STATUS OF THESIS

Title of thesis	Content-Based Image Retrieval Using Enhanced Hybrid Methods with Color and Texture Features

I, FAZAL-E-MALIK

hereby allow my thesis to be placed at the Information Resource Center (IRC) of Universiti Teknologi PETRONAS (UTP) with the following conditions:

1. The thesis becomes the property of UTP
2. The IRC of UTP may make copies of the thesis for academic purposes only.
3. This thesis is classified as

☐ Confidential

☑ Non-confidential

If this thesis is confidential, please state the reason:

____

____

The contents of the thesis will remain confidential for __________ years.

Remarks on disclosure:

____

____

	Endorsed by
_________________________	_________________________
Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor
Village & P/O Surizai Bala, Tehsil &	Assoc. Prof. Dr. Baharum Baharudin
District Peshawar, KPK, Pakistan.	
Date : ____________________	Date : ____________________

UNIVERSITI TEKNOLOGI PETRONAS

CONTENT-BASED IMAGE RETRIEVAL USING ENHANCED HYBRID METHODS WITH COLOR AND TEXTURE FEATURES

By

FAZAL-E-MALIK

The undersigned certify that they have read, and recommend to the Postgraduate Studies Programme for acceptance this thesis for the fulfillment of the requirements for the degree stated.

Signature: ______________________________

Main Supervisor: Assoc. Prof. Dr. Baharum Baharudin

Signature: ______________________________

Head of Department: Dr. Jafreezal Bin Jaafar

Date: ______________________________

CONTENT-BASED IMAGE RETRIEVAL USING ENHANCED HYBRID METHODS WITH COLOR AND TEXTURE FEATURES

By

FAZAL-E-MALIK

A Thesis

Submitted to the Postgraduate Studies Programme

as a Requirement for the Degree of

DOCTOR OF PHILOSOPHY

INFORMATION TECHNOLOGY

UNIVERSITI TEKNOLOGI PETRONAS

BANDAR SERI ISKANDAR

PERAK, MALAYSIA

NOVEMBER 2013

DECLARATION OF THESIS

Title of thesis	Content-Based Image Retrieval Using Enhanced Hybrid Methods with Color and Texture Features

I, FAZAL-E-MALIK

hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UTP or other institutions.

Witnessed by

Signature of Author	Signature of Supervisor
Permanent address:	Name of Supervisor
Village & P/O Surizai Bala, Tehsil & District Peshawar, KPK, Pakistan.	Assoc. Prof. Dr. Baharum Baharudin
Date:	Date:

DEDICATION

This thesis is dedicated to my beloved parents, wife and children.

ACKNOWLEDGEMENT

Alhamdulillah, first of all, I am deeply thankful to the Almighty Allah for the strengths, He has given me and His blessing in completing this thesis. He provided me this opportunity to accomplish my PhD research work.

I would like to bestow special appreciation to my supervisor, Dr. Baharum Bin Baharudin, for his valuable supervision, priceless motivation and constant support. His invaluable help of constructive comments and suggestions throughout the experimental and thesis work have contributed to the success of this research. Furthermore, I would like to express my appreciation to Dr. Aamir Saeed Malik for his technical excellence and tremendous grasp of research and experimental issues.

My acknowledgement also goes to all the teaching, technician and other staffs of my department for their unforgettable support during the course of my research study. Moreover, I would like to pay special credit to the Universiti Teknologi PETRONAS (UTP) Malaysia for awarding me fully funded financial support. I would not have been able to pursue and complete my research work, without that financial support.

Sincere thanks to my friends, Muzafar Khan and Muhammad Asif Khan for their kind support and valuable help during my research work. I pay special thanks to Shahid Nazir, Nazir Ahmad Khan, Khairullah Khan and Dr. Muhammad Azam Khan for the their support and encouragement at different stages. Finally, thanks to all other friends for their kindness and moral support during my study. Thanks for the friendship and memories.

