. Aghav
Head,
Department of Computer Engineering
and Information Technology,
College of Engineering, Pune,
Shivaji Nagar, Pune-411005.

June 2014

Acknowledgments

I express my sincere gratitude towards my guide Prof. Sunil B. Mane his constant help, encouragement and inspiration throughout the project work. Without his heeded parental guidance, this work would never have been a successful one. I also like to convey my sincere gratitude to Dr. J. V. Aghav (HOD), Dr. V. K. Pachghare, all faculty members and staff of Department of Computer Engineering and Information Technology, College of Engineering, Pune for all necessary cooperation in the accomplishment of dissertation. Last but not least, I would like to thank my family and friends, who have been a source of encouragement and inspiration throughout the duration of the project.

Swapnil P. Dravyakar

College of Engineering, Pune

ABSTRACT

Today, huge quantity of data, in the form of images, is produced through digital cameras, mobile phones and photo editing software. A large part of this data is stored in online repositories. This data is private to each user and consequently it should not be accessible by others. In earlier systems, images were searched by the tags or keywords or description assigned to them wherein if an image is wrongly described, querying image search will not result in the required image.

Here, a ‘content based’ search has been proposed which analyzes the content of the image. System will allow user to upload a particular image and depending on combined values of color, shape and texture, the system will retrieve similar images from database.

The system incorporates techniques for upload and search of images over large datasets of images based on the content of the images rather than the keywords and tags or any other textual information. Another part of the system is Hadoop distributed file system (HDFS). Hadoop defines a framework which allows processing on distributed large sets across clusters of computer.

Keywords: HDFS, Feature Extraction, CBIR, Image Retrievals.

Contents

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Nowadays, technology has become progressively advanced. This has led to cheap and advanced multimedia devices which have given rise to huge data volumes. This massive data needs to be stored in database for various applications such as prevention of crime, medical, security, etc. These applications have created a requirement for effective and efficient methods of storage, search and retrieval of images via parallel processing techniques.

With growing technology, security has become a crucial concern for visual forms of information. For example, suppose an organization has developed a novel face recognition algorithm. The organization will wish that the input images as well as images in their database are not revealed publicly by the algorithm or any other means. We attempt to find a method for storage and retrieval of images in a way that such an objective is achieved.

Earlier, the images were stored with associated labels and searching was done on the basis of these labels. But, such a method is prone to many errors. If the images are wrongly annotated, irrelevant images might be retrieved. Moreover, it is a laborious task to search for an image which has been assigned a wrong label [3].

Nowadays, with extensive reach of social networks, users have become more concerned about their privacy and their data being stored on servers. Some users even prefer to hide their details from database admin.

If stored information on server is seen by database admin, a user's privacy is breached and the chances of misuse of this information increase. If an organization's employee details are stored on a server along with their photographs for face recognition, the organization prefers to keep this information out of reach of any other user or database admin [1]. If this information is not hidden perfectly, a compromised database admin may be able to access them and use it for his/her own benefits. Such a situation needs to be avoided.

1.1 Content Based Image Retrieval

The term "Content-Based Image Retrieval" is used for retrieving the corresponding images from the database based on their feature of images which derived the image itself like texture, color and shape and domain specific like human faces and fingerprints.

The retrieval on the based on the content of an image is to be more effective than the text based which is called content based image retrieval that are used for a various applications like vision techniques of computer [2].

Traditionally, search of the images are using text, tags or keywords or annotation assigned to the image while storing into the databases. Whereas if the image which is stored in the database are not uniquely or specifically tagged or wrongly described then it's insufficient, laborious and extremely time consuming job for search the particular image in the large set of databases [9]. for these purpose obtaining the most accurate result CBIR system are used which searches and retrieve the query images from the large databases based on their image content like color, texture and shape which derived from the image itself.

1.1.1 Color

Color is the most commonly used attribute in image retrieval systems. Retrieval in these systems is done on the basis of similarity in color. For each image in database, color histogram is computed which shows pixel position of each color in the image [4]. Most commonly used color based image retrieval methods are RGB and HSV.

1.1.2 Texture

Texture similarity may not seem to be useful for image retrieval. Texels are made of intensity of pixel. They help in differentiating between textured and non-textured images in database. The computation of texels may be defined based on the direction of texture, its coarseness and regularity. Texture has certain classification like fine, gross with respect to pixel of an image. This retrieves textured regions in images on the basis of similarity to automatically-derived towards representing important classes of texture within the collection of images [5].

1.1.3 Shape

At a primitive level, shape is one of the basic attributes of any image. Image retrieval methods based on shape are also devised. A number of features are computed by the shape of the objects in image. Queries of image are answered by computing the same set of features from the input image and the images with most similar features are given as output [6]. Features are defined at two levels - local and global. These features include aspect ratio, circularity and moment invariant, segments of consecutive boundary, etc. Shape matching does not result accurately in case of 3D objects in images.

1.2 Private Information Retrieval

With increasing demands of data storage, users intend to store their data in a reliable storage media. This data may be managed by 3rd party services but should be accessible only to the respective authorized user.

A Private Information Retrieval (PIR) protocol allows a user to retrieve an item from a server in possession of a database without revealing which item is retrieved.

1.3 Image Upload and Search

As discussed earlier, the text based storage and retrieval has drawbacks which are difficult to overcome. In contrast to this, the content based image storage gives the facility to the user for searching the image from the database as per his/her requirement.

1.4 Hadoop

Apache Hadoop is open source software which processes on large scale storage on commodity hardware.

There are several modules for Apache Hadoop.

- Hadoop Common: These libraries are needed for running different programs on Hadoop Module.
- Hadoop Distributed File System (HDFS): Hadoop Distributed file System is the storage which stores the large amount of data on the commodity hardware.

- Hadoop YARN: These are used for scheduling the resources and users' application.
- Hadoop Map-Reduce: Hadoop Map-Reduce processes large data in parallel and gives result with the best performance.

Hadoop has been designed such that its software framework can automatically manage and deal with hardware failure. Hadoop Map-Reduce and HDFS are designed with the help of Google Map-reduce and Google file system.

1.4.1 Hadoop Architecture

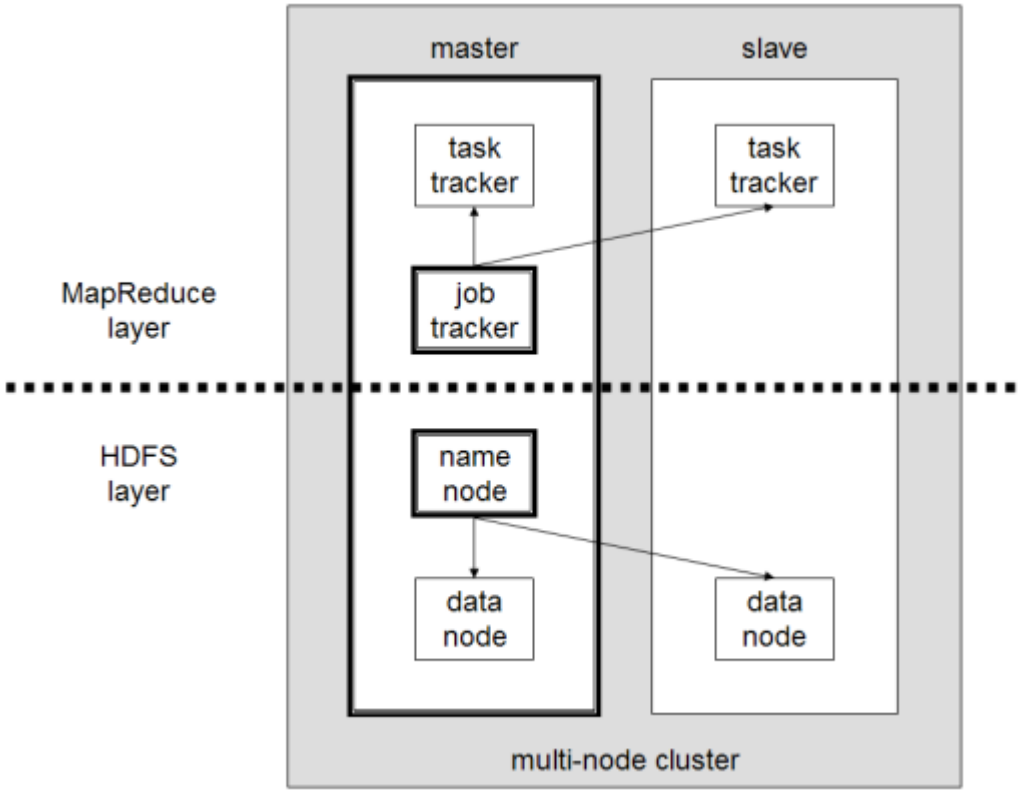


Figure 1.1: A Multi-node Hadoop Cluster

Figure 1.1 shows a multi-node Hadoop cluster. Hadoop provides location awareness compatible file system. HDFS is the backbone of the Hadoop system which stores the data by replication and makes different copies of data on to the different rack for the purpose of fault tolerance.

