Searching Technique for Near Exact Duplicate Images in Cloud Database

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By
Maneesha
(801431012)

Under the supervision of:
Dr. Inderveer Chana
Professor, CSED

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

July 2016

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "*Searching Technique for Near Exact Duplicate Images in Cloud Database*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Dr. Inderveer Chana* and refers other researcher's works which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Maneesha

(Maneesha)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

11/07/16

(Dr. Inderveer Chana)
Professor
Computer Science and Engineering Department
Thapar University,Patiala

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S.S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

i

Acknowledgement

No volume of words is enough to express my gratitude towards my guide, **Dr. Inderveer Chana,** Professor, Computer Science and Engineering Department, Thapar University, Patiala, who has been very concerned and has aided for all the materials essential for the preparation of this thesis report. She has helped me to explore this vast field in an organized manner and provided me with all the ideas on how to work towards a research oriented venture.

I am also thankful to **Dr. Maninder Singh**, Head of Computer Science and Engineering Department and **Dr. Rupali Bhardwaj**, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work.

I would also like to thank the staff members and my colleagues who were always there in the need of the hour and provided with all the help and facilities, which I required, for the completion of my thesis work.

Most importantly, I would like to thank my parents, friends and the almighty for showing me the right direction out of the blue, to help me stay calm in the oddest of the times and keep moving even at times when there was no hope.

Maneesha
(801431012)

Abstract

Present technologies have made image capturing cameras available to everyone and easy access to internet. Internet based services and high popularity of the social networking sites that are used for the sharing the picture has resulted in a large quantity of images shared on the internet. The availability of the cloud storage allowed the individual and companies to keep and maintain the huge collection of images. Some of them are the near exact duplicate images. Near exact duplicate images are the images which are generated after applying some modification such as cropping, rotation etc. These images are stored with the original images, so one image is stored multiple times. Traditional database are not fit for these types of images because they are unable to access the information within the images. These were intended to deal with text based structure and store the information about image like metadata, resolution. They store the image in files but they cannot perform image processing, classification and image matching based on the information stored in images.

In this thesis techniques of the feature detection, feature descriptor and nearest neighbour are discussed. The system is designed for implementation of searching near exact duplicate images in cloud database. MATLAB 2015a is used for implementation of proposed solution. Here Binary Robust Invariant Scalable Keypoints (BRISK) is used as a feature descriptor which is a binary descriptor and for indexing Locality sensitive hashing is used. This allows the user to issue a query image and then search all the near exact duplicate images related to query image. Experiment results show that it effectively searches near exact duplicate images. The performance of the proposed solution is discussed by using precision and recall.

Table of Contents

List of Figures

List of Tables

List of abbreviations

BRISK-Binary Robust invariant and Scalable keypoints

BRIEF- Binary Robust Independent Elementary Features

FAST- Features from Accelerated Segment Test

LSH-Locality Sensitive Hashing

ORB-Oriented Fast Rotated BRIEF

SIFT- Scale Invariant Feature Transform

SURF- Speed up Robust Feature

GUI- Graphical User Interface

CHAPTER 1 Introduction

This chapter discusses about the cloud computing, application of cloud computing, rise of unstructured data in cloud, research issues in cloud, motivation of research and the organization of the thesis.

1.1 Introduction to Cloud Computing

In cloud computing 'cloud' refers to an IT environment which is designed for provisioning scalable and measured IT resources remotely. Cloud originated as metaphor for internet. It can be define as a network of networks that offers to use the decentralized IT resources remotely.

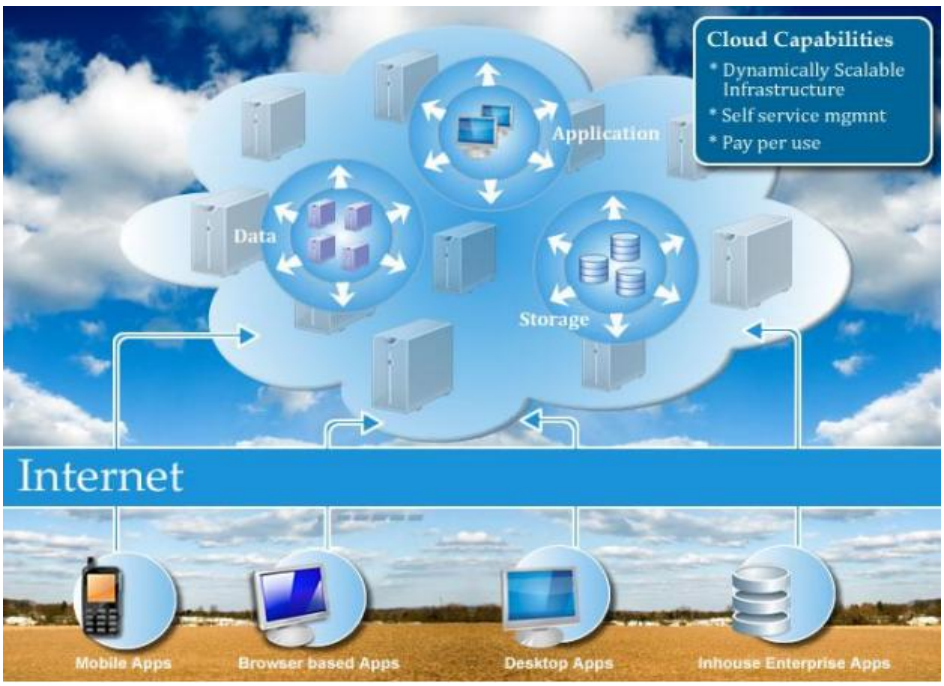

Figure 1.1: Conceptual view of cloud computing [2]

Cloud computing is computing based on internet which provides the ability to process the shared resource and data to the device like computer, mobile on demand. It has evolved after many phases that include grid and utility computing, Application Service Provision (ASP) and Software as a service (SaaS). Cloud computing term is based on a single element: delivering the computing services over the internet, from the remote location based on the demand instead of residing on own laptop, desktop, on organization server or mobile device [3].

Various definitions are given by many researchers as follows:

- According to NIST [1], "cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.".
- Michael Armbrust et al. [9], defined the cloud computing as it delivers the application as a service on the Internet as well as it contains hardware and software on the data centers for providing these services.
- According to Amazon [4], "Cloud computing is the on-demand delivery of IT resources and applications via the Internet with pay-as-you-go pricing. Whether you run applications that share photos to millions of mobile users or you support the critical operations of your business, the cloud provides rapid access to flexible and low-cost IT resources".

1.1.1 Service Model of Cloud Computing

There are three service model of the cloud computing Figure 1.2.

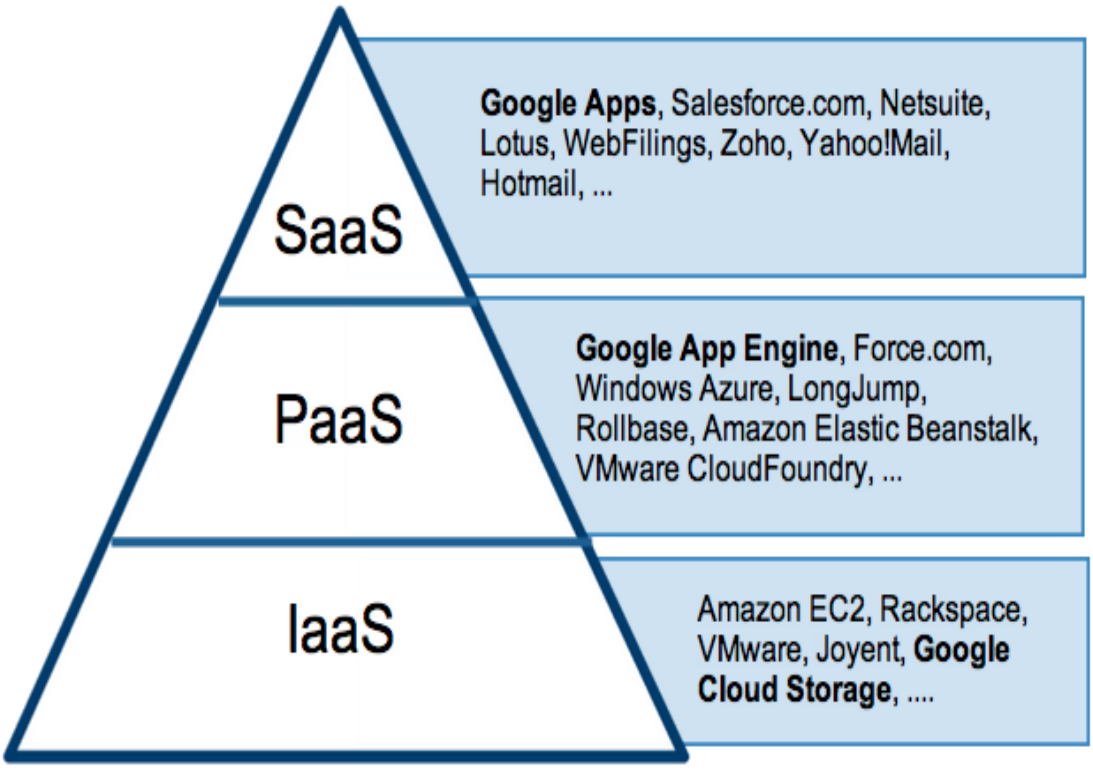

Figure 1.2: Service models of cloud computing [5]

- **Infrastructure as a service:** Infrastructure as a service is the main service model of cloud computing. It a type of cloud computing which provide the Internet based virtualized computing resources. In the IaaS service model, there is a third party provider who hosts the storage, hardware, software and server etc. IaaS provider also host the application of the user and handle the task such as maintenance and back up. IaaS is used by the enterprises customer for creating the easily scalable and cost effective IT solution where expenses and complexities are there for managing the hardware. IaaS has the features and benefits such as scalability of resources, location independence; no investment in hardware is required, physical security of data centre location and no single point of failure.
- **Platform as a service:** Platform as a service is another service model of cloud computing which gives a platform to the customer to develop, run and manage the application without considering what complexities and maintenance is required for the infrastructure during the development and launch of application. PaaS delivers the services in two ways. First is public cloud service from the service provider in which consumer have control on the development of software but its minimal and service provider provide the servers, networks, storages and host the application of customer ; second is as a private service which is inside firewall.
- **Software as a service:** Software as service is also known as the "on demand software" [6]. It is a software distribution model where applications are hosted by third party provider and then makes these applications accessible to the customer on the Internet. SaaS provide a way in which organization need not to install and run the application on their computer. It completely reduces the cost for the acquisition, provision and maintenance of the hardware. The payment of this service is on a monthly basis using pay as you go model. Benefit of SaaS are as follow:
  - Flexible payment
  - Scalable usage
  - Automatic updates
  - Accessibility and persistence

1.1.2 Deployment Model of Cloud Computing

Deployment models represent the categories of cloud environment and these are differentiated by size, access and priority. Theses model describes the purpose and

characteristics of the cloud. There are four deployment model of cloud computing shown in Figure 1.3:

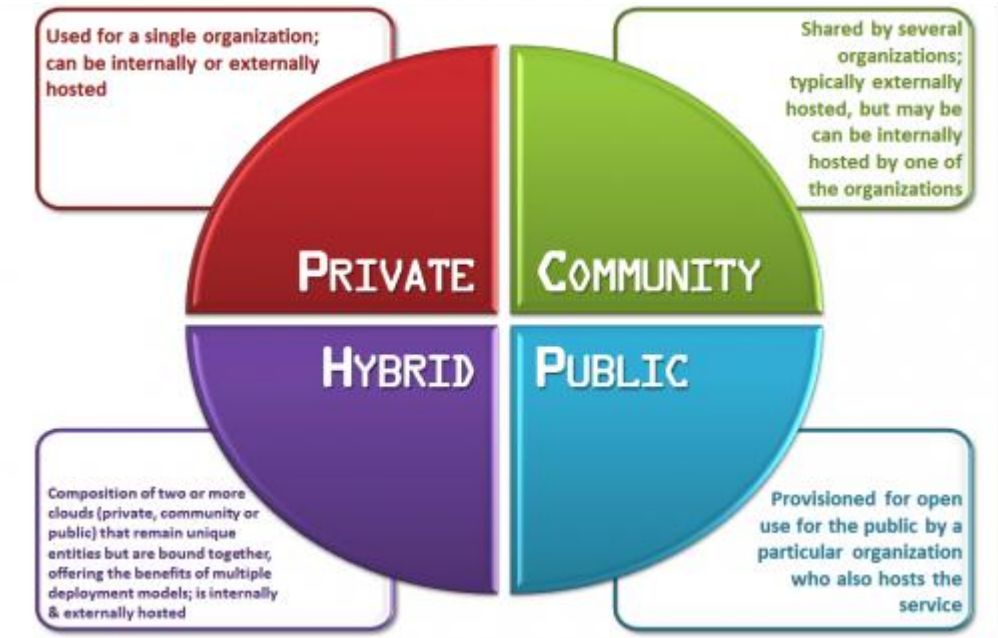

Figure 1.3: Deployment models of cloud computing [7]

- **Public cloud:** Public cloud is the cloud model where resources such as storage, application and servers are provided to the public over the Internet. In this service providers make these resources available to public. It is either according to pay per usage or free. Public cloud uses the services of the cloud computing that help the customers who are not part of the provider's organization. Here customer does not have any control over the infrastructure location. The cost in the public cloud is distributed among all users, so customers get profit from public clouds. Examples of public clouds are Windows Azure Services Platform., Google AppEngine and Amazon Elastic Compute Cloud (EC2) etc. The advantages of public cloud are scalability, increased reliability and cost effectiveness. Disadvantages of public cloud are data security and privacy.
- **Private cloud:** Private cloud provides services to a single organization. It is similar to public cloud but in public cloud**,** service provider delivers the service to many organizations but in private cloud it is indented for only one organization. It is also known as internal cloud. It involves an individual and secure cloud based environment where only a particular customer can work. It is similar to the traditional model for Local Network Access (LAN) for individuals which has been used by the

organizations in past but it includes the concept of virtualization. The benefits of the private cloud are that it has high security and privacy; customers have control over the cloud for configuring and managing the applications.

- **Hybrid cloud:** Hybrid cloud is a cloud computing environment which is built by integrating two clouds. It gives flexibility to the business and provides more option in data deployment because workload distributed between the clouds when needs computing and cost gets affected. Hybrid cloud is very useful in the environment where the workload is flexible. Multiple models can be deployed at one time in the hybrid model. Organization can use it for the big data processing.
- **Community cloud:** Community cloud is the cloud computing environment which is shared among numerous organizations that are included in specific community like trading firm and bank etc. with some concern like compliance, jurisdiction and security etc. The main aim for community cloud is to accomplish the objective of business. It can be managed by a third party provider or can be managed internally. Also it can be either hosted internally or externally. Cost is distributed among the organizations under that community so it has capacity of saving the cost. Advantages of community cloud include ability to share and collaborate easily and it has lower cost. On the contrary, community clouds slowly adapt to updates and are not always the right choice for every organization

1.2 Cloud Computing Applications

Cloud computing has become popular over the past year due to cost reduction, elasticity, flexibility and resource utilization. Here are some situations where this is used for enhancing the ability to achieve various goals.

- Clients would be able to access their data and application remotely at any time. They use the internet for accessing the cloud computing from computer or any devices.
- Cloud computing cut down the cost required for hardware. It eliminates the requirements of hardware on client side. No need to buy computer with great memory because cloud provide this memory as much client requirements. In cloud computing only basic devices are required such as monitor, keyboard etc. Internet connection is necessary for accessing the cloud system.
- Cloud computing reshapes the education sector. The traditional classroom technique is replaced by cloud exercise like smart classes using auditory illustrations and

pictorial representations. It has enabled remote access of educational material which makes learning easy for everyone.