Last but not least, my deepest gratitude goes to my beloved parents for their prayers. Special thanks to my wife, sister, brothers and parents-in-law for their endless love, prayers and encouragement. To those who indirectly contributed in this research, your kindness means a lot to me. Thank you very much.

Fazal-e-Malik, CISD, UTP

ABSTRACT

Content-based image retrieval (CBIR) automatically retrieves similar images to the query image by using the visual contents (features) of the image like color, texture and shape. Effective CBIR is based on efficient feature extraction for indexing and on effective query image matching with the indexed images for retrieval. However the main issue in CBIR is that how to extract the features efficiently because the efficient features describe well the image and they are used efficiently in matching of the images to get robust retrieval. This issue is the main inspiration for this thesis to develop a hybrid CBIR with high performance in the spatial and frequency domains. We propose various approaches, in which different techniques are fused to extract the statistical color and texture features efficiently in both domains. In spatial domain, the statistical color histogram features are computed using the pixel distribution of the Laplacian filtered sharpened images based on the different quantization schemes. However color histogram does not provide the spatial information. The solution is by using the histogram refinement method in which the statistical features of the regions in histogram bins of the filtered image are extracted but it leads to high computational cost, which is reduced by dividing the image into the sub-blocks of different sizes, to extract the color and texture features. To improve further the performance, color and texture features are combined using sub-blocks due to the less computational cost.

In the frequency domain, the statistical quantized histogram texture features are extracted from 8×8 DCT (Discrete Cosine Transformation) blocks and effectiveness of CBIR is studied based on: median and Laplacian filters, distance metrics, different combination of features, combination of texture features in both domains and combination of color and texture features in both domains are presented in order to get an efficient hybrid CBIR. Experimental results using benchmark Corel database have been shown that the proposed approaches achieve an average accuracy of 82% in the spatial domain and 86% in the frequency domain and the improved performance of proposed approaches outperform the approaches in the related works in the literature.

ABSTRAK

Penterjemahan imej berasaskan kandungan atau content-based image retrieval (CBIR) akan menghasilkan imej serupa yang digunapakai dalam kandungan visual seperti cirri-ciri warna, tekstur dan juga bentuk. CBIR yang baik adalah berpandukan kepada keberkesanan penghasilan ciri-ciri untuk kerja-kerja indeks dan juga imej kandungan yang menyamai imej indeks yang sedia untuk diterjemahkan. Walaubagaimanapun, isu utama dalam aplikasi CBIR adalah bagaimana untuk menghasilkan ciri-ciri yang bertepatan dengan imej yang dikehendaki dan digunapakai dalam penjodohan imej untuk keberhasilan penaksiran yang robust atau mantap. Isu ini merupakan sumber inspirasi utama dalam kajian thesis kerana ia bertujuan untuk pembangunan model hybrid CBIR berprestasi tinggi dalam ruang spatial dan domain-domain frekensi. Kami telah mencadangkan beberapa pendekatan yang menggunakan gabungan teknik-teknik yang berlainan untuk menghasilkan ciri-ciri warna dan tekstur yang tepat di kedua-dua bahagian domain. Pada domain spatial atau ruang, ciri-ciri histogram warna secara statistik diolah menggunakan taburan piksel melalui tapisan imej jelas Laplacian berdasarkan skema-skema kuantum yang berlainan. Namun begitu, histogram warna tidak berupaya memberi sebarang informasi spatial atau ruang. Jalan penyelesaiannya adalah melalui pembaikpulihan teknik histogram yang mana ciri-ciri statistik pada selang-selang histogram yang berbilang atau bins histogram bagi imej yang telah ditapis dihasilkan. Hal ini akan membawa kepada kos komputasi yang tinggi, yang akhirnya akan dikurangkan melalui pembahagian imej kepada sub-sub blok yang terdiri daripada saiz yang berbeza-beza untuk menghasilkan ciri-ciri warna dan tekstur itu tadi. Selain dapat meningkatkan prestasi yng lebih baik, ciri-ciri warna dan tekstur yang digabungkan menggunakan sub-sub blok adalah kerana kos komputasi yang digunapakai adalah kurang dan lebih menjimatkan.