Hadoop clusters consist of a Master node and multiple worker nodes. Master node consists of Job tracker, Task tracker, Datanode and Namenode. Task tracker and datanode operate on slave nodes.

HDFS works on large cluster. It hosts the file system using Namenode. Secondary Namenode generates snapshots of Namenode. Similarly, the job scheduling is done by job tracker. All the information stored on the datanode is in known to the Namenode.

When user interacts with Hadoop system, the Namenode becomes active and it possesses all the information about datanode data.

1.4.2 Hadoop Distributed File System (HDFS)

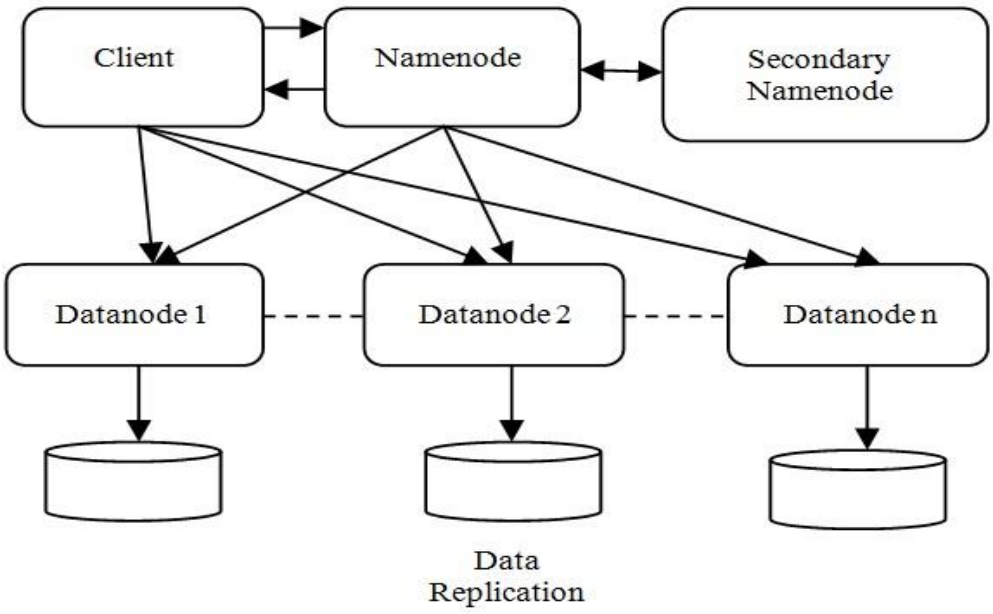


Figure 1.2: HDFS Architecture

Hadoop distributed file system (HDFS) supports java programs for storing the data and contains portable file system with the features of scalability and distributed across multiple machines. Figure 1.2 shows the HDFS file system. User can access and store their data with the help of Namenode. In such a case, Namenode becomes a single point of failure. As such, a secondary Namenode is always active and it takes snapshots from Namenode and stores all the information itself. If the Namenode fails, it can recover data from the secondary Namenode. Secondary Namenode behaves like

a checkpoint. HDFS has the advantage of data awareness between task tracker and job tracker.

1.4.3 Map-Reduce

Map-Reduce is the programming model that works on the large datasets with parallel and distributing algorithm on cluster. Map-reduce program is composed of Map function which performs sorting and filtering of large data sets. Reduce function performs the summary operation which combines the result and gives optimized result. Working of Map-Reduce is shown in figure 1.3. These functions are run parallel on large data. The computation is done on key/value pairs.

For processing large data, process parallelization is done using Map–reduce framework across huge datasets. Computational processing occurs on stored data on file system.

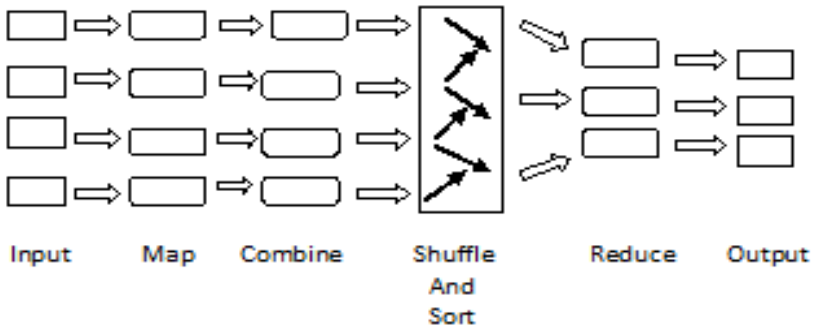


Figure 1.3: Map-Reduce Phase

"Map" step: The node, called the Masternode, takes the input, divides a large problem are into smaller sub-problems and distributes them among worker nodes in a multi-level tree structure. The worker nodes process these sub-problem and pass the result back to the master node.

"Reduce" step: Reduce function accepts input from Map-Function, combines the answers to all the sub-problems, collect it in master node and forms the output. Map operations are run in parallel and accordingly, the reducer performs the reduce phase on same keys presented in same time. Each map function is associated with a reduce function.

The map-reduce function usually works on large data of the order of Petabytes and Gigabytes on commodity hardware and is capable of sorting data in few hours. Map and Reduction operations of the Map-Reduce allow for distributed processing. Since each mapping operation is independent of the other, all maps can be performed in parallel although, in practice, it is limited by the number of independent data sources and/or number of CPUs near each source. Similarly, a set of 'reducers' can perform the reduction phase provided that each reducer receives the output from its associate map operation at the same time or if the reduction function is associative.

1.4.4 HBase

HBase works on top of HDFS. It is a column oriented distributed database built on top of HDFS. HBase is used when real-time read/write random-access to very large datasets is required. It is a Hadoop application. HBase can scale linearly just by adding nodes. It is not relational and does not support SQL, but given the proper problem space, it is able to do what an RDBMS cannot: host very large, sparsely populated tables on clusters made from commodity hardware.

CHAPTER 2

LITERATURE SURVEY

2.1 Private Content Based Image Retrieval

J. Shashank et al.[1] describe the content level access in case of querying an image from a database. The private content of users, in form of images, should not be visible to any person except the owner. This data should even be protected from database admin. PCBIR deals with this issue and is capable of retrieving similar type of images from the image database.

This paper describes the user issues about data privacy when the images are stored on the database server. Secure image retrieval from the database server will be an active application for oblivious transfers. The algorithm is customizable for structure of hierarchical data. The paper also provides experimental study on accuracy, efficiency and scalability of the algorithm. It is proved that the algorithm works on large database using variety of image indexing technique.

2.2 Content based image retrieval using color, texture and shape

S. M. Patil et al.[2] deal with images which contain complex information having dense form. Human eye can extract and understand such images after a year of training. This paper describes the information related to low level properties of image by physical and mathematical properties using a complex model. This system allows storing image into the database by its features and gives the result on the basis of feature extracted of Query image and then results most similar matches.

Images retrieval is increasing and crucial importance is given to domain specific information. Large and distributed collections of scientific, technical images are retrieved using sophisticated and precise measures of similarity and query based semantics.

2.3 Map Reduce Neural Network Framework for Efficient Content Based Image Retrieval from Large Datasets in the Cloud

V. Sitalakshmi et al.[3] described that searching the images which are wrongly annotated or defined gives incorrect result because of the text based search. Such a scenario is avoided by employing image retrieval from large database using map reduce framework. Their system was tested on medical image database using neural network on the cloud environments.

The map reduce framework distributes the work in parallel and gives faster result in shorter time with the working on the Gigabyte and Petabyte of data.

Image retrieval has been done on the combined value RGB values on the low, medium and large values whereas the images stored on the database are retrieved on the basis of neural methods.

2.4 Local Tetra Patterns: A New Feature Descriptor For Content – Based Image Retrieval

S. Murala et al.[4] proposed an algorithm for indexing and retrieval of images for Content Based Image Retrieval Using Local tetra Patterns (LTrPs). It defines the standard combination of local binary patterns (LBP) and local ternary patterns (LTP) which are computed by referenced pixel and surrounding its neighbors by calculating difference in gray level. The proposed method gives the relationship between referenced pixel with its neighbors pixel with respect to the direction that are computed by derivatives in first order in vertical and horizontal direction.

2.1.1 LOCAL PATTERNS

LBPs

Local binary patterns are introduced for texture classification. Given a center image pixel, this LBP values are compared for both gray values with neighbors.

Whereas g_c is center pixel gray value, g_p neighbor gray value, neighbors number is represented by P and neighborhood radius is the R.