- Industries are empowered by cloud as it solves the business problem faced by industries in executing their own data center. It also saves money.Cloud computing allows the industries to increase their resources. Cloud computing provides an easy management of industries data and records. It improves the quality of service of industries because now industries do not require purchasing the software and hardware.
- In medical field, cloud computing provides a platform to medical professional for securing the patient's information so that it can be accessed from anywhere without going to hospital's computer. It enables the professionals to update the patient's information even if they are not in hospitals.
- Now all the banking companies have been automated by using cloud services. But due to data security issues adoption of cloud is relatively low.
- Online cloud storage used by the people to store and share their photos. They provide free storage upto some limit. From customer's point of view the key major advantage of cloud storage is that it enables the customers to reduce expenses for procuring and maintaining storage. Cloud service providers provide a way to pay for the amount of storage customer required that can be increase or decrease according to the customer's requirement.

1.3 Research Issues in Cloud computing

In today's world, Cloud computing has a great impact on the IT industry because of services provided by cloud computing. There are various research issues and challenges of cloud computing that are related to the data security, performance, and design issues [8] [9] [10]. Some of them are discussed below:

- Virtual Machine Migration

In cloud computing virtualization provides help for load balancing among data center by enabling the virtual machine migration. Migration of the operating system among the hosts is beneficial to data centers and clusters. High performance and minimum service downtime are achieved by the operating system migration [11]. The main advantage of virtual machine migration is to avoid the hotspot. Disaster recovery is also one advantage of the virtual machine migration.

- Server Consolidation

Server consolidation is an approach for effective resource utilization in respect to reduce the number of servers required by the organizations. Server consolidation provides a facility to the applications and services to share a server's resources simultaneously. The main goal of server consolidation is to use all available resources of the server and reduce the cost and operational expenses related with multiple servers. The application performance should not be degraded by the server consolidation. The benefits of the server consolidation include flexibility to transfer the workload among the server, higher computing efficiency and energy management.

- Data Security

Security threats are faced by cloud user from outside as well as inside the cloud. Most of the threats issues include the protection of cloud from outside threats which are same as those already facing data centers. However, in cloud tasks are divided among the many parties such as cloud user, third party and cloud vendor. It is very difficult to provide security at each layer in cloud.

- Energy Management

Energy management is a major issue in the cloud computing. Data centers have large number of servers with storage and hardware. The results of the data center are high operational expenses and emission of the carbon dioxide [12]. In United State "It takes 34 power plants, each capable of generating 500 megawatts of electricity, to power all the data centers in operation today. By 2020, the USA will need another 17 similarly sized power plants to meet projected data center energy demands as economic activity becomes increasingly digital"[13]. The need of data center energy is increasing because of demand for the computational power increment. The increment in the electricity used by the datacenters saw 56% increased between 2005 and 2010 [14]. So there is need to implement energy measures to prevent too much usage in future.

- Automatic Service Provisioning

The aim of service provider is to allocate and de-allocate the resource from cloud for satisfying the service level objective and decreasing the operational cost. Approaching typically involve: (a) Construct application performance model which forecast the number of instance required for handling the demand at every level that will satisfy the QoS requirements; (b) Periodically predict the demand which can be arise in future and find out resource requirements by performance model.; and (c) Automatically resource

allocation according to predicted resource requirement. Reactive approach works on immediate demand variation that is available without prediction of periodic demand. In proactive approach, calculate the future demand of resource before they are needed.

- Cloud Storage

Cloud backup is becoming the major factor of the cloud storage because of the emerging trend of user data. As the amount of data increased in data center, more bandwidth and storage space required. The benefits of cloud storage are flexibility, no infrastructure cost, disaster recovery and data can be easily accessed from any compatible device remotely. According to IDC, on cloud 20 percent data is structure data and 80 percent is unstructured data. It creates challenge for the data center to manage the data because as data increase the cost of maintenance and back up is also increased. As cloud service providers provide some free storage to people for storing their pictures online. Most of pictures are the duplicate or near duplicate images so cloud providers face the problem to manage them and their replicas. So it is a challenge to service provider to detect these images and delete. These require more bandwidth and more storage space. A survey on cloud storage providers is carried by Fixya[46]. The report indicates the security, storage and other issues for the public cloud service providers. In Fixya report five main issues of Dropbox are shown in Figure 1.4.

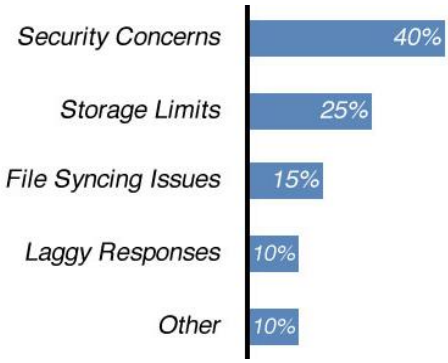

Figure 1.4: Issues in Dropbox[46]

1.4 Rise of Unstructured Data in Cloud

There is continuous increment in volume and detail of data captured by organization such as rise of Internet of Things (IoT), social media, multimedia has produced a huge amount of data. This data can be in any format i.e. structured data, unstructured data or semi structured data. These three types of data format are discussed below:

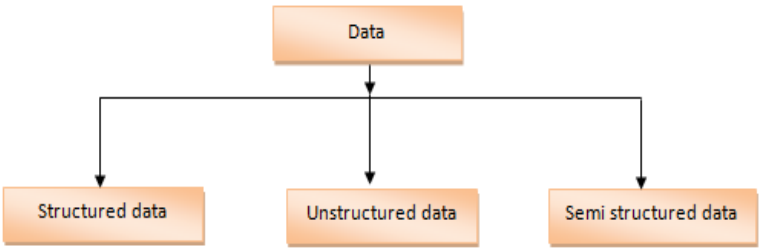

Figure 1.5: Data formats

- **Structured data**

This data format contains high degree of the organization. It is usually text file, relational database which has rows and column that are easy to organized and processed. Structured data has advantages of easy to entered, stored, queried, and analyzed. Structured data is machine readable and pre-defined. Examples of structured data are customer data, sensor data, web-log data, financial data and sales data etc.

- **Unstructured data**

It is the information which has no pre-defined structure or cannot be organized in pre-defined manner. These are not easy to understand. We have to process the data to understand it. Twenty percent of data in enterprise is structured data and other eighty percent is unstructured data. Machine generated data and human generated data are types of unstructured data. Examples of machine generated data are Satellite images, scientific data, Photographs, video and Radar data etc. Examples of human generated data are text, Social media data, Mobile data and website content.

- **Semi Structured data**

It is the information which cannot be stored in relational database but has some organizational properties which makes it easy to analyze. Semi-structured data can be stored data in relational database by using some processes. NoSQL databases, XML, email are examples of the semi-structured data.

According to IDC there is 20 percent of the total data is structured data and rest 80 percent is unstructured data which is growing exponentially.

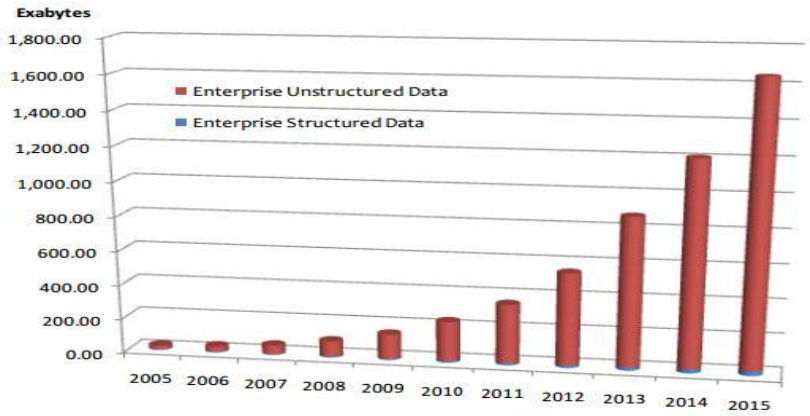

Figure 1.6: Ratio of unstuctured and structured data

Data storage and mining technologies allow for preservation of large amount of data in the format defined by the organizations. A major challenge for the practitioners and researcher is that growth rate of unstructured data exceeds their ability to preserve data in cloud computing platforms to analysis data. Images on the cloud are increased because people store their photos in cloud. Most of these images are near exact duplicate images because cloud provides a free storage upto some limit. After that limit user have to pay what storage they are using.

1.4.1 Types of Duplicate Images in Cloud

Due to the advancement of cloud storage techniques people are storing their personal photos on cloud. Most of the photos are duplicate photos. So cloud storage comes with problem of rise of duplicate images on cloud. Types of duplicate images presented in cloud are discussed below:

- **Exact duplicate Images**

Exact duplicates are the images which are having exactly same appearance. There is no transformation on the images performed, so these are the replica of the images.

Figure 1.7: Example of exact duplicate images

- **Global duplicate images**

Global duplicate images are generated after changing some content in the original images.

Figure 1.8: Examples of global duplicate image

- **Near duplicate image**

Near duplicate images are produced by capturing independent photos of the same thing under various conditions in resolution, illumination and so on. Also they can be formed by modifying the images using few transformations such as rotation and scaling.

Figure 1.9:Examples of near duplicate images

- **Near exact duplicate images**

Near exact duplicate image are the image that are generated by after applying some transformation like cropping, scaling, rotating by some angle, changing color etc.

Figure 1.10: Examples of near exact duplicate images

1.5 Research Motivation

With the rapid development of the Internet technology and availability of the image capturing device and image editing software, the numbers of images on the internet has increased. Due to the advancement, innovation and filtration of technology in the present electronics devices has led to change the way in which visual data is captured, stored and used. There have been many changes in area of image capture; the first thing is efficiency of the capturing device such as cameras, Smartphone, which lead to their incorporation in existing and rising technologies. Secondly, the capacity of the storage devices has increased with tumbling price. Thirdly, digitalization process for records has become common almost everywhere because of easy translation into digital format and storage. Fourthly, emergence of low charges of Internet access and its worldwide availability provide capability to the people to share photos. So in cloud, enormous data is increasing. Cloud storage providers face the problem of increment in storage cost and maintenance cost. Energy consumption to access these duplicate images is also increased. So to deal with these problems there is requirement to develop a system that searches near exact duplicate images in cloud. After searching the near exact duplicate images data deduplication can be apply which deletes these duplicate images.

1.6 Organization of thesis

This chapter discussed the introduction to thesis. The rest of thesis is organized in following way:

Chapter 2 Literature Review – This chapter includes the various techniques for the feature detection and feature description. The study is focused on the digital image processing and nearest neighbor search. It discussed how researchers combined these

techniques for detecting near duplicate images. Further it compares the various near duplicate images detection techniques.

Chapter 3 Problem Statement – This chapter discusses about the research gap analysis. This chapter includes the problem statement of thesis; the objective of the thesis and what methodology is used to solve the problem.

Chapter 4 Design Techniques – This chapter discusses design techniques used to solve the problem stated in previous chapter. In this, design techniques are discussed in details; block diagram of the proposed solution and the steps of proposed solution are also discussed.

Chapter 5 Implementation and Results- This chapter focuses on the tool used for the implementation, the details of implementation and the experiment result of technique with the help of snapshots.

Chapter 6 Conclusion and Future Scope- This chapter discusses about the conclusions, thesis contributions and the future scope.

CHAPTER 2 Literature Review

Searching the near exact duplicate image in large database has become important as the images on the internet have increased. Many researches have been done on searching and retrieving the near duplicate images. This chapter discusses the digital image processing, nearest neighbour search algorithm and existing near duplicate image detection techniques.

2.1 Digital Image Processing in Cloud Computing

2.1.1 Digital Image Processing

Digital image processing used the computer algorithm on digital image for image processing. In this large range of algorithms on the input data and avoid many problems like noise and signal distortion in processing. It is a branch of digital signal processing. In the area of digital image processing there is more development than analogue image processing since computer systems became less pricey and have faster computation. Several image processing methods are colour correction, image resizing, colour space conversion.

Digital Image

A digital image is composed of pixels. In the image, each pixel represents colour at a particular point. Digital image is the projection of two dimensional visual data like a scene, photograph, printed text and scanned document. Images are reconstructed by measuring the colour of image at many points and then create the approximation of the image. Every pixel has been assigned a pitch value such as white, black and shades of gray or colour by using fixed number of bits .These bit represent the information stored in the image and known as bit depth of image. Usually each pixel contains 24 bits of information where 8 bit each for Red, Green and Blue (RGB) which gives 2^8(0-255) levels for each color. In 32 bit pixel, 8 bits are added for alpha or transparency of the pixels to RGB. There are various formats for the image such as jpg, bmp, png etc. So the numerical values for each pixel are stored according to the format standard.

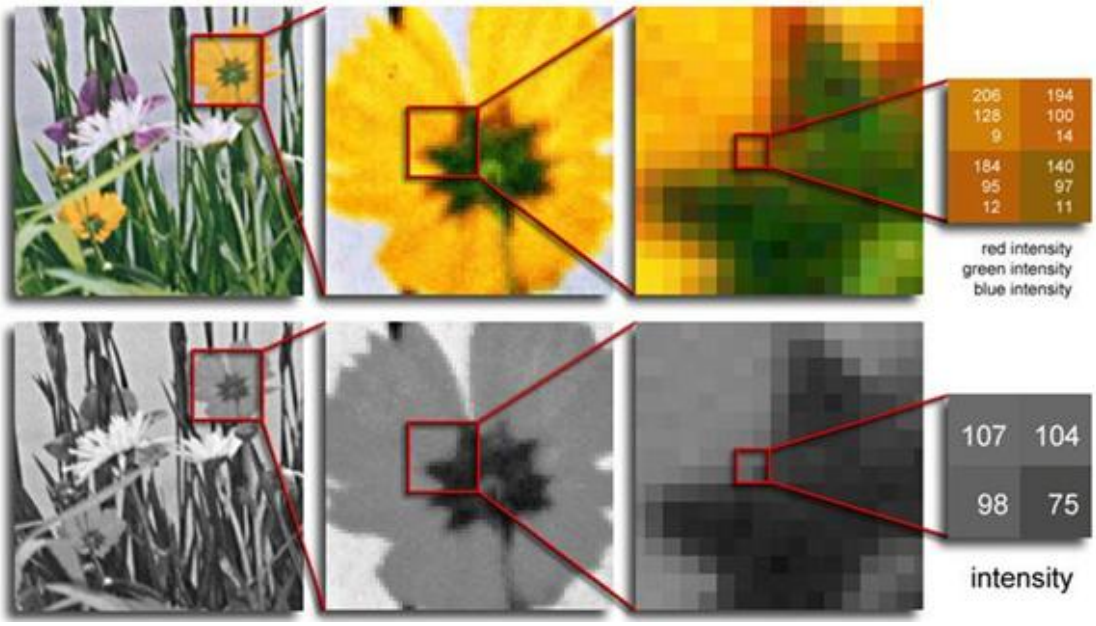

Figure 2.1: A color and grayscale image with highlighted pixel and their values [15]

There are two types of images:

i. Raster Images

ii. Vector Images

- **Raster Image**

These images stored the images in a structured format that depends on image format and displayed on the monitor bit by bit copy of stored image file. Raster images are the captured images using the camera or scanning the document to generate electronics copy of document. Raster images depend on the resolution i.e. quality and retrievability of information in image is affected by resolution. Generally "digital images" are referred to as raster image. A raster image is generally complicated to modify without any loss of information. Examples of the raster image are TIFF, JPEG, BMP and GIF.

- **Vector Image**

Vector image are not composed of pixel, these are produced by using some mathematical formulas. These are composed of the line, curve and polygons defined by mathematical equation. These mathematical equations are used to represent objects which form an image. Vector images are not resolution dependant. Vector image can be transformed, scaled and rotated in any form without loss of information. The famous storage format for the vector image is the Scalable Vector Graphics (SVG).Vector image can be easily converted into the raster images.

2.1.2 Features of Digital Images

Feature is a part of information which is used for solving the computational work for a certain application in computer vision and image processing. Feature is defined as the interesting portion of an image. Types of image features are discussed below:

- Edge: Edges have one dimensional structure. These are the point where two images connect. These are defined as sets of the points in image which have strong gradient magnitude. There are the some points at which the image intensity changes sharply; these points are ordered into a set of bent line segment which is known as edge.
- Corner: Corners have a two dimensional structure. A corner is the intersection of two edges. These are also known as "interest point" because interest points are the points which have well-defined structure. Corner feature are used in image registration, motion detection, object recognition and 3D modeling.
- Blob: Blob gives an explanation of image structure regarding regions. Blob is an area of an image where few properties are either not changed or changed little bit. In blob all the points can be considered similar to each other.
- Ridge: The idea of ridge is for the extended objects. A ridge is a one dimensional curve which represents the symmetry axis. There is a feature of local ridge width that is coupled with every ridge point. Ridge descriptors are mostly used in the medical image for blood vessels extracting.