Dalam frekuensi domain, ciri-ciri tekstur histogram kuantum secara statistik adalah dihasilkan daripada blok 8×8 DCT (Discrete Cosine Transformation) dan ketepatan CBIR dikaji berdasarkan; median dan tapisan Laplacian, jarak antara elemen-elemen dalam satu matrik atau matrik jarak, gabungan ciri-ciri yang berlainan, gabungan ciri-ciri tekstur serta kombinasi warna dan tekstur pada kedua-dua jenis domain bagi mendapatkan CBIR hybrid yang tepat. Keputusan eksperimen menunjukkan bahawa penggunaan pangkalan data Corel sebagai penanda aras telah membuktikan yang kajian ini telah Berjaya memperolehi purata ketepatan sebanyak 82% pada domain ruang aau spatial dan 86% dalam domain frekuensi. Peningkatan prestasi juga didapati adalah lebih memberangsangkan daripada teknik-teknik yang telah digunapakai dalam kajian-kajian terdahulu yang berkaitan dengan hasil penyelidikan thesis ini.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

CBIR	Content-Based Image Retrieval
DCT	Discrete Cosine Transformation
RGB	Red Green Blue
DFT	Discrete Fourier Transformation
JPEG	Joint Photographic Experts Group
MPEG	Moving Picture Experts Group
YUV	Luma (Y) and two chrominance (UV) components
SAD	Sum of Absolute Difference
SSAD	Sum of Squared of Absolute Difference
BTC	Block truncation coding
HSV	Hue Saturation Value
GA	Genetic Algorithm
CH	Color histogram
HE	Histogram Equalized
CCV	Color Coherence Vector
VCAD	Vector Cosine Angle Distance
SOM	Self-Organizing Map
FCH	Fuzzy Color Histogram
MSHP	Most Similar Highest Priority
CHRM	Color Histogram Refinement Method

CHAPTER 1

INTRODUCTION

1.1 Motivation

The visual information of an image is worth more than a thousand words. The image can describe an event, story, accident, location, experiment, any process etc., at just a glance. Nowadays, a huge collection of digital images is collected due to the improvement in the digital storage media, image capturing devices like scanners, web cameras, and digital cameras, and the rapid development in the Internet. Due to these reasons, there is a need for an efficient and effective retrieval system to retrieve these images for visual information in many professional fields like medical, medicine, art, architecture, education, crime preventions, fashion, news media etc. (Pass and Zabih, 1996; Rani and Saravanan, 2011). For example in crime prevention, the similar image with complete information will be retrieved for a suspected person's image. In the 1970's, the first approach was text-based. In this approach, images are manually annotated and are retrieved by key words. But this approach has two drawbacks: the first is that annotating a huge number of images manually is not efficient and this requires a lot of human labor. The second drawback is the different subjective perception of humans; for example, the lily flower can be annotated as water lilies, flowers in a pond, floating flowers or any other biological name. The performance of the text-based image retrieval depends upon the best annotation which reflects the actual meaning and description of the image. Usually the annotation depends upon the fairness, level of view, like, dislike and purpose of annotators. Moreover, mostly the annotations represent the partial content of images. However, searching for the images using the text-based approach is still very popular among users along with the retrieval of images by using the conventional database rules based on key words, for example Yahoo[1] and Google[2]. Considering the disadvantages of the text-based approach, the desired complete, efficient and effective retrieval of similar images will not be achieved (Liu et al., 2007).