LTPs

It is the Three Value Code which is called LTP, zone of the gray value which is the width $\pm t$ around g_c are zero quantized, those values are above $(g_c + t)$ are +1

quantized and those values below ($g_c - t$) are -1 quantized, i.e. indicator f1 (x) is replaced by three-valued function.

LDPs

Face recognition uses the proposed LDPs. They are nondirectional and higher order extended called LDP. In this system calculation of the images are along $0^0, 45^0, 90^0\ 135^0$ directions of center pixels.

LTrPs

Local Tetra Patterns are the combination of LBP, LDP and LTP. It describes the special local texture structure using center gray pixel by direction. The center level direction of gray level pixel is denoted by I. g_c denotes the center pixel in I, g_h horizontal and g_v vertical neighborhoods of g_c respectively. Each part is converted into three binary patterns by tetra patterns.

Upload: In this system, the images are already uploaded to the database.

Query Image: The search images are input by the user. It is then converted into grayscale which is calculated along the direction of each pixel compared with database image on the pixel value.

This implementation is for small size images because it takes more time for uploading and comparisons of images on the database.

2.5 Comparative study on content –Based Image Retrieval

A. Hussain et al.[5] stated that the importance of similarity measurement. The images need to be kept in databases due to increased quantity of digital data in fields like medicine, private life photos and journalism. In order to retrieve the desired images from the database, efficient and accurate retrieval system is required. An image can be retrieved based on the features like color, texture, shape and the content. Most similar images having the least distance are searched and given as output. This paper gives three similarity methods that find the similarity between two images. This technique is then used to compare the query images from images that are stored into database. CBIR possess great importance for fields wherein they have to manage within single retrieval similar images methods. For quick accessing the database, the paper describes two methods of comparing by Euclidian distance formula with helps the system to get an accurate result.

Summary of Literature survey

From the study of various sources, it was concluded that in today's era the amount of digital images are growing in a very explosive manner. The storage requirement for storing these images is also increasing from gigabyte to petabyte. Searching and retrieval of particular images from the massive database is not possible when the images in the database are wrongly annotated and described. For getting the correct image, during the search, content based image retrieval can be used to search and retrieve the images from the massive collection of images. The query image is compared with database images on the basis of their feature descriptor; this can improve efficiency of searching and retrieval.

CHAPTER 3

MOTIVATION

Every day the digital images data are growing explosively. Photography and television technology forms the invention that have played major role in facilitating the communication and capture of image data. The digitization process cannot manage collection of images by itself easily. It requires catalogue and indexing. Images are important in electronically-mediated communication. It is difficult to locate a particular image in a varied and huge image collection by simply searching and uploading. Earlier image retrieval was based on the text based or annotation assigned to different images. But querying a wrongly annotated image by text does not give required image in result. There is a need of fast and secured technique to upload, search and retrieve these images on user demand.

3.1 Problem Definition

The purpose of this research is to make efficient private content based image retrieval for easy upload and search of an image from the huge collection of images that are stored on the database. Additionally, image access from the database should be secure such that the content is not revealed.

3.2 Scope of Research

This research focuses on scenario wherein private content based image retrieval is needed to be done on the basis of a queried image. The search results in images similar to the queried image from the huge database without revealing the content of the image to the database admin or any other user. It gives the storage and retrieval of encrypted images over HDFS by changing the CBIR algorithm.

3.3 Objectives

- To implement CBIR with Hadoop
- Search an image by content
- Implement encryption technique for secure image retrieval
- To evaluate the proposed solution

3.4 System Requirement Specifications

Hardware:

One Desktop system with Intel or AMD processor and 2 GB of RAM is required at the least. Latest technology will provide better results for e.g. cluster of five nodes having Intel core series processor (i3, i5, i7) of high frequency with 4 GB RAM will definitely perform better than five node cluster having old processors working on less frequency and less RAM. The nodes in a cluster should have same configuration. If the nodes have different RAM capacity or processor frequency, cluster will not perform up to capacity.

Software:

1. Operating System: Any open source Linux operating system.
2. Coding Language: software Java language (JDK)
3. IDE: Eclipse environment

CHAPTER 4

SYSTEM ARCHITECTURE

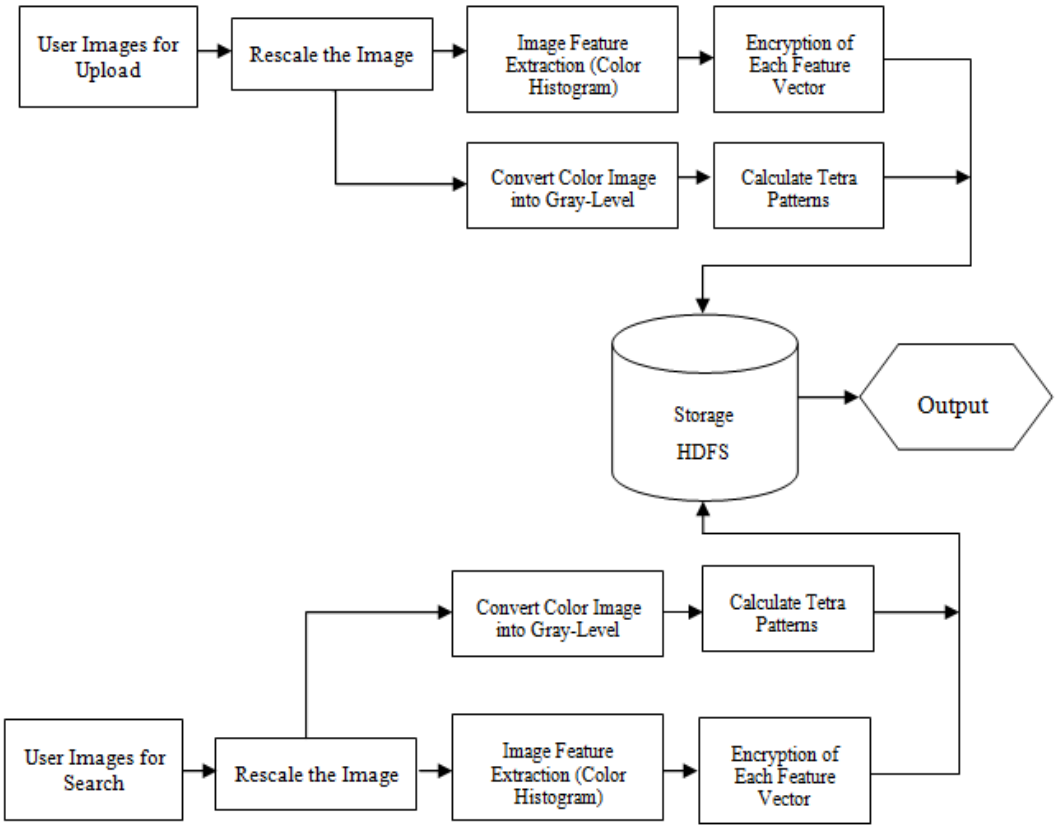


Figure 4.1: System Architecture

The developed system provides the application interface for users. The system incorporates two phases – 1) Upload the images and 2) Search the image from the database. Both modules are deployed on Hadoop cluster. For uploading phase, the data is inputted for storage of HDFS and Query Image is searched from the database.

4.1 Upload Images: System allows user to upload one or more images at a single point into the system. In this proposed system, adding the image into the databases has sub processes. The user uploads the images on database of Hadoop which is called HDFS.

This process has been split into two phases as shown in figure 4.1.

Phase 1: user can upload one or more images into the database by giving the image path at a single point from the GUI. The uploaded images by the user may be of different sizes. Large or different sized images may take more time for computation. To deal this scenario, the system has the facility of rescaling the image. After the image has been rescaled, it is converted into grayscale. The first-order derivatives of vertical and horizontal axes are applied and every direction of pixel that is calculated based on the center pixel forms patterns which are divided into four parts. Then tetra patterns are calculated and are separated into three binary patterns. Magnitude of center pixel is computed. The feature extracted from the binary patterns and binary patterns obtained from the magnitude patterns are combined to form a feature vector. The converted image is then stored in .text file format on database.

Phase 2: After the image is rescaled, as described in phase 1, its color features are calculated i.e. RGB values for each pixel of image. These calculated feature values are encrypted with simple encryption and a .txt file is formed from the result which is then stored into the database.

4.2 Searching of Images: Like uploading phase, system provides GUI for user to search and retrieve the images from the database by the query image. When a user wishes to search a particular image, he/she has to provide the search image path. The processes included in upload phase are same for search process. The searched image is matched with the help of pixel value for LTrP method. It is also searched by color method where the color encrypted values are compared with the values stored in database.

4.3 Image Database Privacy: In this system, the image stored is in the encrypted values into .text format on the HDFS database of hadoop. This format, in raw form, does not provide any information about the image.