2.1.3 Digital Image Feature Detector and Descriptor Techniques

Feature detection and descriptor are compulsory for performing the matching, indexing and recognition on image. Feature detection and feature description techniques are discussed below:

- **Feature Detection**

Feature detection is a method that computes the abstraction of the image information and produce local decision for each image point. The resultant features are the subset of the domain of the image, frequently in the form of isolated points, connected regions or continuous curves.

Some of the feature detectors are discussed here:

i. Canny: Canny edge detector is used for edge detection. It uses multistage algorithm for detecting large collection of edges in the images. Canny edge

detector was developed in 1986 by "John F. Canny" .In this firstly Gaussian filter is used to remove the noise. Then intensity gradient of image is calculated. After getting the intensity gradient, non-maximum suppression is applied to avoid false response to edge detection, and then determine potential edge by applying double threshold. At last it tracks the edge by hysteresis [16].

ii. Sobel: Sobel operator is used in edge detection algorithms where it creates an image highlighting edges. It is also called as Sobel-Feldman operator because Irwin Sobel and Gary Feldman are its developers [42].

iii. Moravec corner detection algorithm: It is one of the initial corner detection algorithms which defines corner to be point with little self similarity. This algorithm tests every pixel in image to check if any corner is present. Corner is checked by taking into account how alike a patch cantered on the pixel is close to. The similarity is estimated by taking sum of squared differences (SSD) among the corresponding pixels of two patches. More similarity is indicated by lower number. The main problem of this is that it is not isotropic [17].

iv. FAST (Features from Accelerated Segment Test): It is an algorithm for corner detection. It was developed by Edward Rosten and Tom Drummond in 2006. The main feature of FAST is the computational efficiency. It is faster than SIFT, Harris detector etc. FAST is used in real time video processing It is not good for high noise because it depends on threshold.

v. Maximally stable extremal regions: MSER is a method used for blob detection in images. The regions are defined only by extremal assets of intensity function in region and on its external boundary. MSER is basically based upon the idea of the taking regions which stay nearly identical during a broad variety of thresholds [18].

Table 2.1 discusses the comparison of above discussed feature detection techniques.

Table 2.1: Comparison of Feature Detectors

Feature detector	Feature type	Scale independent
Canny[16]	Edge	No
Sobel[42]	Edge	No
Moravec corner detection Algorithm[17]	Corner	No
FAST[41]	Corner	No
MSER[18]	Blob	Yes

- **Features Descriptor Techniques**

This is an algorithm that chooses points from image. These points are based on some principle.It takes image and output feature vectors. Feature descriptors determine interesting information into sequence of number and work as type of numerical fingerprint which can be used to make a distinction of one feature from other feature. Descriptor is a vector of values which describes image patch. These patches are around interest point. This information is invariant from image transformation.

Some of the feature descriptors are discussed below:

- SIFT (Scale Invariant Feature Transform): SIFT is an image descriptor developed by David Lowe in 1999. SIFT descriptor is invariant to transform such as rotation, translation and scaling in image domain .It is very good for moderate perspective illumination and transform variation. There are four major stages for computation to generate set of features of image. Difference-of-Gaussian (DOG) is used to detect the interest point of image. After detecting the interest point key point localization is performed in which key points are selected based on their stability. Orientation is assigned to every key point location which is based on local image gradient directions. In the last stage key point descriptor is generated. In key point descriptor local image gradients are uses at selected scale.

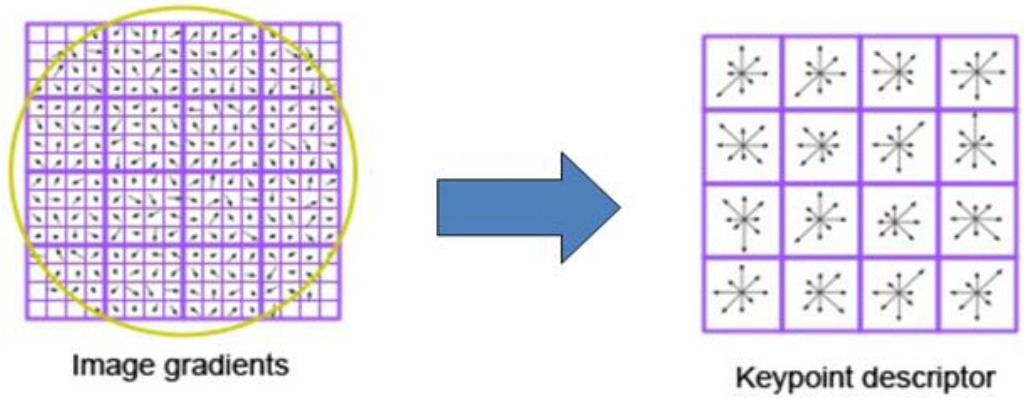

Figure 2.2: SIFT keypoint descriptor [19]

This compute histogram for local oriented gradients about the interest points and then stores bin in 128D vector (4*4*8=128 where 8 orientation bin are there for 4*4 location bins) [19].

- SURF (Speed up Robust Feature): SURF was presented by H. Bay et al. in 2006.SURF is a local feature detector and feature descriptor. For feature detection SURF use the hessian matrix which is a blob detector for interest point detection. It uses the determinant of hessian matrix which can be computed using precompiled integral image. For localize interest point in image and over scale, non-maximum suppression in 3*3*3 neighborhood applies. SURF feature descriptor is based on sum of HAAR WAVELET response about the point of interest [20].
- BRIEF (Binary Robust Independent Elementary Features): It is a binary feature descriptor. It relies on small number dissimilarity tests to signify image patch like a binary string. In BRIEF Euclidean distance is replaced by hamming distance to check similarity in descriptor. It is not intended to be rotationally invariant [21].
- ORB (Oriented fast rotated BRIEF): ORB is developed by Ethan Rublee et al. in 2011.It is very fast and robust local feature descriptor. For feature detection it uses FAST feature detector and BRIEF is used for feature description. The main aim for this is to provide speedy and proficient alternative to SIFT [22].
- FREAK (Fast retina keypoints): It is binary descriptor, presented in 2012. It is inspired by human retina .There is a sequence of binary string which is computed by comparing the image intensities over retinal sampling pattern [23].
- HOG (Histogram of oriented gradient): It is a feature descriptor which is used for object detection. This counts the occurrence of the gradient orientation in the

localized portion of image. It is based on evaluating normalized local histogram of the image gradient orientation for dense grid. It is similar to SIFT but it is computed on dense grid which has uniform spaced cells. It uses the overlapped local contrast normalization which improves the accuracy [43].

- BRISK (Binary Robust Invariant Scalable Keypoints): It is also a binary feature descriptor. It uses hamming distance for matching instead Euclidean distance. It is faster than SURF/SIFT. BRISK is invariant to transform like rotation. It is well suited for real applications and low power devices. It estimates the interesting points in continuous scale space. BRISK uses the FAST for feature detection. This descriptor is composed as a binary string by using concatenating results of simple brightness evaluation tests. BRISK descriptor uses pattern which is used for sampling neighborhood of key point. Short pair distance is used for BRISK descriptor [24].

Table 2.2: Comparison of Feature Descriptor

Feature descriptor	**Binary**	**Rotation invariant**	**Scale invariant**	**Classification**
SIFT [19]	No	No	Yes	Yes
SURF [20]	No	Yes	Yes	Yes
BRIEF [21]	Yes	No	No	No
ORB[22]	Yes	Yes	No	No
FREAK[23]	Yes	Yes	Yes	No
HOG[43]	No	No	No	Yes
BRISK[24]	Yes	Yes	Yes	No

2.2 Digital Image Indexing Techniques

2.2.1 Nearest Neighbour Search

Nearest neighbor search is an optimization problem that find the closest point. It is also known as the closest point search and similarity search. Nearest neighbor search finds the similar object in both low dimensional and high dimensional data. Nearest neighbor is a simple, effective and efficient technique in the area of text categorization, pattern recognition and object recognition. Nearest neighbor search problem is described as: let S is set of point in space M and query point Q ∈ M, then search the closest point to Q in S.

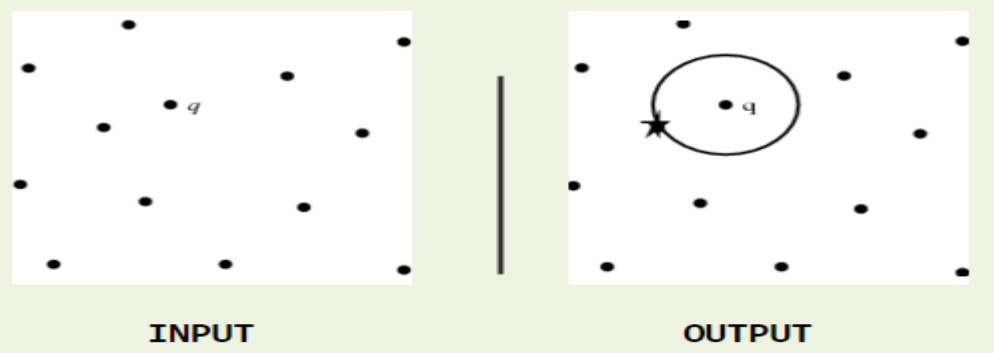

Figure 2.3: Example of nearest neighbour search

2.2.1.1 Variants of nearest neighbor search

Many variant of the NNS problem are there, some of them are given below:

- Approximate Nearest Neighbor

i. Few applications accept the good guess of nearest neighbor. This type of algorithm is used in applications where no guarantee of actual nearest neighbor.

ii. ANN is indented for the small data set so that the searching structure can be easily stored in main memory.

iii. There are many ways to show the distance between the points.

iv. In this distance between the points are calculated through any distance function. These distance functions are known as Murkowski metrics.

v. It includes Manhattan distance, Euclidean distance and max distance [25].

vi. It is not good if we want exact result of the query.

- K Nearest Neighbor

i. This is a non parametric method that is used for regression and classification.

ii. Class membership is the output of the classification in k-NN. Suppose k=1, then object assigned to class has single nearest neighbor.

iii. For k-NN output is the property of the object. The output is the average output of its all k nearest neighbor.

iv. K-NN is lazy learning because in this all the computations are delayed until classification of data.

v. It is simplest among all the algorithm of machine learning but it is very sensitive to the local data structure.

- All Nearest Neighbors

In few applications if N are data points there is a need to know which is the nearest neighbor for every one of those N points. This is achieved by performing the nearest neighbor search for every point.

2.2.1.2 Application of Nearest Neighbor Search

- Data compression
- Statistical classification
- Pattern recognition
- Recommender system
- Computer vision
- Plagiarism detection

2.2.1.3 Method of Nearest Neighbor Search

- Linear searching

i. It is the easiest solution for the nearest neighbor search problem that calculates the distance from the query point to every point present in database.

ii. This approach is also known as naïve approach and it has running time O(dN) in which d is dimensionality and N is the cardinality.

iii. It has no space complexity except the storage for database because there is no need to maintain the data structure.

- Locality Sensitive Hashing

LSH is an approach that groped the point into buckets that are based on some distance metric performed on points. The points which are near to each other are mapped to same bucket according to selected metric [26].

- Space Partitioning

i. There is various space partitioning methods for the nearest neighbor search region.

ii. K-d tree [27] is the simplest method for this which divides the search space into two regions.

iii. When request a query then it process traversal of tree from root to leaf by evaluating query point at every split.

iv. The average complexity for the constant dimension query is $O(\log N)$.

v. For randomly distributed point, the worst case complexity is $O(kN^{(1-1/k)})$.

vi. If there is dynamic context then R-tree data structure is designed.

2.3 Existing Techniques in Near Duplicate Image Detection

In [28] Yan Ke et. al. present an approach for the near duplicate image detection and retrieval of sub image. In this image is represented in parts by use of (PCA-SIFT) that provide high quality match and for indexing the local descriptor locality sensitive hashing is used because large numbers of features are extracted. As in this, part based approach is used so this system is greatly defiant to scaling, cropping and other transform on which global descriptor can't be apply. Drawback of this is that system requires to query from hundred to thousand of feature at a time because it use the part base approach.

Wan-Lei Zhao et al [29] have proposed an approach for near duplicate keyframe identification with the use of matching, filtering and learning of the local interest point with the use of PCA-SIFT descriptor. In this LIP based feature are extracted and for the matching, new algorithm is proposed known as one-to-one symmetric matching (OOS) algorithm. LIP-IS index structure is used with OOS algorithm that has the ability to maximize the collision possibility when two LIPs are same. LIP-IS cannot effectively handle million of the keyframe pair in large video quantity.

There are various systems which used BOF but there are many problems with these systems. For local feature detection they have high time complexity, local descriptor has reduction in discriminability because of BOF quantization and they neglect geometric relationship amongst local feature obtained after BOF representation. In [30] H. Xie proposed a new framework to overcome the above discussed short comings by using Graphical Processing Unit (GPU). A fast local feature detector is designed known as coined Harris-Hessian which speeds up the detection of local feature. In this spatial information about every local feature is integrated so that discriminability is improved. For refining the search result, a new pair wise weak geometric consistency constraint (P-WGC) algorithm is proposed.

Zhaofeng Li and Xiaoyan Feng[31] have proposed approach that automatically detect the near duplicate image based on the visual word model. SIFT descriptor is used for representing the visual content of image which is very effective method for detecting the local feature of the images. Then SIFT feature of the image are clustered into a number of cluster with the use of k-mean algorithm. Visual word is considered as the centroid of every cluster and these centroids are used for generating the visual word vocabulary. LSH is used for mapping these high dimensional visual words into the low dimensional hash bucket space. Then visual features are converted to histogram. Near duplicate images are detected by computing the histogram distance.

In [32] J. Li et al presents the improved framework of BOW that detect the near duplicate images on web .For encoding the features , Locality-constrained Linear Coding (LLC) by spatial pyramid is established which is based upon the SIFT feature descriptor. To compare two histograms, weighted chi-square distance metric is

introduced with inverted indexing scheme. They built 6K dataset which have eight categories of objects that are applicable for image retrieval and classification also.

F. Nian et al [33] have proposed efficient and effective local based representation method that encodes the image as binary vector known as LBR. In this from the image, local region are extorted. The extracted region is then converted into simple and efficient feature describing its texture. Then, they calculate statistical histogram over all local features and this statistical histogram are encoded to binary vector as image representation. This method use global visual distribution and local region texture. Memory cost is reduced in real application and online computation is also done efficiently by this method.

Q. Zhu[34] have presented new contextual descriptor which measure the contextual similarity of the visual word that quickly eliminate the dissimilar visual word and decrease the number of candidate images. This descriptor determines relationship of spatial position and dominant orientation between referential visual words and their context. This contextual descriptor rigorously determines the spatial relation of context. This is strong for image editing operations like rotation, cropping and scaling. This descriptor is not robust for perspective transformation of the image. This approach is extremely valuable for near duplicate image searching in the large scale.

In [35] Li Liu et al proposed variable length signature for the near duplicate image matching. In this signature represents the image, where length of signature is fluctuating with respect to number of patches in image. Probabilistic center-symmetric local binary pattern descriptor is proposed to categorize the look of every image patch. Apart from this, spatial relationships between the patches are also captured. Earth mover's distance is computed for checking the similarity between the images. This approach is applied on two applications, first is near duplicate document image retrieval and near duplicate natural image exposure.

In Saehoon Kim et al [36] presented a solution for near duplicate image discovery if one billion images are there. This method is effortlessly implemented on the Map Reduce framework. In this paper they pioneer a seed growing step that is indented to effectively decrease the numbers of false positive in the cluster seed which have the time complexity of O(cNL), where N is the number of images ,L is the seed and c is

little sufficient for a data set of billion images. Here the main feature of seed growing phase is bottom k-min-hash that create dissimilar signature in a sketch for removing the candidate images which have only one common visual word in cluster seed. They show in evaluation that their method searches the cluster of near duplicate with high value of recall and precision.