1.2 Introduction to Content-Based Image Retrieval (CBIR)

The image consists of rich contents like color, texture and shape. In order to utilize these contents or features for the efficient and effective retrieval of images and to overcome the disadvantages of keyword-based retrieval, in the 1980's, another approach emerged; it is called the Content-Based Image Retrieval (CBIR) (Pass and Zabih, 1996). CBIR automatically retrieves similar images by using the visual contents such as color, shape and texture instead of keywords (Shan and Liu, 2009).

Many CBIR systems have been developed under different categories such as commercial, production, research and demonstration. Some examples are: QBIC, ADL, BDLP, Virage, AltaVista, SIMPLIcity and VisualSEEk. A detailed survey is given by (Veltkamp and Tanase, 2002).

A CBIR system can be divided into two steps: feature extraction or indexing and similarity measurement or searching as shown in Fig.1.1. In the first step of indexing,

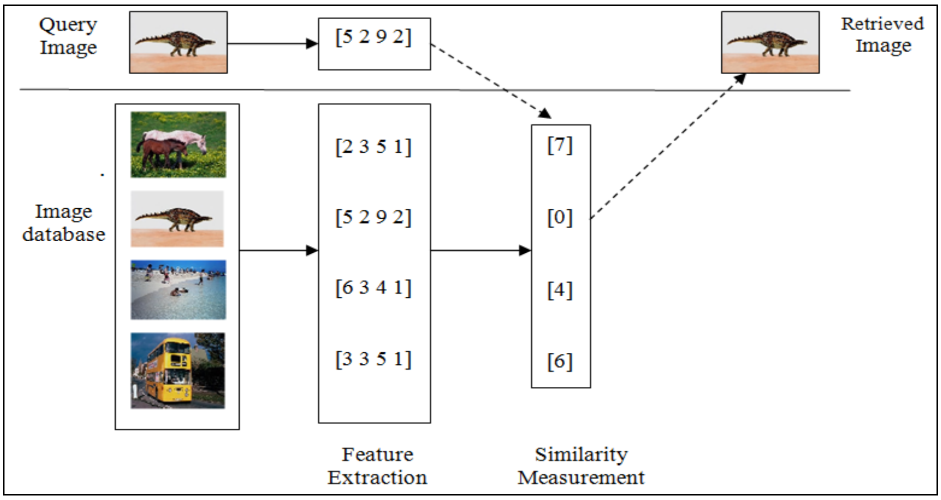


Figure 1.1 The sample abstract block diagram of the CBIR system.

[1] http://images.search.yahoo.com/search/, last visit was on April 1, 2012.
[2] http://www.google.com.my/, last visit was on Aril 1, 2012

the low level features are extracted from the images by using certain techniques and then these features are represented in a form called a feature vector. These feature vectors of all the images are stored in a database. In the second step, similarity is measured by calculating the distance between the query image feature vector and the feature vectors of the database images using a distance metric like the Euclidian distance. If the distance is zero or smaller then that image is relevant to the query image (Rani and Saravanan, 2011).

1.3 Applications of CBIR

The efficient and effective CBIR systems can be applied to a wide range of professional fields. They can play a very vital role in the retrieval of the images for visual information. The important fields where CBIR can play an important role (Gudivada and Raghavan, 1995; Eakins and Graham, 1999) are:

- Crime prevention
- The military
- Trademark and Copyright Prevention
- Architectural and engineering design
- Fashion and interior design
- Journalism and advertising
- Medical diagnosis
- Agricultural and Remote Sensing
- Cultural heritage
- Education and training

- **Crime Prevention**

The profile of the past suspects is recorded by crime prevention agencies which include fingerprints, shoeprints and facial photographs. The CBIR searches the database for the matching of the evidence (fingerprints, photo) of the criminals when a crime is committed and whenever the matches are successful, then the whole record of the criminal is accessed for identification.

- **The Military**

CBIR is used very effectively for military applications, for example, in the identification of targets and aircrafts of the enemy from radar satellite to hit the aircraft or cruise missiles.