CHAPTER 5

IMPLEMENTATION OF SYSTEM

For ease of interaction between user and system, GUI has been provided in the system. The GUI has been developed in Eclipse with the lightweight Java swing components which are platform independent and are a part of Java GUI widget toolkit. With the help of swing components, look and feel of the GUI is customizable to match different platforms. Hadoop incorporates Java because of its object-oriented nature and other features like concurrency and capability of running enterprise software.

5.1 Upload Process

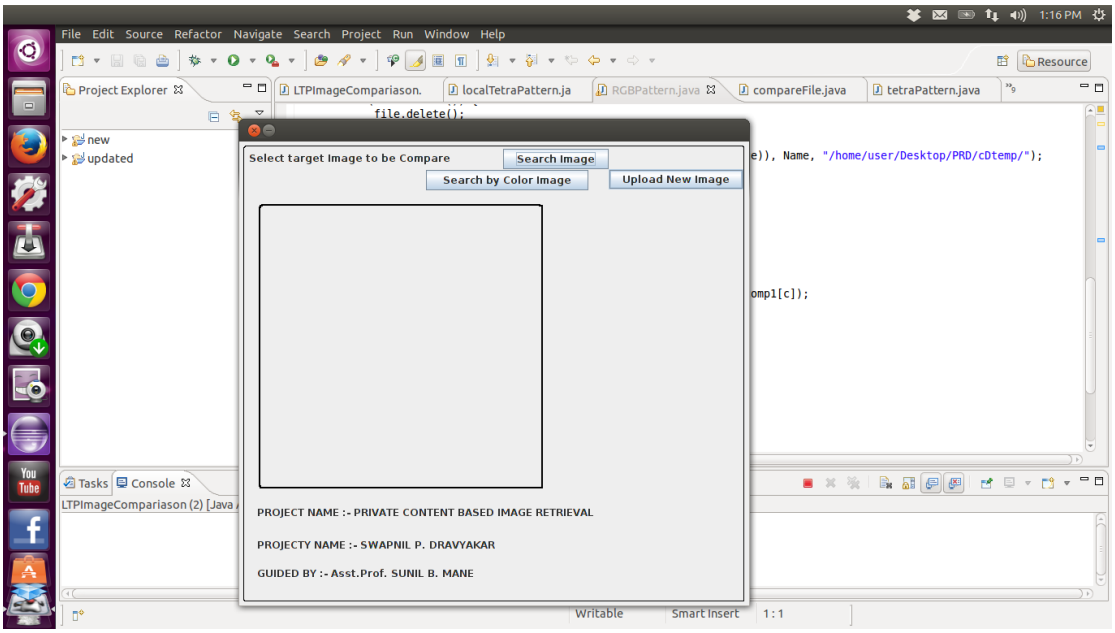


Figure 5.1.1: Design of Graphical User Interface (GUI)

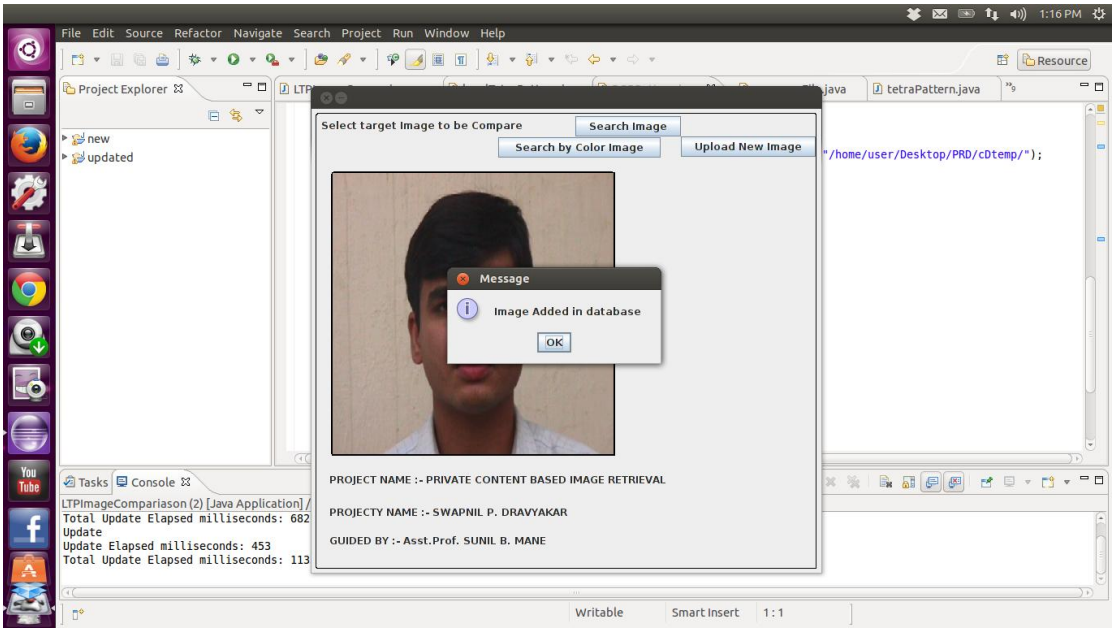


Figure 5.1.2: Upload Process

Figures 5.1.1 and 5.1.2 show the GUI. The code is exported to .jar file. To run the code on Hadoop cluster, the exported .jar file is run on the Hadoop cluster which then generates the GUI. This GUI has three buttons – Upload, Search Image and Search by Color Image button. When user clicks on the Upload Button, the path for uploading image into the Hadoop Distributed File System (HDFS) is generated.