Table 2.3 : Comparison of Near Duplicate Image Detection

Authors	**Year**	**Type of features extracted (Globel /Local)**	**Feature detector**	**Feature descriptor**	**Indexing/ Similarity distance**
R. Sukthankar et al. [28]	2004	Local features	Lowe's Difference of Gaussian (DoG) detector	PCA-SIFT	LSH
W. Zhao et al. [29]	2007	Local feature	Lowe's difference of Gaussian(DoG) detector	PCA-SIFT	LIP-IS
H. Xie et al[30]	2011	Local feature point	Coined Harris-Hessian (H-H)	SIFT	Inverted indexing,Pair wise geometric similarity
J. Li et al.[32]	2013	Local feature points	LLC	SIFT	Inverted indexing, χ^2 Distance metric
F. Nian et al[33]	2015	Local and global features	LBR	Features histogram	Compare histograms
J. Yao et al. [34]	2015	Local features	Selecting the referential local feature	SIFT descriptor , contextual descriptors	Maximal distance of contextual descriptor is used
L. Liu et al.[35]	2015	Patches are used		Probabilistic Center-symmetric Local Binary Pattern (PCSLBP)	Earth moving distance

J. Li et al[44]	2015	Globel and local		Color moment and hierarchical wavelet packet descriptor for global features and SIFT for local features	K-mean clustering

This chapter discussed and compared the feature detection techniques, feature detection techniques, nearest neighbour search algorithm and techniques of near duplicate images detection. Next chapter will discuss about the research gaps, problem statement, objectives and methodology.

CHAPTER 3 Problem Statement

This chapter includes the research gaps, problem statement of the thesis, objectives and what methodology is used to solve the problem.

3.1 Research Gaps

With the advancement of the Internet and Smartphone, it is easy to capture, edit store and then share the image data. This data may be photograph, painting, document and medical image which brought the challenge to store and retrieve these images. One of these problems is searching of near exact duplicate images from cloud image database using the query by the user. Currently, most of the available techniques are detecting near duplicate images. No technique for searching the near exact duplicate images in cloud database is currently available. The problem with searching and retrieving the image is to find the image features those are calculated from one image to another reliably. These features are unique for information it captured.

3.2 Problem Statement

A method of discovering the near exact duplicate image based on existing technologies is proposed. Their formulations as well as its relative efficiency are discussed. Binary Robust Invariant Scalable keypoint and Locality sensitive hashing are used for implementation of searching near exact duplicate images. This system is able to load the images, index the large numbers of the images features and then search the near exact duplicate images from cloud database related to the query image.

3.3 Objectives

The objectives of thesis are:

- Study, examine and explore the different technique for feature detection, feature description and similarity search techniques.
- Search the near exact duplicate images using Binary robust invariant scalable keypoint and locality sensitive hashing technique.
- Test and validate the proposed method using precision and recall.

3.4 Methodology

The following steps are taken to search the near exact duplicate images in cloud database:

- Study the basics of the feature detection, feature description and indexing techniques.
- Analyze the near duplicate images detection techniques implementation.
- Determine the features of the image for differentiation.
- Calculate the features of each image and build the binary descriptor of features.
- Index the images in the best way so that they are easy to access and fast matching can be performed.
- Search and retrieve the near exact duplicate images related to query image.

The image database is taken from Multimedia Information Retrieval Flickr (MIRFlickr) [45] for the purpose of testing the proposed near exact duplicate images detection system.

CHAPTER 4 Design Techniques

This chapter discusses how the problem stated in previous chapter can be solved. Here BRISK and LSH are combined to solve the stated problem. In this chapter BRISK and LSH are discussed in detail. Also, block diagram and the steps of proposed solution are discussed.

4.1 Binary Robust Invariant Scalable Keypoints (BRISK)

BRISK is a binary robust invariant and scalable keypoint detector and descriptor. Robust invariant and scalable means that an object can be identified even if size of object is increased or decreased and it is rotated about axis in image. Variances occur due to things exist in reality and capture from devices. For the image feature invariance is important property. It is a measurement of the similarity with features of two images that cannot be duplicated. So features describe the unique data point that any image contains [24].

There are three key stage of BRISK:

- Scale Space Keypoint Detection
- Descriptor composition,
- Keypoint matching.

4.1.1 Scale Space Keypoint Detection

The methodology for detection of region of interest in image is motivated from the work of Mair et al.[33] that focuses on the computation efficiency. The extension of their AGAST is used for the feature detection known as FAST. FAST has higher computational efficiency. BRISK is invariant to scale. So in BRISK maxima is counted not only in image plane, but also in scale space by using FAST.

Rather than discretizing the scale axis on the coarser interval, BRISK detector estimates true scale of every feature in continuous scale space.

In BRISK method, there are n octave c_i and n intra octave d_i for the pyramid layer in scale space where i=0, 1,2,3,4..... , n-1 and generally n=4. Half sampling of the original image is carried out gradually then octave is obtained. Every intra octave d_i is placed in-between layers c_{i+1} and c_i Figure 4.1:

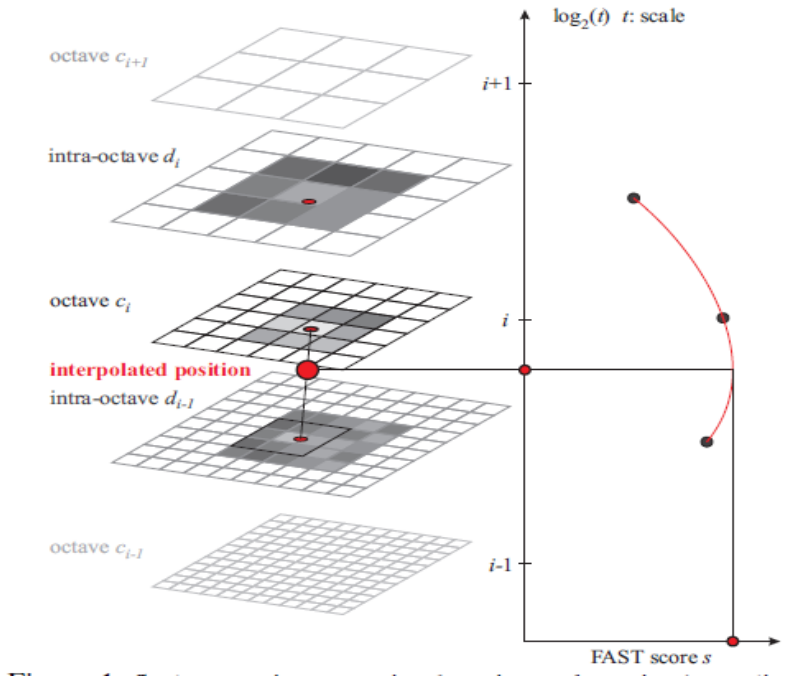

Figure 4.1: Scale space point detection [24]

Firstly downsampling is applied on the original image c_0 where factor for this is 1.5 and it gives the intra-octave d_0. The rest of intra octaves are obtained by applying the consecutive halfsampling. So if t represent scale then $t(c_i)=2^i$ and $t(d_i)=2^i *1.5$.

Usually 9-10 mask is used in BRISK that basically require at least 9 successive pixels in 16 pixel circle. It is either adequately darker or brighter than central pixel for the FAST condition to be fulfilled. Firstly, to recognize the potential region of interest, apply the FAST 9-10 detector on every intra octave and octave independently by using same threshold T. Then points which belong to these regions are deal with the non-maxima suppression in the scale-space. Initially, the points have to fulfil the maximum clause regarding its 8 neighbouring FAST score s in similar layer. Here s is the maximum threshold which considers image point as a corner. Secondly, score in layer below and above a need to be lower. The same sized square patches are checked: side length is 2 pixels in layer which is max value. At the boundaries of the patch some interpolations are applied because neighbour layers are represented by different discretization. Figure 4.1 shows the sampling and maxima search. There is a special case to detect the maxima across scale axis at octave c_0: apply FAST 5-8 mask on the c_0 for getting the FAST score of the virtual intra octave d_{-1} that is below c_0.

Short pair and long pair [37]:

For the sampling pattern, long pairs and short pairs are used. Short pairs are the pairs of sampling point which have distance less than the threshold δ_{max} and long pairs are the pairs of sampling point which have distance greater than δ_{min},where δ_{min}>δ_{max}.

For intensity comparisons short pairs are used and for determining the orientation long pairs are used. Short pairs build the BRISK descriptor.

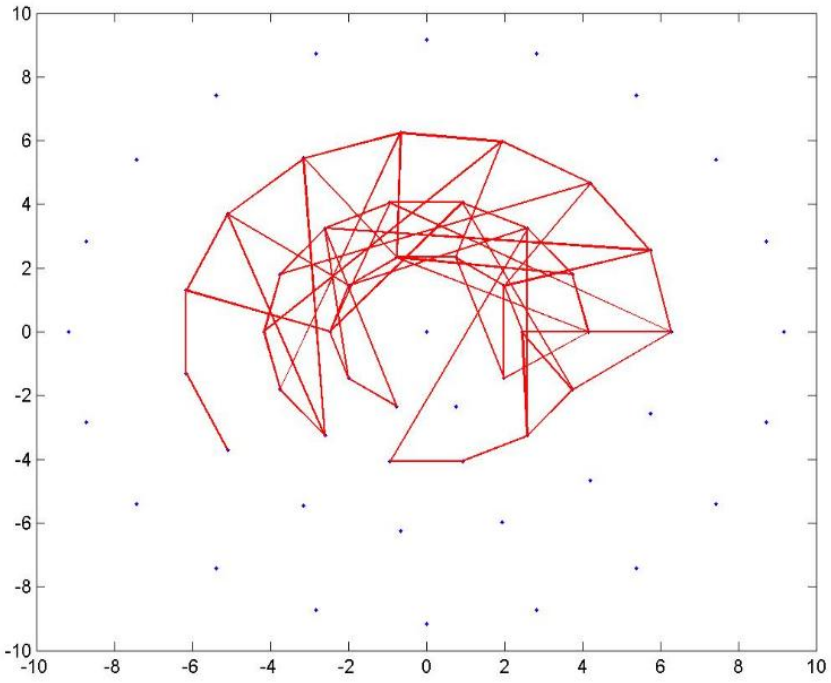

Figure 4.2: Short pairs, here one pair is denoted with the red line [37]

In BRISK Gaussian smoothing is applied with standard deviation σ_i .Standard deviation is proportional to the distance between points on relevant circle. This smoothing is applied to keep aliasing effect away during sampling intensity of image for a point p_i in pattern. Scaling and positioning of the pattern is done according to the particular keypoint k in image. Consider any sampling-point pair (p_i,p_j) from N*(N-1)/2. Smoothed intensity values are used to estimate the local gradient on these points. These are I (p_i,σ_i) and I(p_j,σ_j) respectively. Gradient (p_i, p_j) is given as,

$$g(p_i, p_j) = (p_j - p_i).\frac{I(p_j,\sigma_j)-I(p_i,\sigma_i)}{\|p_j-p_i\|^2} \quad (1)$$

Set $\mathcal{A}$ consist all pair of sampling-points:

$$\mathcal{A} = \{(p_i, p_j)\epsilon\mathbb{R}^2 \times \mathbb{R}^2 | i < N \wedge j < i \wedge i, j \in N\} \quad (2)$$

Let S is the set of short distance pair and $\mathcal{L}$ is set of long distance pair:

$$S = \{(p_i, p_j) \in \mathcal{A} | \|p_j - p_i\| < \delta_{max}\} \subseteq \mathcal{A} \quad (3)$$

$$\mathcal{L} = \{(p_i, p_j) \in \mathcal{A} | \|p_j - p_i\| < \delta_{min}\} \subseteq \mathcal{A} \quad (4)$$

Threshold distance are δ_{min}=13.67t and δ_{max}= 9.75t where t is scale. The characteristic pattern direction for keypoint k is:

$$g=\begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i,p_j)\in\mathcal{L}} g(p_i, p_j) \quad (5)$$

From above computation long distance pairs are utilized based upon the assumption. Local gradients extinguish. Each other so they are not compulsory in global gradient determination. It is confirmed through experiment that variations are done using δ_{min} which is a distance threshold.

4.1.2 Building the Descriptor

BRISK applies sampling pattern which is rotated by $\alpha =$ arctan2 (g_y, g_x) about keypoint k. This rotation is applied for the formation of scale and rotation normalized descriptor. Bit vector is constructed by performing the short distance intensity comparison on point pairs. Each bit b is corresponding to intensity:

$$b= \begin{cases} 1, & I(p_j^\alpha, \sigma_j) > I(p_i^\alpha, \sigma_i) \\ 0, & otherwise \end{cases}$$

$$\forall (p_i^\alpha, p_j^\alpha) \epsilon S \quad (6)$$

BRIEF and BRISK both build the descriptors based upon the intensity comparison but BRISK has some advantages over BRIEF. BRISK uses the deterministic sampling pattern that results in uniform sampling point density for a given radius about keypoint. BRISK perform sampling on point rather than pair wise comparison.

BRISK limits the complexity of look up value of intensity. In this comparison are limited spatially so that brightness variations are only locally consistent.

4.1.3 Descriptor Matching

Matching of BRISK Descriptor is done by hamming distance. Hamming distance is calculated by counting the number of the bits different in two descriptors.

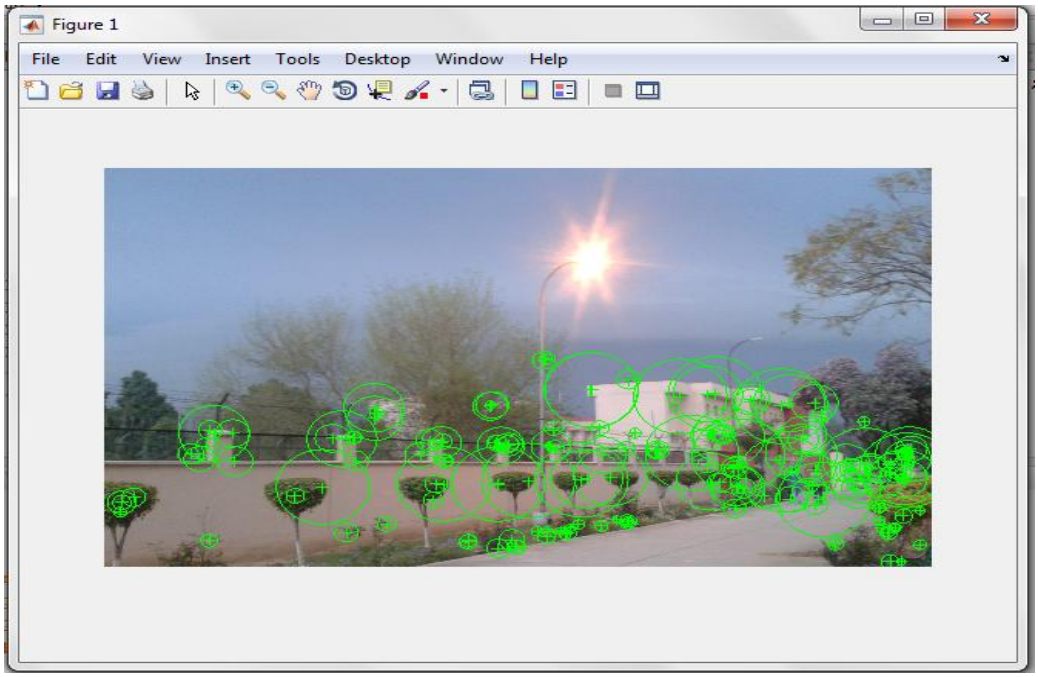

Figure 4.3: BRISK points in image

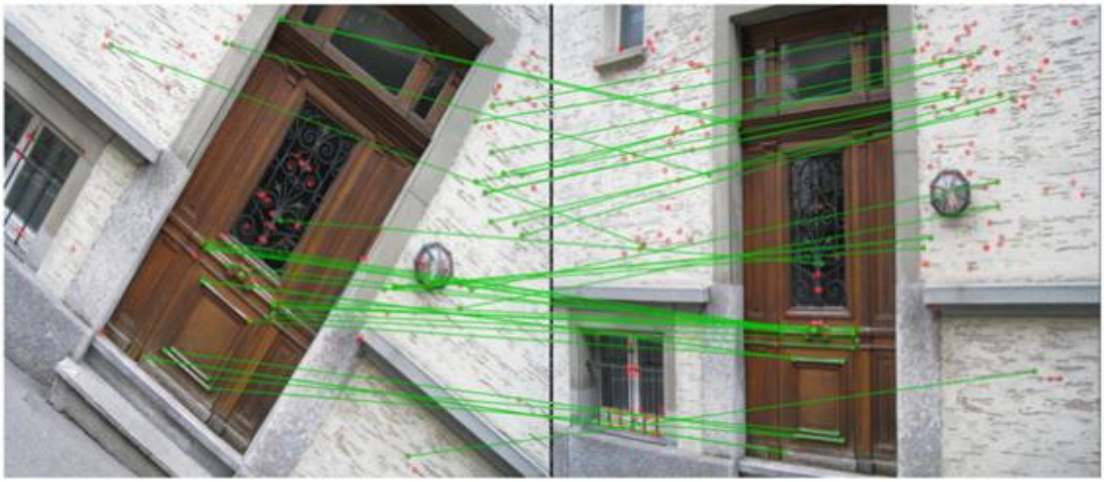

Figure 4.4: Matching of the BRISK point where one is original image and second is rotated version of original image [24]

4.2 Locality Sensitive Hashing (LSH)

Locality sensitive hashing was introduced by Piotr Indyky et al. [38] in 1998. This is used for finding the solution of approximate nearest neighbour search. Locality hashing sensitive is a technique that is used for identifying the similar item in given set. In the naive approach there is a comparison of all pairs of items within a set. In LSH, items are hashed into the buckets in such a way that similar items are placed into the same bucket. So the number of comparisons for finding similar item will be reduced because only the items present in one bucket are compared. Locality sensitive hashing is mostly used when a large amount of data is needed to be compared.

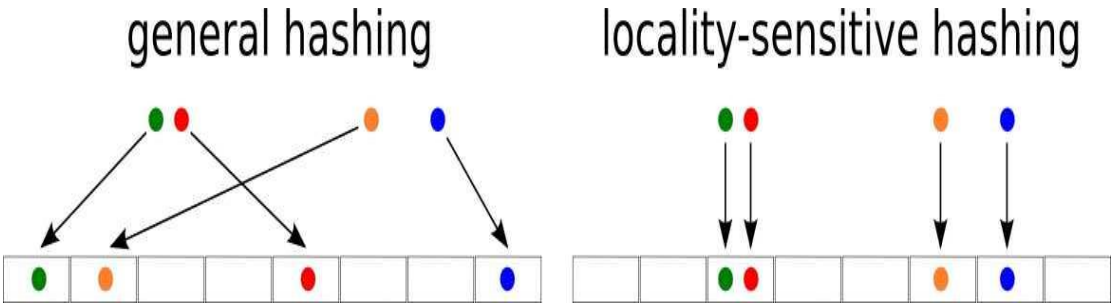

Figure 4.5:General hashing versus locality sensitive hashing [39]

If there is large amount of data or data has high dimensions then feature extraction techniques are used to reduce the dimensions. The major application of the locality sensitive hashing is to provide a technique which performs effective approximate nearest neighbour search via probabilistic dimension reduction of the high dimensional data. This type of the dimensional reduction is performed through some feature extraction by hashing. There are various schemes of hashing are used. These schemes depend on the types of data. Locality sensitive hashing is used in many fields like data mining, computer vision, chemical similarity and plagiarism detection, pattern recognition.

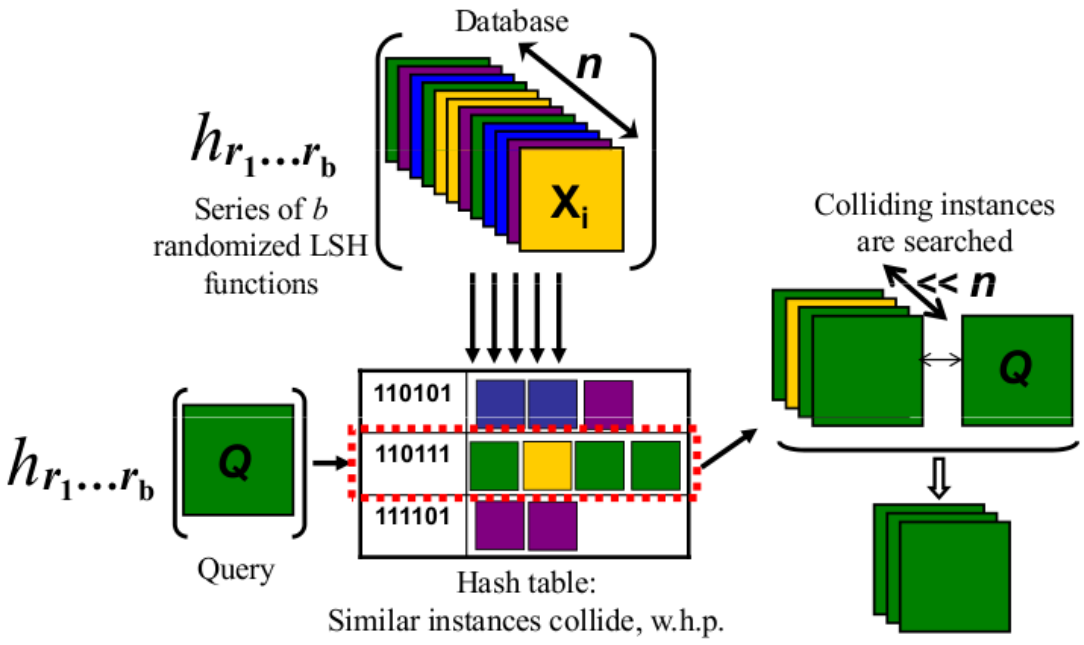

Figure 4.6: How LSH gives result based on query [40]

In [38]1999 Piotr Indyky et al. presented the paper on the similarity search using hashing in the high dimensional data. The basic thought of the LSH is to project the high dimensional data into the low dimensional data.

The idea behind the LSH is that if there are two points which are close together then these points are remain close together after some projection operations. It effectively works for the high dimensional data. LSH reduces the dimensionality of the data which has high dimension.LSH hashes the input data due to which the similar data maps to similar bucket with the high probability. To determine the approximate or near exact neighbour, hashing used the query points and then retrieves the elements which are stored in the buckets containing that points.

Suppose there is a family of hash function $\mathcal{H}$ mapping $\Re^d$ to universe$\mho$. $\mathcal{H}$ is called (r, cr,P_1, P_2) sensitive, if for p,q∈ $\Re^d$

$$\text{If } \|p-q\| \leq r \text{ then } P_{\mathcal{H}}[h(p)-h(q)] \geq P_1,$$

$$\text{If } \|p-q\| \leq cr \text{ then } P_{\mathcal{H}}[h(q)-h(r)] \leq P_2$$

For the usefulness of the LSH family, $P_1 > P_2$.

Consider there is a database which is a set of vectors a1, a2, a3… then there is a given query q and there is need to find out the similar items in the database. In LSH, data is projected into low dimensional binary space or there is a bit vector known as hash key to which data points are mapped. If projection is done appropriately then approximate

near neighbor can be find out in sub-linear time. These hash keys are generated by applying the b binary hash functions h1, h2…h_b to the every database objects. For the validity every hash function h has to satisfy LSH property:

$$\Pr[h(a_i) = h(a_j)] = sim(a_i, a_j),$$

$Sim(a_i, a_j) \in [0,1]$ is similarity function defined on the set of objects. The objects of high similarity are hashed into same buckets in hash table with high probability. The main task in LSH indexing is to generate a set of hash functions which satisfy the above mentioned criteria. The simplest method to generate the LSH family is the bit sampling. This works for hamming distance for d-dimensional binary vectors ie. $\{0,1\}^d$.

$$\mathcal{F} = \{h: \{0,1\}^D \rightarrow \{0,1\} | h(a) = a_i, i = 1 \dots \dots D\}$$

Here a_i is i^{th} coordinate of a. 'h' is a random function which is selected from $\mathcal{F}$ and it selects a random bit from input vector.

- **Hamming distance in LSH**

Hamming distance between two vectors in a given space of vectors is defined as the number of different components in which they are different. It is already known that it is a distance measurement. Hamming distance is never negative and if hamming distance is zero then vectors are equal. It is not dependent on the sequence of the vector. Hamming distance is used only when vectors are Boolean i.e. they contain 1's and 0's only.

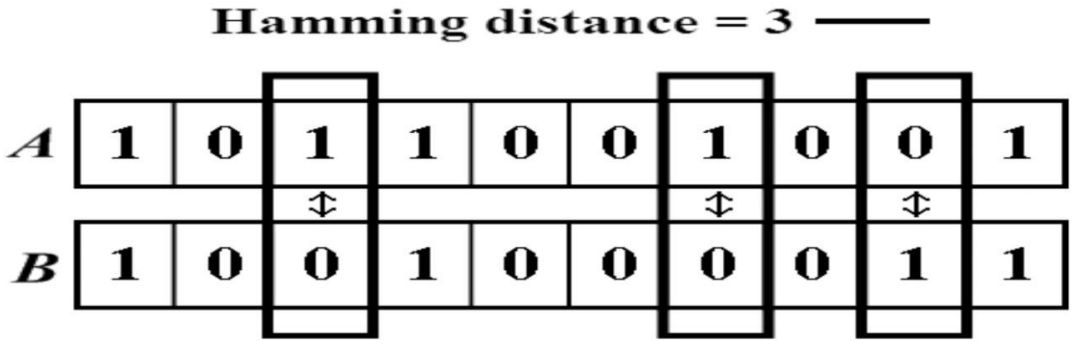

Figure 4.7: Example of hamming distance

Consider there is a space of d-dimensional vectors and hamming distance is denoted by h(x, y) for vectors x and y. Then choose any position of vectors let it be i^{th} position then a function $f_i(x)$ can be define for the i^{th} bit of vector x. So $f_i(x) = f_i(y)$ is possible if and only if at the i^{th} position vector y and x agree. The probability of $f_i(x) = f_i(y)$ for

randomly selected i will be exactly 1-h(x, y)/d. So it is the fraction of the position on which vector x, y agree.

4.3 Block Diagram of Proposed Solution

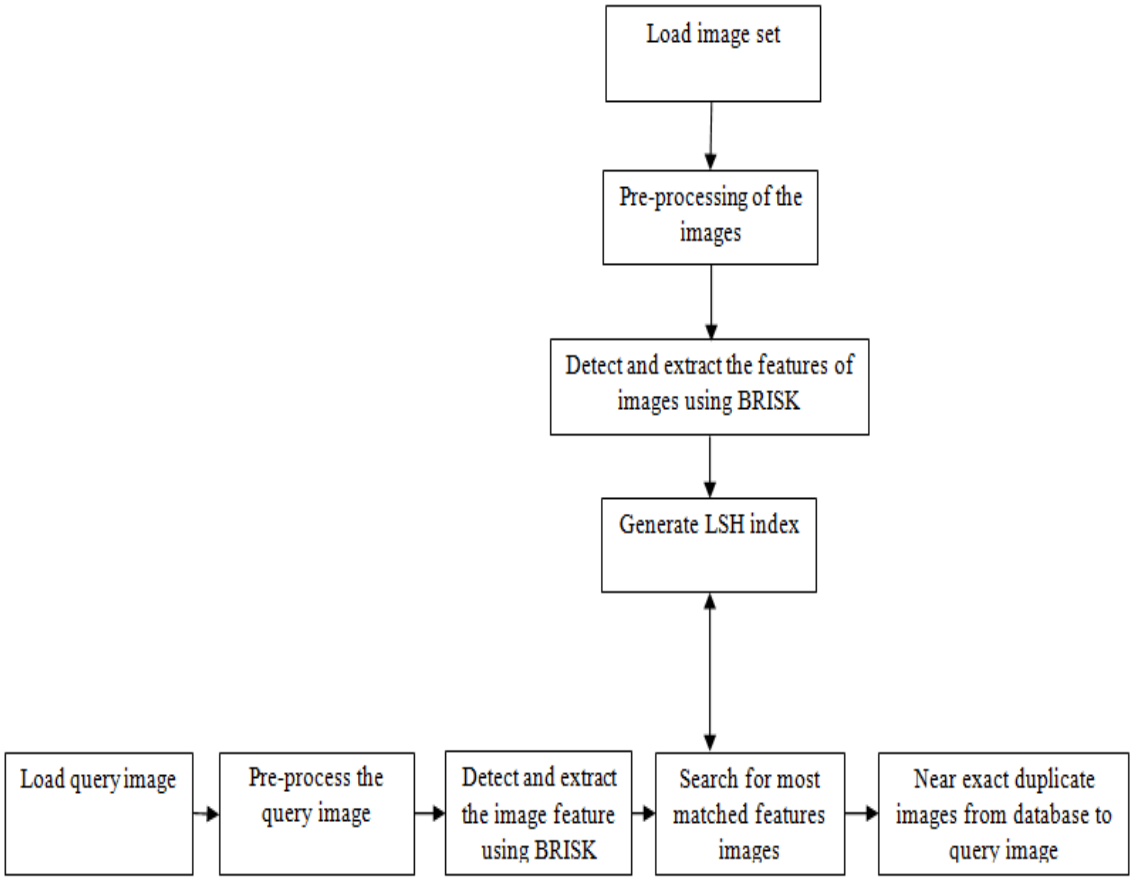

Figure 4.8: Block Diagram of Proposed Solution

4.4 Steps of Proposed Solution

- Index constructing

In this there is three data structure file table, keypoint table and hash table. File table stores the name of file, Keypoint table stores the keypoint of the images and hash table point to the keypoints of images. For the keypoint BRISK is used as feature detector and descriptor. *l* hash table are built by locality sensitive hashing algorithm, each hash table has its individual hash function. Hash table are concatenated and these are stored on disk sequentially. Hash table has fixed size, so there is need to determine

the number of keypoints before creating the hash table. Every hash table has B bucket, every bucket can store maximum *m* keypoints with the use ofα, $B= n/ (\alpha\ m)$ bucket for storing n keypoints. Steps for this are given below:

i. For each image in dataset find the keypoint and build the descriptor using BRISK
ii. Construct and store the file table
iii. Construct and store the keypoint table
iv. For each of *l* hash table: for each keypoint hash the keypoint and store id in table then store the hash table.

- Query Phase

After creating the index, user query an image for finding the near-exact duplicate images related to queried image. First find the keypoint and build descriptor. Then, calculate bucket id of every keypoint by applying hash function. Check all the keypoints in a bucket and verify if hash values match or not. Then read keypoint data from keypoint table to create list of near exact matches for query keypoint. Steps are given below:

i. Find the keypoint for query image using BRISK and a descriptor is build
ii. Calculate the *l* hash for descriptor.
iii. Sort the hashes by using bucket id and scan hash table.
iv. Return the keypoint id and scan the keypoint table and retrieve the near exact duplicate images.

This chapter included a detailed view of BRISK and LSH. The block diagram of proposed solution and steps of solution to the stated problem discussed in the previous chapter are also discussed. In next chapter implementation tool, implementation of proposed solution will be discussed. Next chapter will discussed the results by taking various query images and find the near exact duplicate images related to query image.

CHAPTER 5 Implementation and results

This chapter discusses the tool used for implementation of the searching the near exact duplicate image in the cloud database, implementation of proposed solution and query results. Performance evaluation of system is discussed by using precision recall graph.

5.1 Implementation tool: MATLAB 2015a

MATLAB stands for matrix laboratory. It was developed by the LINPACK and EISPACK projects .MATLAB is a high performance language which is used for the technical computing. MATLAB combines the programming, visualization, computation in a very simple and easy to use environment. This provides a way for the solution of problem in a well known mathematical format. In this array is the basic data element here dimensioning is not required. This is usually used for the

i. Algorithm development
ii. Math and computation
iii. Modelling, simulation, and prototyping
iv. Scientific and engineering graphics
v. Data analysis, exploration, and visualization
vi. Application development, including Graphical User Interface building.

This is used to solve lots of technical computing problems which are mainly related to vector and matrix formulation.