5.2 Hadoop Distributed File System (HDFS)

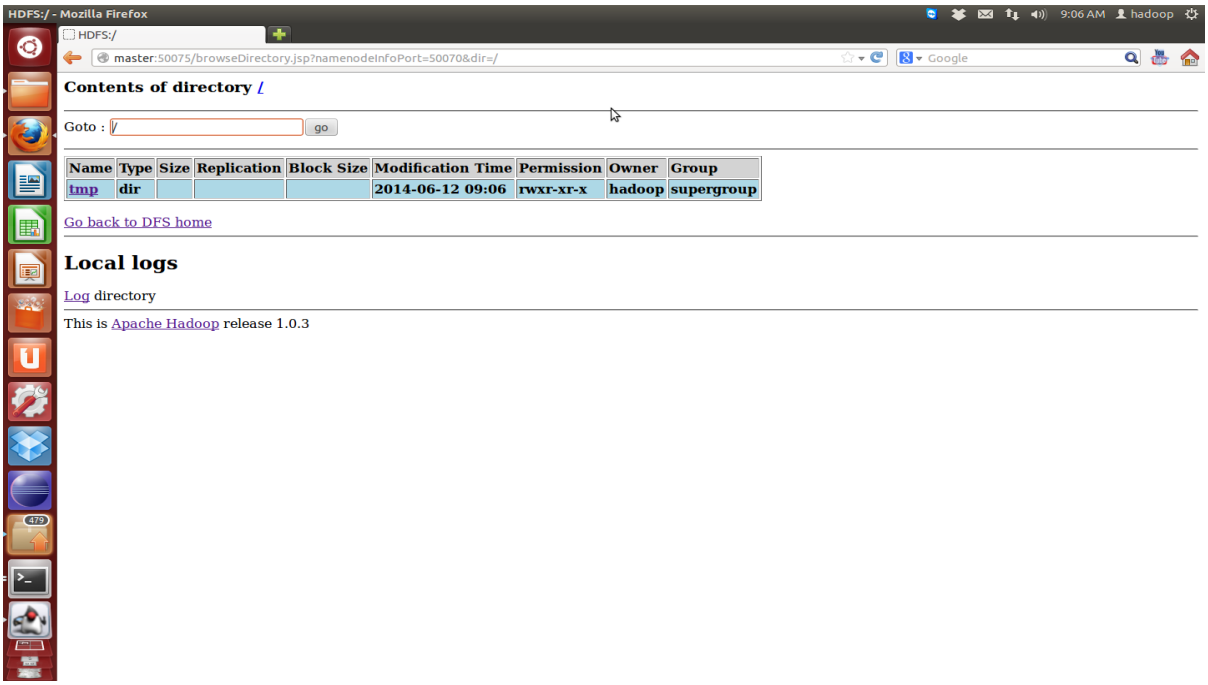


Figure 5.2.1: Hadoop Distributed File System (HDFS) Environment

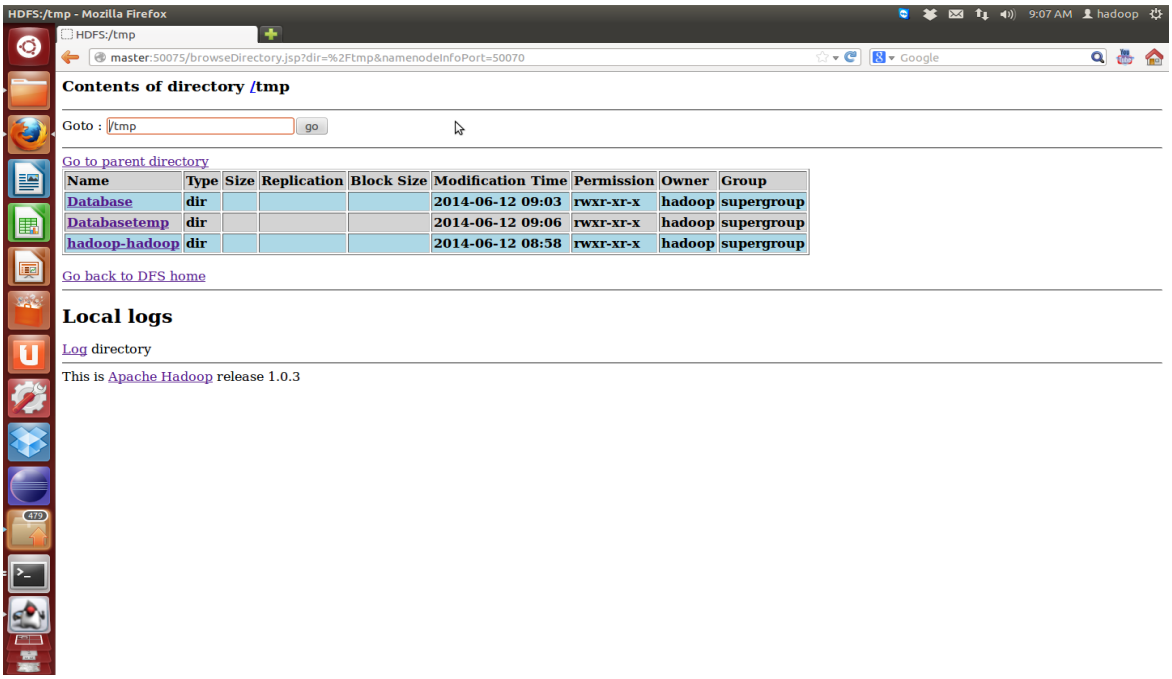


Figure 5.2.2: Hadoop Distributed File System (HDFS) database

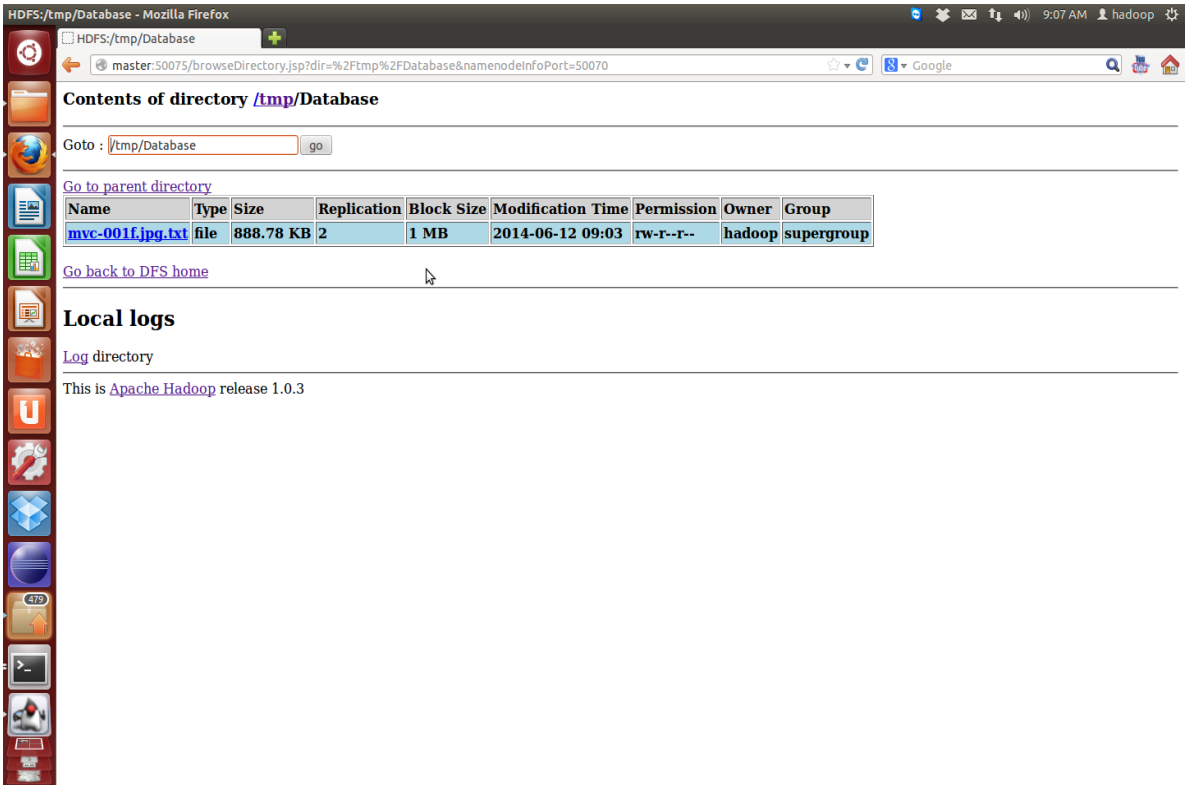


Figure 5.2.3: Hadoop Distributed File System (HDFS) Displaying Image Data

Figure 5.2.1, 5.2.2 and 5.2.3 describe how actually data is stored in HDFS database. In the implemented system, HDFS is used for storing images. In HDFS storage, data is stored in the format of text. Data is broken down into smaller pieces (called blocks) and these chunks are distributed throughout the Hadoop cluster. Hadoop provides us the facility to read/write binary files. As a result, anything which can be converted into bytes can be stored into HDFS. This provides the scalability that is desirable for large data processing.

5.3 Storing Images Securely on Database

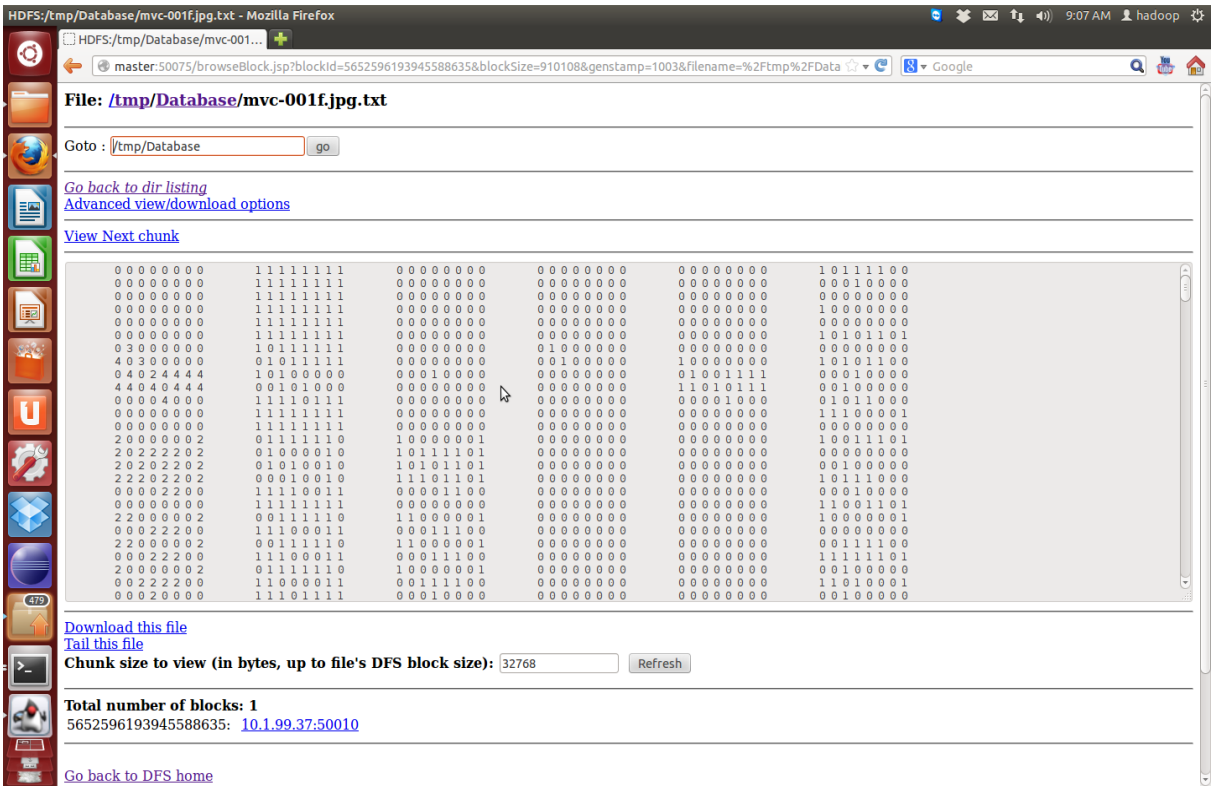


Figure 5.3.1: Image Content on Hadoop Distributed File System (HDFS)

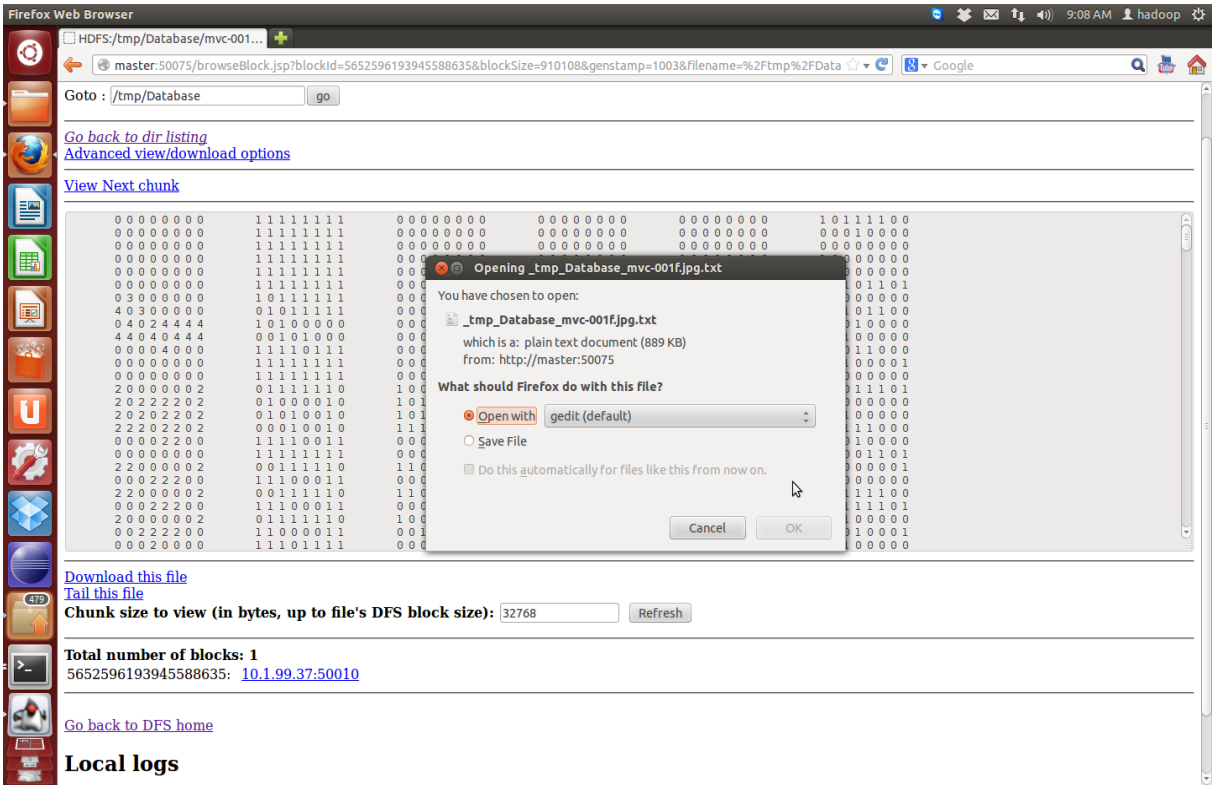


Figure 5.3.2: Downloading File from Hadoop Distributed File System (HDFS)

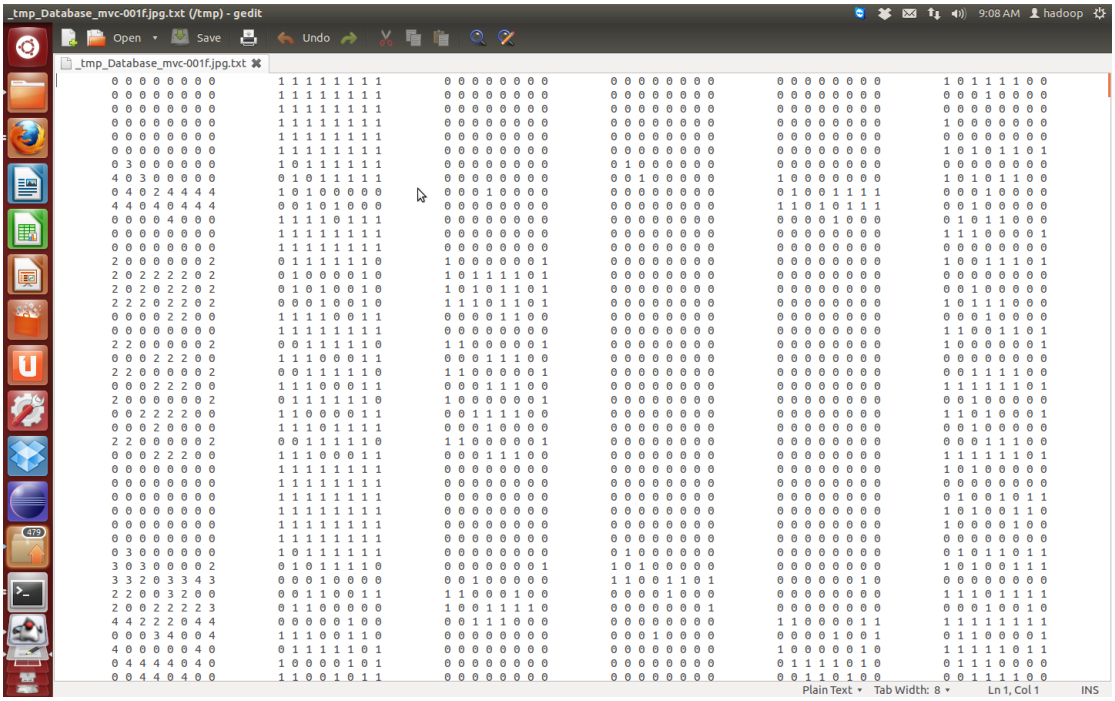


Figure 5.3.3: After downloading Image File

The image data which is stored on the HDFS is in the form of text file and these text files do not reveal any information about images even to the Hadoop database admin. The images cannot be distinguished from one another on the basis of size i.e. all images appear to be of same size on server. Moreover, the data of images is in binary file. As such, the images cannot be seen. Consider a scenario where an admin wishes to access some images. He might search through all the data but no information about the images will be available from the HDFS datasets. The procedure is shown, step by step, in figures 5.3.1, 5.3.2 and 5.3.3.

Algorithm 5.1 Upload Image by LTrP Pattern

Input: Query image; Retrieval result

1. Rescale the image.
2. Convert it into gray scale
3. Apply the calculation of the direction of every pixel.
4. Divide the pattern into four parts based on the direction of the center pixel.
5. Calculate the tetra patterns and separate them into three binary patterns.
6. Save into Database.

Algorithm 5.2 Upload image by color

Input: Query image; Retrieval result

1. Rescale the image.
2. Calculate Image feature Extraction (Color Histogram (RGB)) values of each Image.
3. Encryption of color values.
4. Save into database.

Algorithm 5.3 Search Image by LTrP Pattern

1. Upload the image for search from user.
2. Rescale the image.
3. Convert it into gray scale
4. Apply the calculation of the direction of every pixel.
5. Divide the pattern into four parts based on the direction of the center pixel.
6. Calculate the tetra patterns and Separate them into three binary patterns.
7. Compare the query image with the images in the Database.
8. Result.

Algorithm 5.4 Search Image by Color

Input: Query image; Retrieval result

1. Rescale the image.
2. Calculate Image feature Extraction (Color Histogram (RGB)) values of each Image.
3. Encrypt these values.
4. Compare the query image with the Database on the encrypted color values.
5. Result.