MATLAB has evolved in many phases according to the user needs. There are various sectors in which MATLAB is used such as in higher productivity research, analysis and development in the industries as well as in institute. MATLAB contains different application specific solutions that are known as toolboxes. Areas for which MATLAB contain toolboxes are fuzzy logic, simulation, neural network and image processing etc.

In the proposed solution for detecting and extracting the BRISK feature Computer Vision Toolbox in MATLAB2015a is used. Computer Vision Toolbox discussed below:

- **Computer Vision Toolbox**

In MATLAB, computer vision system toolbox provides the functions, algorithm and apps to design and simulate the video processing system and computer vision. In computer vision system toolbox anyone can perform object detection and track object; detection, extraction and matching of the features; calculate the motion; and process the video. Camera calibration, stereo vision and reconstruct 3-D are supported by system toolbox for 3-D computer vision. For the machine learning frameworks training can be done for detecting the object, recognition of object, and image retrieval system. Computer vision toolbox supports the C-code generation and fixed point arithmetic for quick prototyping and designing of embedded system.

Here are some functions that are used in implementation of proposed solution:

i. **rgb2gray () -** It converts RGB image to gray scale.

   Syntax:

   I = rgb2gray (RGB)

   This converts the truecolor RGB image to gray scale image I. Here function rgb2gray converts the image in grayscale via remove saturation and hue information but it preserves the luminance.

ii. detectBRISKFeature()- This function detect the BRISK feature then return the BRISKpoints object.

   Syntax:

   points = detectBRISKFeatures(I),

   points = detectBRISKFeatures(I,Name,Value)

   here I is the 2-D grayscale image I.

iii. extractFeature() – This function extract the interest point descriptor.

   Syntax:

   [features,validPoints] = extractFeatures(I,points)

   [features,validPoints] = extractFeatures(I,points,Name,Value)

5.2 Implementation of Proposed Solution

Proposed solution for searching of near exact duplicate images in cloud database is implemented on MATLAB 2015a. Its basic functionalities are shown by snapshots below. Figure 5.1 shows the GUI of the system.

Figure 5.1: Home page

- **Indexing phase**

In the first phase of implementation, create indexing for all the images. Figure 5.2 shows that after click on the load image database button one dialogue box open through which the database of images is selected.

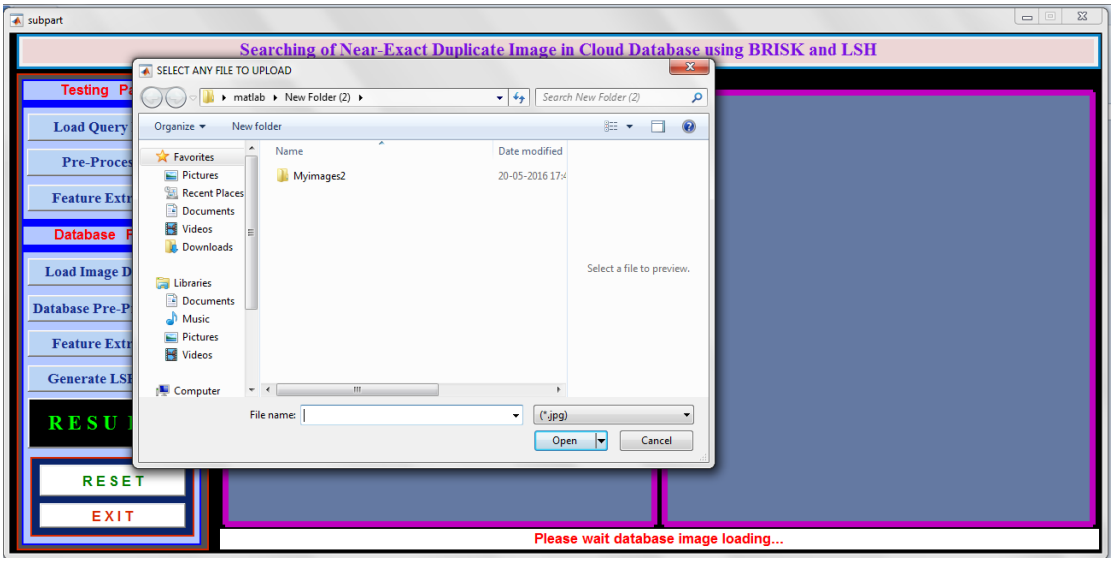

Figure 5.2: Browsing image database

After selecting the image database all images are loaded, Figure 5.3 shows that image database is successfully loaded.

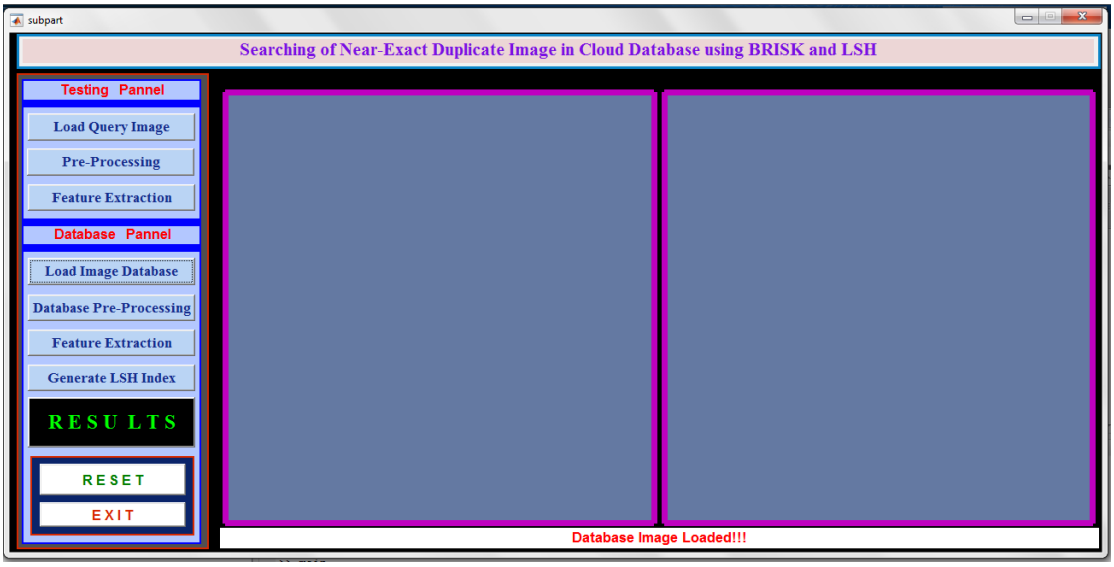

Figure 5.3: Image database successfully loaded

For creating the BRISK feature descriptor there is need to convert the truecolor images into the gray scale because BRISK doesn't works on the color images. So preprocessing of the images is performed where colored images are converted into the gray scale image, figure 5.4 show that preprocessing of the images has done.

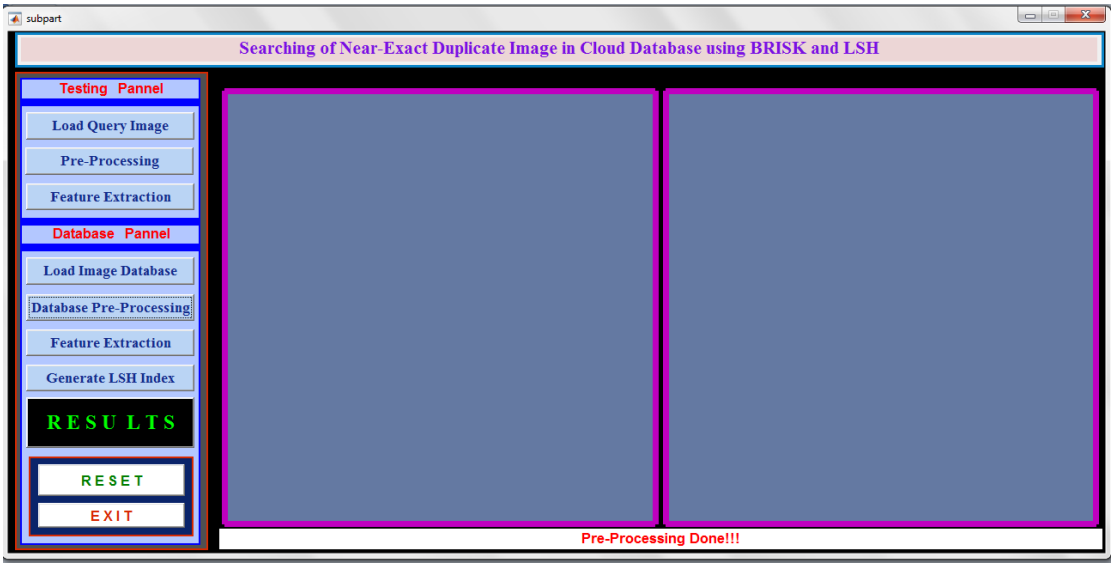

Figure 5.4: Preprocessing of image database done

Feature of the images are extracted which is shown in Figure 5.5. Here the BRISK descriptor is build for each image.

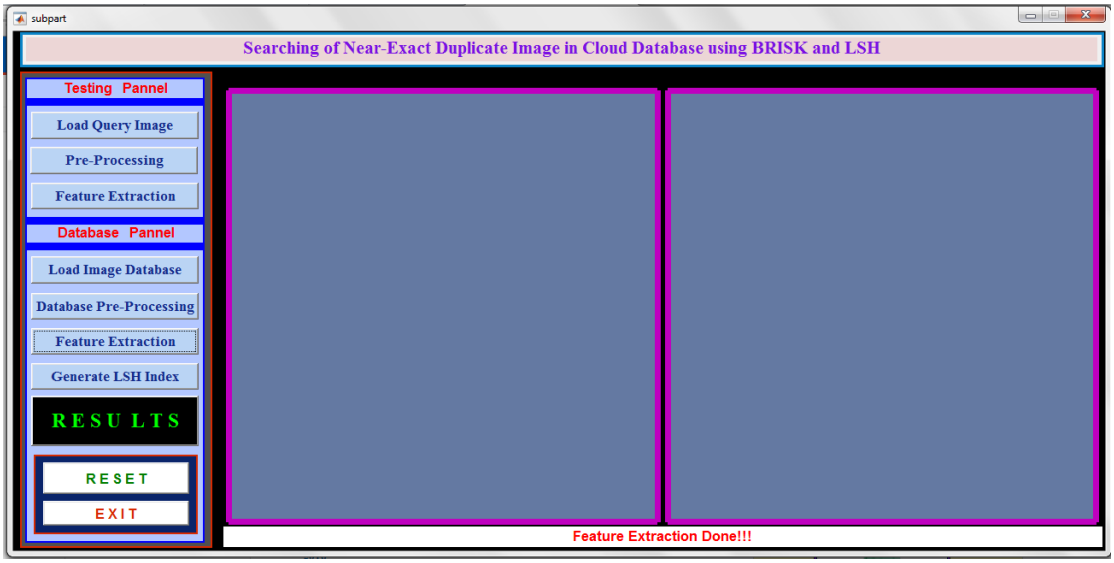

Figure 5.5: Feature extraction of image database

After getting the feature descriptor of all images then apply the LSH for creating the indexing which create hash table and feature table according to the feature descriptor, figure 5.6 shows the indexing is done.

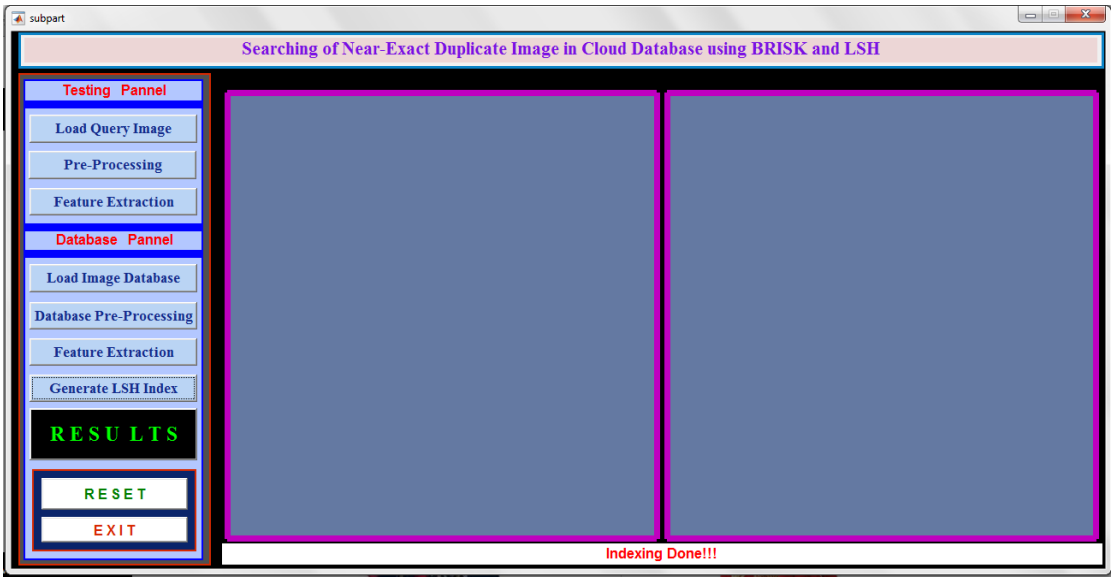

Figure 5.6: LSH indexing of image database

- **Query phase**

In this phase user issues the query for searching the near exact duplicate images in cloud database.

Firstly, load one query image for which user wants to search the near exact duplicate image. Figure 5.7 shows the query image loaded successfully. After loading the image, pre-processed the image where colored image is converted to the grayscale

image which is shown in Figure 5.8.

Figure 5.7: Query image is loaded

Figure 5.8: Pre-processing of query image done

When an image is converted to the gray scale it is ready for feature extraction, after click on the feature extraction button in query panel, the feature descriptor of the query image is generated Figure 5.9.

Figure 5.9: Feature extraction of query image

For searching the near exact duplicate image of query image click on Results button. Figure 5.10 shows the retrieved near exact duplicate image related to query image.

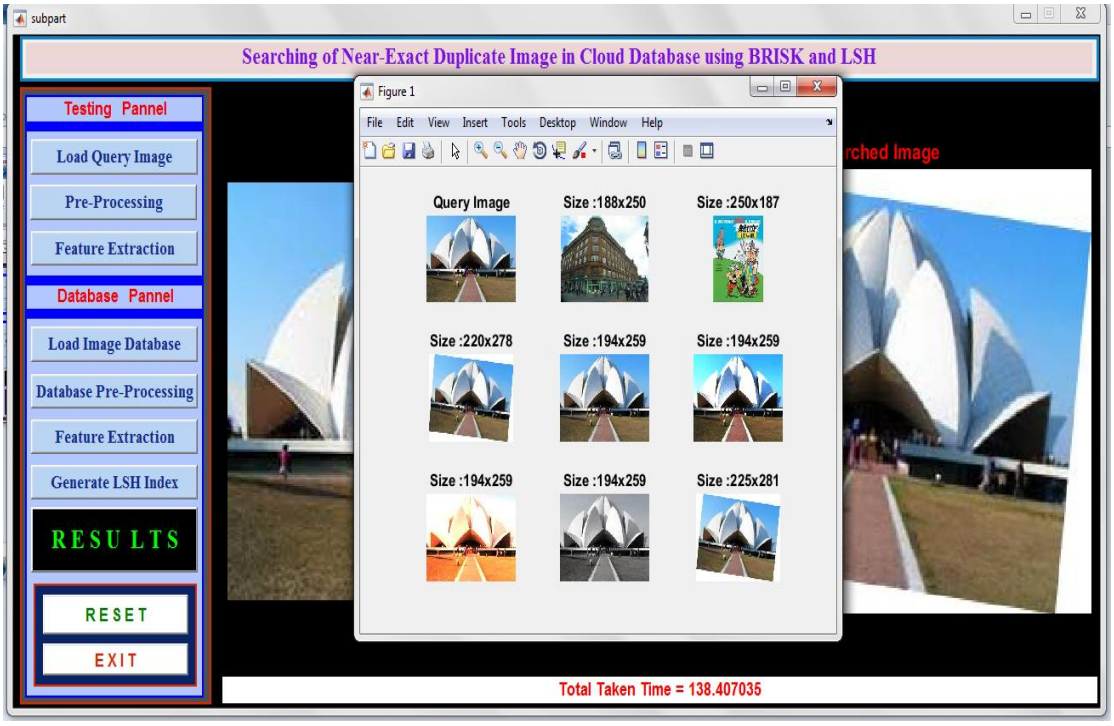

Figure 5.10: Retrieve the near exact duplicate image of query image

5.3 Image Query and Results

Here some query images are given to the system and search the near exact duplicate images related to those query images.