5.4 Search Process

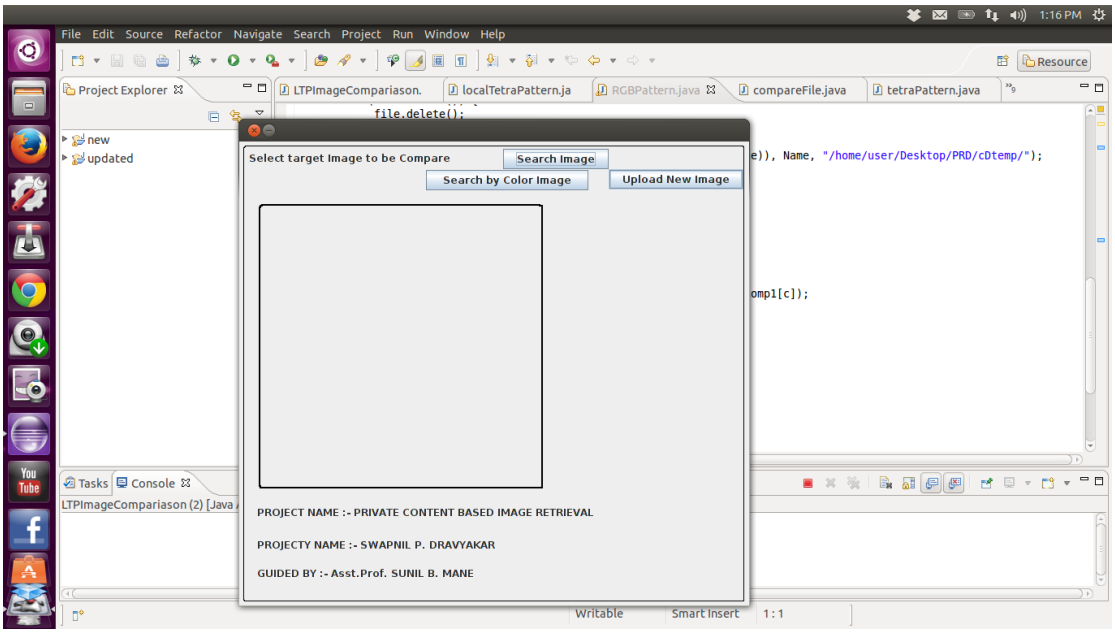


Figure 5.4.1: Search Process

Search process is also same as upload process which is shown in figure 5.4.1. After running the .jar file on Hadoop cluster, it generates the GUI shown in figure 5.4.1. In this interface, path for uploading the query image is provided. The query image is then compared in the database of Hadoop Distributed File System (HDFS).

CHAPTER 6

Experiments, Testing & Result Analysis

Initially the experiments are carried on the small set of images for evaluation and testing. The code was run on this set and all operations were performed on stored .txt files. This system allows images of a variety of sizes and formats. For testing and results, a face dataset has been used. The images, in the dataset, are in .jpg format and have the dimension of 640 by 480 pixels.

Experiments are carried out at upload and search stage with repository of images. These images are stored data size are 1, 10, 50….100 and then 200, 300, 400, 500, 1000, 1500.

6.1 Cluster Configuration

Configuration of each node and the role it plays in the cluster is shown in Table. 6.1. Master node handles all major processes. These are Namenode process, Datanode process, Jobtracker, Tasktracker and secondary namenode process and HMaster process of Hbase. Slave nodes have Datanode, Jobtracker and Tasktracker running. A small cluster of three nodes was also created in order to check performance issue.

Nodes	Description	Role	Memory Gb	CPU	CPU Freq. GHz	Disk Gb	OS
1	Desktop Workstation	Master	4	Dual Core AMD Optron	1.8	48	Ubuntu 12.04
2	Desktop Workstation	Slave1	2	Dual Core AMD Optron	2.6	50	Ubuntu 12.04
3	Desktop Workstation	Slave2	2	Dual Core AMD Optron	2	60	Ubuntu 12.04

Table 6.1: Cluster Configuration

6.2 Result Analysis

Experiments are carried out on Hadoop cluster which has one master PC and two slave PCs. Each of the PC has different processor, RAM and disk storage. The data to be stored on the HDFS are inputted with the help of GUI. We have tested uploading time for images from one image to 1500 images at one point and search a particular image on database. The results of the same are shown in Table 6.2. The time taken for upload and search are measured in milliseconds. For better result of upload and search image on HDFS datasets needs the good internet connection for communication between master and all slave PCs.

No. of Images	Upload time	Search time by LTrP	Search time by color
1	2567	1303	1533
10	12535	1403	3456
50	44851	1642	6886
100	102247	3070	7892
200	215503	4461	9678
300	270928	5651	107864
400	347151	2151	166776
500	432230	3027	187769
1000	832269	5682	207786
1500	1304653	8502	256678

Table 6.2: Difference of Uploading and Searching Time

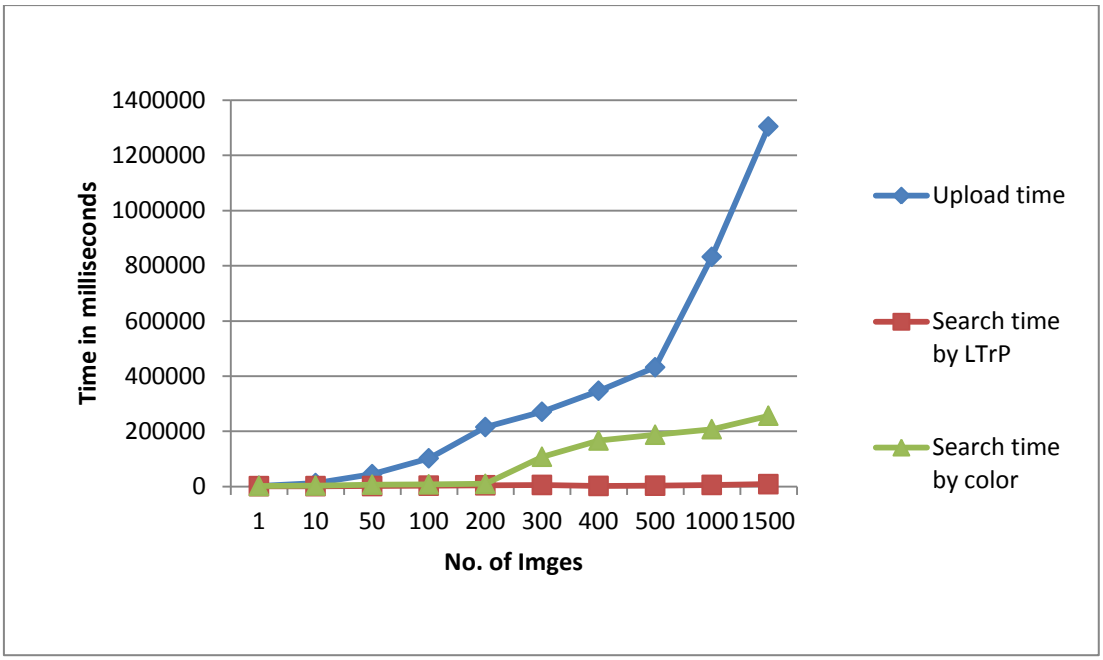


Figure 6.1.1: Graph of Difference between Uploading and Searching Time

In Figure 6.1.1, difference of uploading and searching time. The graph shows the total images uploaded at a single point on a system and searching the images with respect to uploaded measured time in milliseconds. The results clearly show LTrP search time is less than Color Histogram search.

6.3 Comparison of Different Systems

Table 6.3 shows comparison of current system with the existing techniques based on some major criteria.

System Implemented	Security	Upload Process	Color Histogram	LTrP Method	Metadata Search
Ostensive Model(OM)	No	No	Yes	No	No
FIRE	No	No	Yes	No	No
Triangle Inequality Algorithm(TIA)	No	No	-	-	No
imgseek	No	No	Yes	No	No
My system	Yes	Yes	Yes	Yes	No

Table 6.3: Comparison of different Systems

CHAPTER 7

SYSTEM OUTPUT

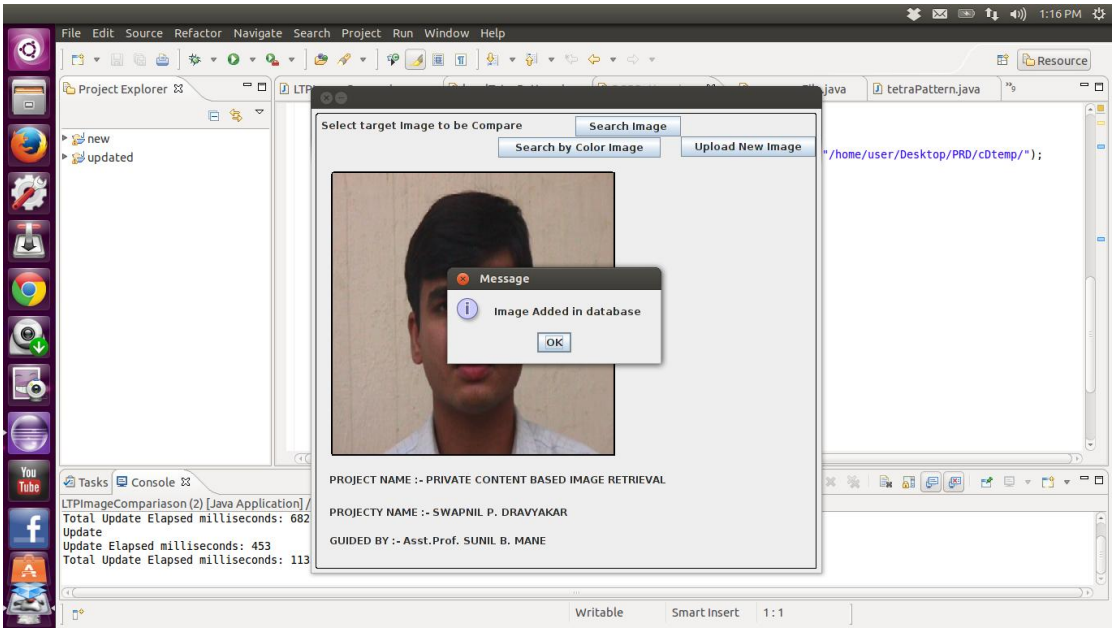


Figure 7.1: System output For Upload

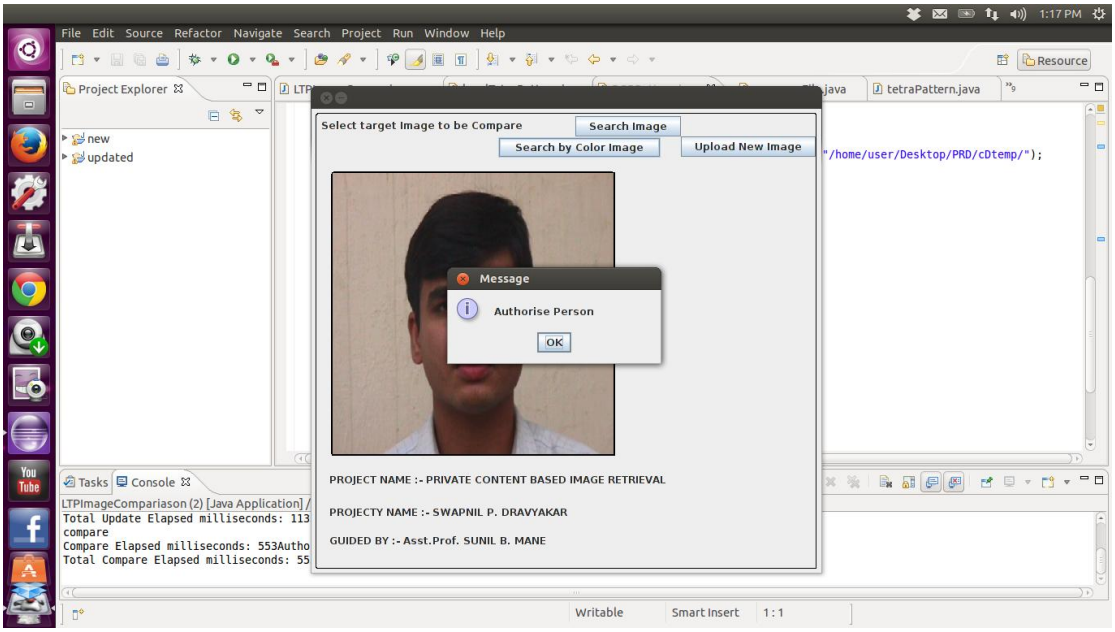


Figure 7.2: System output For Compare

Figures 7.1 and 7.2 show the input and final output. As discussed before, same processes are done for Upload and Search. For testing, the images are uploaded in the database in the same file format i.e. .jpg. Search image is compared with all database

images on the basis of the pattern and color histogram values in .txt file that are stored on the database after computation. The result is provided in time of the order of few milliseconds.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

We experimented and evaluated the proposed system with four aspects. One is uploading, second is search query image by LTrP Method, third is the query image search by the color values and fourth is to store all encrypted image value to database which provides security.

Images are growing through the various digital devices and these images are added to the image databases and internet for various applications. These images need to be stored and retrieved in effective and efficient manner. The searching time is the most important for any search method while searching it in large datasets of images.

For these purposes, we described the novel method LTrPs and color histogram for PCBIR of large datasets incorporated with HDFS file system of Hadoop framework.

8.2 Future Scope

This application is designed by combining a number of different domains into one. After implementing this system, we find that there is wide scope for different feature extractions like shape and for secure storage of images on database using different encryption techniques. This application has large scope in cloud domain and internet world because every domain servers are using cloud technology and they want to provide their customers with new cutting edge applications for image storage and retrieval which are secure, efficient and are delivered with high throughput.

Appendix A

Publication Status

Title	Journal	Status
Private Content Based Image Retrieval on Map-Reduce	International Journal of Computer Science Engineering and Technology	Published

REFERENCES

[1] J. Shashank, "Content Based Image Retrieval Using Color, Texture and Shape Features*," International Conference on Advanced Computing and Communications, ADCOM* , pp.780 – 784,2007.

[2] Shankar M. Patil "Content Based Image Retrieval Using Color, Texture and Shape," *International Journal of Computer Science & Engineering Technology (IJCSET)*, Vol. 3, Sept. 2012.

[3] Sitalakhmi and S. kulkarni "MapReduce neural network framework for efficient content based image retrieval from large datasets in the cloud" *IEEE Transactions on Hybrid Intelligent Systems*, pp. 63 – 68,2012.

[4] Ryszard S. Choras "Image Feature Extraction Techniques and Their Applications for CBIR and Biometrics Systems" *International Journal of Biology and Biomedical Engineering*, Vol. 1, 2007.

[5] Murala S., Maheshwari R.P., and Balasubramanian R., "Local Tetra Patterns: A New Feature Descriptor for Content-Based Image Retrieval," *IEEE Transactions on Image Processing*, Vol. 21,pp.2874 – 2886, May 2012.

[6] Khan, S.M.H. , Hussain, A. , and Alshaikhli, I.F.T."Comparative Study on Content-Based Image Retrieval (CBIR)" *International Conference on Advanced Computer Science Applications and Technologies* (ACSAT), 2012 .

[7] Oberoi Ashish, Bakshi Varun, Sharma Rohini and Singh Manpreet "A Framework for Medical Image Retrieval Using Local Tetra Pattern," *International Journal of Engineering Science & Technology*, Vol. 5, pp.27, Feb2013.

[8] Liangliang Shi , Bin Wu ,Bai Wang and Xuguang Yan "Map/reduce in CBIR application," *International Conference on Computer Science and Network Technology (ICCSNT),* Vol.4 , pp. 2465 – 2468, Dec. 2011.

[9] Chapter 7.Textures [Online]. Available http://csweb.cs.wfu.edu/

[10] Muneto Yamamoto and Kunihiko Kaneko, "Parallel Image Database Processing With MapReduce And Performance Evaluation In Pseudo Distributed Mode,"

International Journal of Electronic Commerce Studies,Vol.3, No.2, pp.211-228, 2012.

[11] Apache hadoop. [Online]. Available: http://hadoop.apache.org/

[12] Mapreduce - hadoop wiki. [Online]. Available: http://wiki.apache.org/ hadoop/MapReduce.

[13] Hdfs users guide. [Online]. Available: http://hadoop.apache.org/docs/ hdfs/current/hdfs user guide.html.

[14] Apache hbase. [Online]. Available: http://hbase.apache.org/

[15] P. R. Sabbu, U. Ganugula, S. Kannan, and B. Bezawada, "An oblivious imageretrieval protocol,*" in Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, ser. WAINA'11. Washington, DC, USA: IEEE Computer Society*, pp. 349{354},2011.

[16] Y. Rui and T. S. Huang, "Image retrieval: Current techniques, promising directions and open issues," *Journal of Visual Communication and Image Representation*,vol. 10, pp. 39{62, 1999.

[17] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani,J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query by image and video content: the QBIC system," Computer, vol. 28, no. 9, pp. 23{32, 1995.

[18] C.-C. Chen and H.-T. Chu, "Similarity measurement between images,"*in Proceedings of the 29th annual international conference on Computer software and applications conference, ser. COMPSAC-W'05. Washington,DC,* USA: IEEE Computer Society, pp. 41{42, 2005.

[19] C. W. Niblack, R. Barber, W. Equitz, M. D. Flickner, E. H. Glasman, D. Petkovic,P. Yanker, C. Faloutsos, and G. Taubin, "*QBIC project: querying images by content, using color, texture, and shape*," C. W. Niblack, Ed., vol. 1908, no. 1.SPIE, pp. 173{187, 1993.

[20] P. R. Sabbu, U. Ganugula, S. Kannan, and B. Bezawada, "An oblivious image retrieval protocol,*" in Proceedings of the 2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications, ser. WAINA'11. Washington, DC, USA: IEEE Computer Society*, 2011,

[21] J. Zhang, X. Liu, J. Luo, and B. Lang, “Dirs: Distributed image retrieval system based on mapreduce," *in Pervasive Computing and Applications (ICPCA*), 2010 5th International Conference on, 2010.