Query 1

Figure 5.11: Query image 1

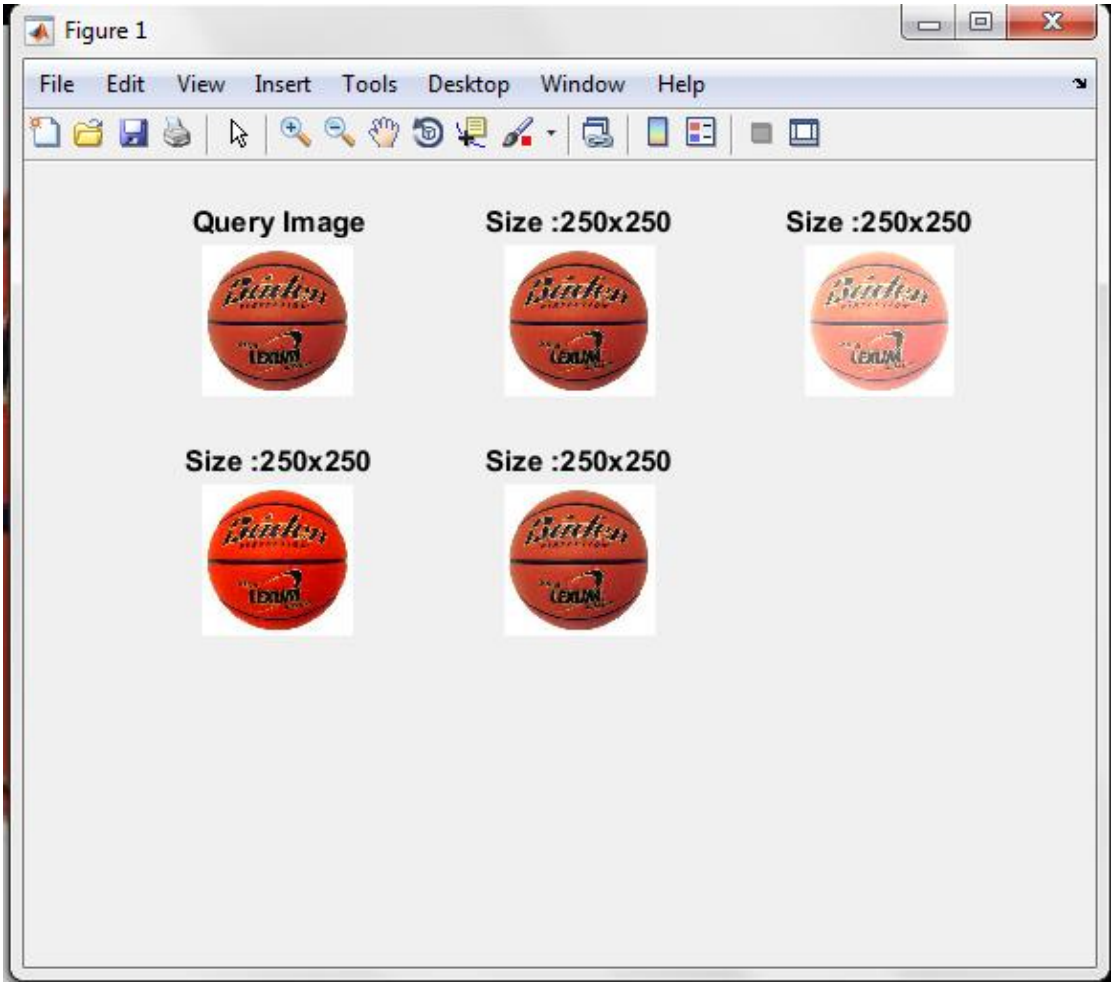

Figure 5.12: Near exact duplicate of query image 1

Query Image 2

Figure 5.13: Query image 2

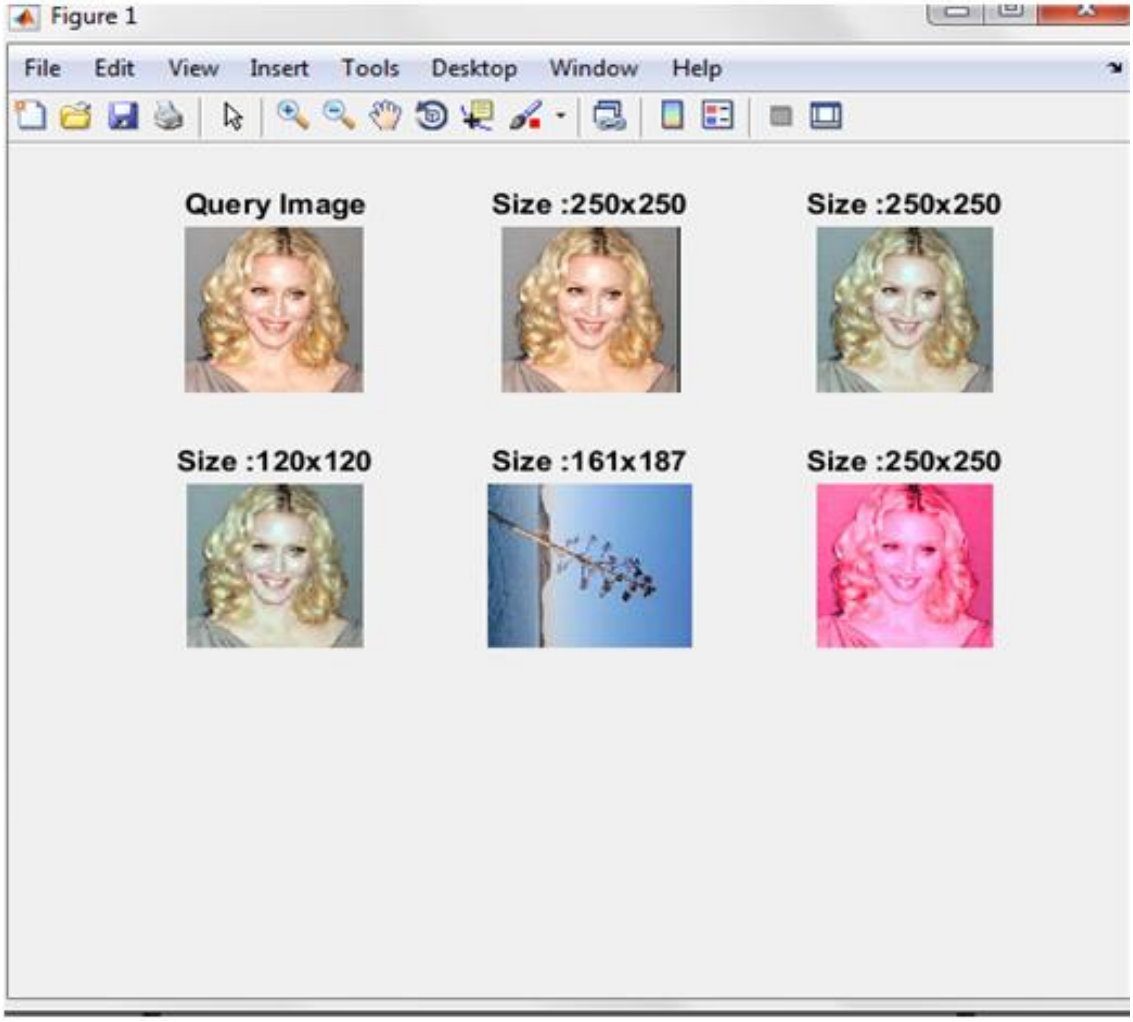

Figure 5.14: Near exact duplicate images of query 2

Query image 3

Figure 5.15: Query Image 3

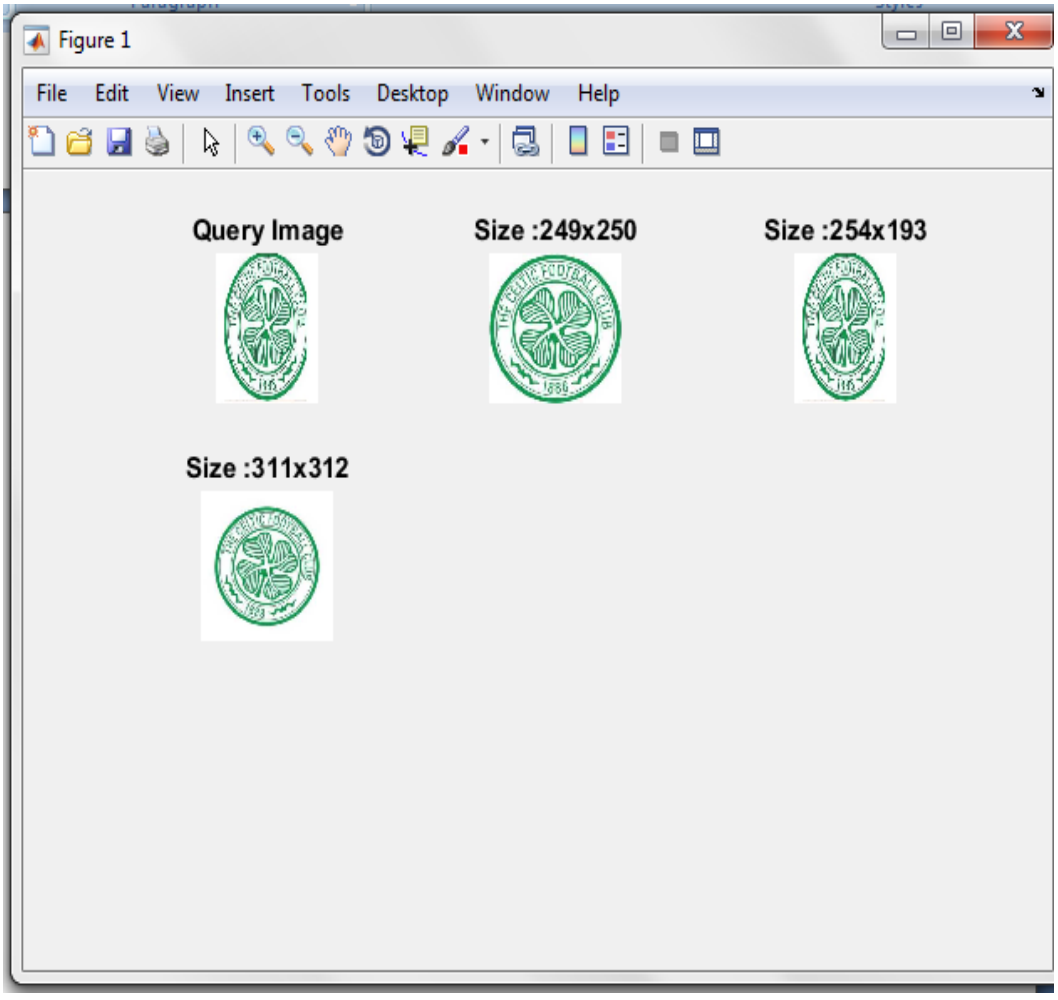

Figure 5.16: Near exact dupliccate images related to query image 3

Query Image 4

Figure 5.17: Query Image 4

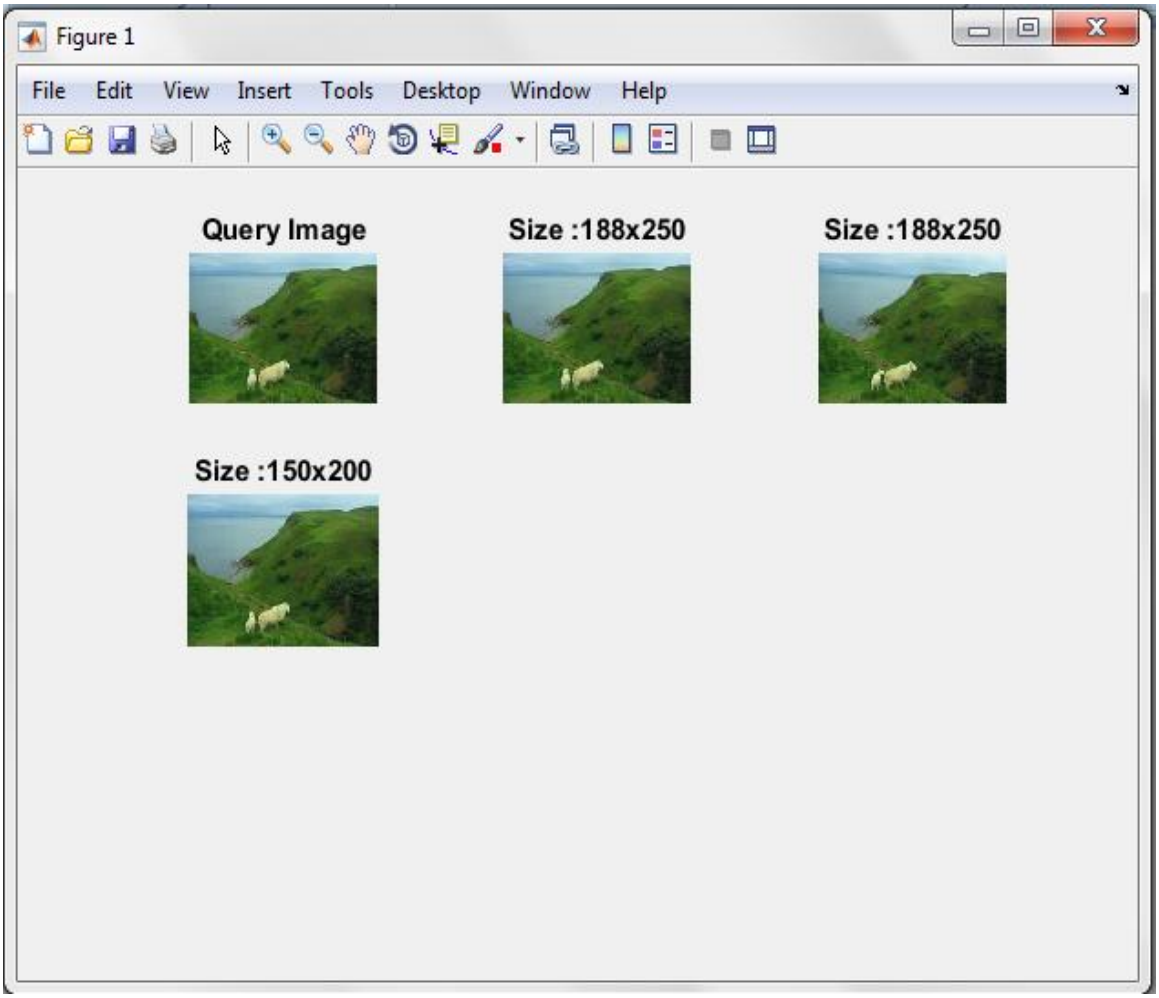

Figure 5.18: Near exact duplicate images related to query image 4

Query Image 5

Figure 5.19: Query image 5

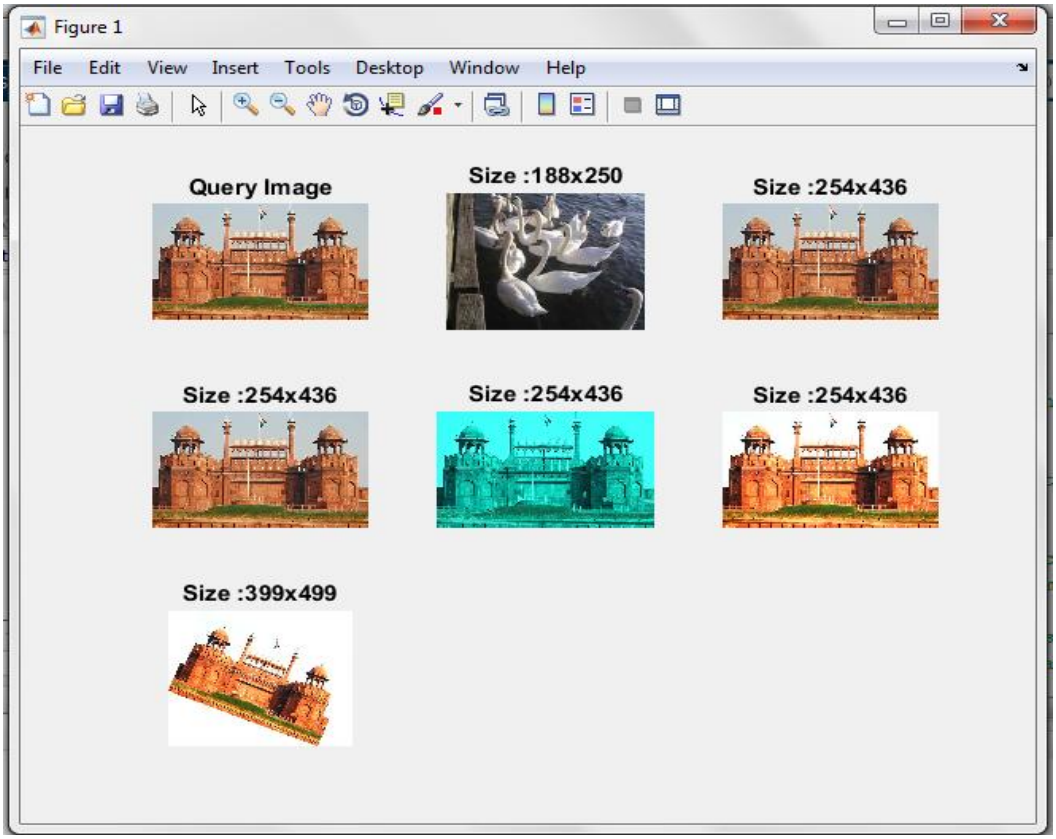

Figure 5.20: Near exact dupliccate images related to query image 5

5.4 Performance evaluation by using Metrics

Precision and recall are defined by Perry, Berry and Kent in 1955. In the pattern recognition and information retrieval, precision and recall are the metrics used to calculate the quality of output of classifier.

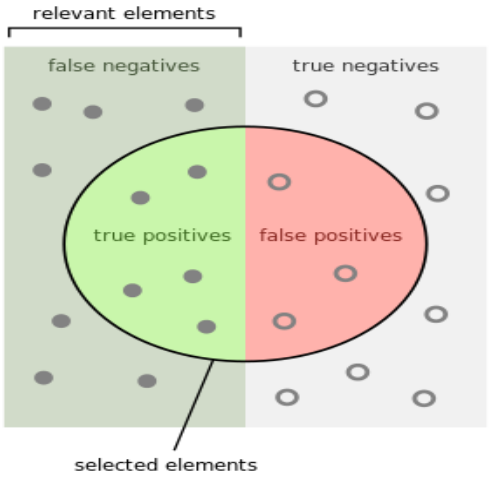

Figure 5.21: True positives and false positives

During the searching of the near exact duplicate images some false positive are also there. For the evaluation of the performance of system measure the precision and recall of proposed solution. Example of the false positive retrieved in the searching of near exact duplicate images is shown in Figure5.22. Here main aim is to minimize numbers of false positives and maximize correct positives. Correct positive is the match between the query image and its transformed form in database.

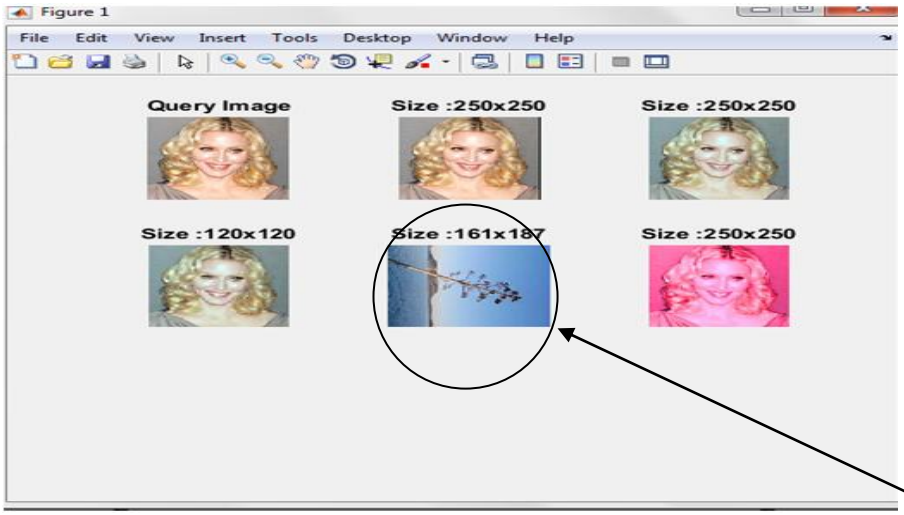

Figure 5.22: Example of false positive in retrieved near exact duplicate images

- **Precision**
  Precision estimates the result relevancy. The formula to calculate the precision value is give below:

$$precision = \frac{True\ positive}{True\ positives + False\ Positives}$$

- **Recall**
  Recall calculates the number of true relevant results returned. The formula to calculate the recall value is given below:

$$Recall = \frac{True\ positives}{True\ positives + False\ negatives}$$

If system has low precision but high recall then system returns numerous results but many of them are incorrect. If system has low recall but high precision then system returns few results but mostly results are correct. The best system with high recall and high precision will return large number of results and all results are correct.

In the implementation of the proposed solution some false positive are also retrieved in near exact duplicate images related to the query image. Accuracy of the system is affected by these false positives. Calculate the values of precision and recall for the retrieved near exact duplicate images and then plot the Precision-Recall graph which is shown in Figure5.23.

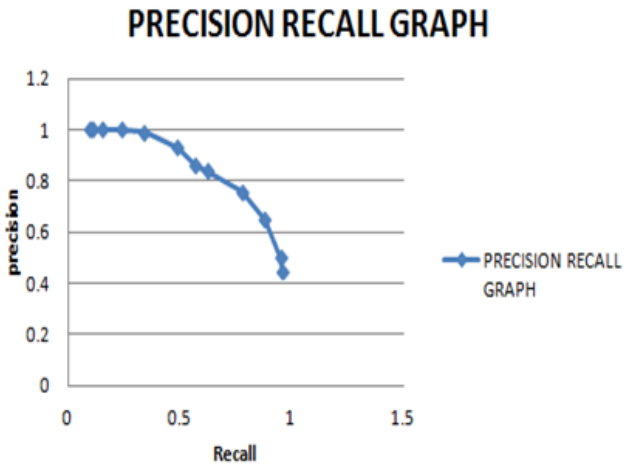

Figure 5.23: Precision-Recall graph

This chapter discussed about the implementation tool. Implementation of the proposed solution discussed with the help of snapshots. Next chapter will discuss about the conclusion of thesis work, thesis contribution and future work.

CHAPTER 6 Conclusions and Future Work

This chapter discusses the conclusions of the work presented in this thesis. This chapter ends with the discussion of the thesis contribution and the future work.

6.1 Conclusions

This thesis gives the overview of cloud computing and background of various feature detectors, feature descriptor and indexing techniques in digital image processing. In this thesis existing near duplicate images detection techniques are analyzed and compared. In this work a technique for searching the near exact duplicate images in cloud database has been proposed. A GUI in MATLAB 2015a is designed to query images and search the near exact duplicate images related to the query image. The technique is proposed by combining the Binary Robust Invariant Scalable Keypoint (BRISK) and Locality Sensitive Hashing (LSH). Precision and recall values are calculated for images retrieved and then precision recall has been graph. It takes less time to retrieve the near exact duplicate image for queried image requested by user.

6.2 Thesis Contribution

i. In this thesis, feature detector and feature descriptor techniques, nearest neighbor search algorithm have been discussed and analyzed.
ii. A technique has been proposed by combining BRISK and LSH for searching the near exact duplicate image in cloud database.
iii. Technique has been implemented in MATLAB 2015a.
iv. Experimental results are shown by taking some query images and duplicate results have been reported.

6.3 Future Work

The future work of the thesis can be:

i. Use other binary descriptor and other indexing like coherency sensitive hashing and matching scheme for enhancing the performance and reducing the time for finding the near exact duplicate image. Another distance measure may give improved discrimination between the distances of image features of same category and those from another category.
ii. This technique can be extended to data duplication technique for deleting the

duplicate images and which in consequence frees the storage space in cloud.

References

[1] P. Mell, T. Grance, “The NIST Definition of Cloud Computing,” National Institute of Standards and Technology, 2009.

[2] T. Harris, “Cloud Computing-An Overview”, Whitepaper, Torry Harris Business Solutions, 2010.

[3] David C. Wyld, “Moving to the Cloud: An Introduction to Cloud Computing in Government”, E-Government Series, 2009.

[4] Sajee Mathew, “Overview of Amazon Web Services”, 2014.

[5] Gartner AADI Summit. (2009). Cloud Computing as GartnerSees it. Gartner's Application Architecture, Development &Integration Summit.

[6] "IT Channel Glossary", CompuBase. March 2013.

[7] Online available: http://blog.econocom.com/en/blog/why-companies-are-adopting-hybrid-cloud,2015.

[8] J. Yang and Z. Chen, "Cloud computing research and security issues", *International conference on Computational Intelligence and Software Engineering (CiSE)*, pp. 1-3,

[9] M. Armbrust , A. Fox , R. Griffith , A. Joseph , R. Katz , A. Konwinski , G. Lee , David Patterson , A. Rabkin , I, Stoica , M. Zaharia, “A view of cloud computing” *Communications of the ACM*, vol.53 no.-4, April 2010 .

[10] M. Armbrust et al., “Above the Clouds: A Berkeley View of Cloud Computing”, Technical Report No. UCB/EECS-2009-28, February 2009.

[11] C. Clark, K. Fraser,S. Hand, J. Hansen, E. Jul, C. Limpach and A. Warfield, “Live Migration of Virtual Machines", *Proc. 2nd Usenix Symp. Networked Systems Design and Implementation*, pp. 273-286, 2005

[12] A. Beloglazov, R. Buvya, “Energy Efficient Allocation of Virtual Machines in Cloud Data Centers” in 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 577, 2010.

[13] P. Thibodeau, August 26, 2014, “Data centers are the new polluters” Computerworld [Online] Available: http://www.computerworld.com/article/2598562/data-center/datacenters-are-the-new-polluters.html [Accessed: March 1, 2015].

[14] Information Age, August 1, 2011 “Data centre energy usage slowing down –

report” www.information-age.com [Online], Available: http://www.information-age.com/technology/datacentre-and-it-infrastructure/1644738/data-centre-energy-usageslowing-down---report [Accessed: March 1, 2015]

[15] Peters, Richard Alan, II, "Introduction and Overview", Lectures on ImageProcessing, Vanderbilt University, Nashville, TN, April 2008, Available:http://www.archive.org/details/Lectures_on_Image_Processing.

[16] J. Canny, "A computational approach to edge detection", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 679-698, 1986.

[17] H. P. Moravec, “Towards Automatic Visual Obstacle Avoidance”, *Proc. 5th International Joint Conference on Artificial Intelligence*, pp. 584, 1977.

[18] S. Obdrzalek and J. Matas, "Object recognition using local affine frames on distinguished regions",*Proc. BMVC.*, pp. 113-122,2002.

[19] D. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91-110, 2004.

[20] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features”, i*n Proceedings of the European Conference on Computer Vision (ECCV),* 2006.

[21] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "Brief: Binary robust independent elementary features", *Computer VisionECCV* , pp 778-792 ,2010.

[22] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, "ORB: an efficient alternative to SIFT or SURF", *Computer Vision (ICCV) IEEE International Conference on*, 2011.

[23] A. Alahi, R. Ortiz and P. Vandergheynst, "FREAK: FAST retina keypoint", *Computer Version and Pattern Recognition*, pp. 510-517, 2011.

[24] Leutenegger, S., Chli, M., and Siegwart, R.Y., "Brisk: Binary robust invariant scalable keypoints", *IEEE Int. Conf. on Computer Vision*, No, pp. 2548-2555, 2011.

[25] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman and A. Wu, “An optimal algorithm for approximate nearest neighbor searching”, *Journal of the ACM*, vol. 45, no.6, pp. 891-923, 1998.

[26] A. Rajaraman, J. Ullman, "Mining of Massive Datasets, Ch. 3.", 2010.

[27] A. W. Moore. “An introductory tutorial on kd-trees,” extract from Efficient Memory-based Learning for Robot Control, Technical Report No. 209. Computer Laboratory, University of Cambridge, 1991.

[28] Y. Ke, R. Sukthankar, L. Hustonet "Efficient Near-duplicate Detection and Sub-Image Retrieval", *Proc. ACM Int',l Conf. Multimedia,* pp. 869-876,2004.

[29] W. L. Zhao , C. W. Ngo , H. K. Tan and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning", *IEEE Trans. Multimedia*, vol. 9, pp. 1037-1048, 2007.

[30] H. Xie, K. Gao, Y. Zhang, "Efficient feature detection and effective post-verification for large scale near-duplicate image search", *IEEE Trans. Multimedia*, vol. 13, no. 6, pp. 1319-1332, 2011

[31] Li Z, Feng X, "Near duplicate image detecting algorithm based on bag of visual word model", *J Multimedia*, vol.8, no.5, pp.557–564, 2013.

[32] J. Li et al "An Efficient Approach to Web Near-Duplicate Image Detection", Second IAPR Asian Conference on Pattern Recognition, 2013.

[33] F. Nian, T. Li, XinyuWu, Q. Gao, F. Li, "Efficient near-duplicate image detection with a local-based binary representation", *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2435-2452, 2015.

[34] J. Yao, B. Yan,Q. Zhu., "Near-duplicate image retrieval based on contextual descriptor", *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1404-1408, 2015.

[35] Liu, L., Lu, Y., Suen, C.Y., "Variable-length signature for near-duplicate image matching", IEEE *Trans. Image Process.* Vol. 24, no.4, pp. 1282–1295, 2015.

[36] Saehoon Kim , Xin-Jing Wang, Lei Zhang, Seungjin Choi, "Near Duplicate Image Discovery on One Billion Images" , *IEEE Winter Conference on Applications of Computer Vision*, pp. 943-950, 2015.

[37] Gil's Computer vision blog, "A tutorial on binary descriptors – part 4 – The BRISK descriptor"[online], online available: https://gilscvblog.com/2013/11/08/a-tutorial-on-binary-descriptors-part-4-the-brisk-descriptor/.

[38] Indyk, Piotr.; Motwani, Rajeev, "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality." *Proceedings of 30th Symposium on Theory of Computing*, 1998.

[39] I. Bressler, M. Hausburg, R. Richter, "Nearest neighbor problem" Online information available: http://cybertron.cg.tu-berlin.de/pdci08/imageflight/nn_search.html

[40] B. Kulis, K. Grauman, "Kernelized Locality-Sensitive Hashing Page",Online

information available: http://people.bu.edu/bkulis/klsh/klsh.htm.

[41] E. Rosten and T. Drummond, “Machine learning for high speed corner detection,” *in 9th Euproean Conference on Computer Vision*, vol. 1, pp. 430–443, 2006.

[42] Irwin Sobel, 2014, History and Definition of the Sobel Operator.

[43] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol 1,pp. 886 - 893 2005.

[44] J. Li, X. Qian, Q. Li, Y. Zhao, L. Wang and Y. Y. Tang, "Mining near-duplicate image groups", *Multimedia Tools and Applications*, vol. 74, no. 2, pp. 655-669, 2015.

[45] M. J. Huiskes, M. S. Lew, “The MIR Flickr Retrieval Evaluation”, *ACM International Conference on Multimedia Information Retrieval*, 2008.

[46] FixYa Cloud Storage Report [Online Available] Online: http://blog.fixya.com/pr/nov2012/cloud-storage-report.html.

List of Publications and Video Link

Publications

- **Accepted**
  Maneesha, Inderveer Chana, "Searching of Near Exact Duplicate Images in Cloud Database", IEEE International Conference on Inventive Computation Technologies (ICICT 2016).

Video Link

https://www.youtube.com/channel/UC-F5xY8jdtyV0QukpVXGjVA