- **Trademark and Copyright Prevention**

Trademark identification is a principal area of application of CBIR, where the existing trademark is matched with a new trademark to avoid any jeopardy of cheating. The violation of copyright of the electronic images over the Internet can be controlled by CBIR. The owner of the images can recognize the unauthorized transfer of copies of the images.

- **Architectural and Engineering Design**

The 2D and 3D models of objects of engineering and architecture are stored in a database with features. These models can be retrieved by using CBIR techniques for the training of new clients or students. Moreover, to design a new model using existing models, CBIR can retrieve similar or relevant models.

- **Fashion and Interior Design**

In fashion and interior, to support a particular design development, a meticulous combination of color and texture can be searched and retrieved from a database of fabrics by using CBIR techniques. Similarly, by giving some fabric samples, the desired pattern can be retrieved

- **Journalism and Advertising**

A huge number of photographs are collected and maintained by newspaper and advertising agencies. These photos can be used in newspapers and advertisement effectively. Keyword searching will be very costly due to the meaningful annotation of such a huge number of images. To achieve the solution of this problem, CBIR can be used efficiently and effectively.

- **Medical Diagnosis**

In most of the hospitals, a huge number of medical images are stored. The major role of these images is to display the medical images concerning a particular patient; moreover, these images also play a very vital role in diagnosis by comparing the images stored in a database of the past patients using the CBIR techniques. For example, the medical image retrieval will help the medical doctors in diagnosing brain tumors by matching the current new patient image with the existing similar effected images.

- **Agricultural and Remote Sensing**

CBIR can also be used very effectively by agriculture and geographical systems. The remote area images can be utilized widely by geographers and agriculturists for research as well as for practical purposes. Using CBIR techniques, the areas having diseased crops can be retrieved and identified. The government can also be informed about lands for which the amount has been paid to the farmers for developing crops while those lands are lying empty.

- **Cultural Heritage**

A collection of paintings, statues and heritage images is maintained by museums and art galleries. These images can provide useful historical visual information to archeologists, painters and researchers using CBIR. They can retrieve images according to their interests and desires by giving some example images.

- **Education and Training**

The images can provide very useful visual information in education and training. Those key points which cannot be explained during lectures can be identified in retrieved images; for example, safety in jungles, safety on mountains and diseased crops. These can improve the teaching quality and training.

1.4 Problem Statement

Recently a huge number of images are available due to the development in Internet and hardware devices of image capturing. Therefore there is a need of a CBIR having an efficient and effective feature extraction to retrieve these images for many fields of life like arts, education, engineering and medical etc. Features can be statistical or structural (Selvarajah and Kodituwakku, 2011). In recent years the statistical features formulation has been improved and new derived features have been developed. Many CBIR systems have been developed using different features in the spatial and frequency domains. However some have lack of efficiency in terms of accuracy (Hiremath and Pujari, 2007; Murala *et al.*, 2009; Thawari and Janwe, 2011) and other have high computational complexity (Kavitha *et al.*, 2011).

Individual features cannot describe completely the properties of images which can affect the performance of CBIR. Hence different features which are extracted by using various feature extraction techniques, can be combined together to improve the retrieval performance (Deselaers et al., 2007). However the main issue in CBIR is that how to extract the features efficiently because the efficient features describe well the image and they are used efficiently in matching of images to get robust retrieval. This issue makes the CBIR's retrieval results unstable and inefficient. Therefore this issue is the main inspiration for this study. In the present study different techniques are fused together to extract statistical color and texture features efficiently in the spatial and frequency domains. Then these features are combined to get an efficient hybrid CBIR to give the robust retrieval up to near optimal accuracy. However feature extraction in the spatial domain has low computational complexity.

1.5 Research Objectives

In this study, in order to improve the efficiency of CBIR, various approaches are proposed in which different techniques are fused together in the spatial and frequency domains to design and implement hybrid CBIR.

In the spatial domain the first approach is based on Laplacian filter using color histogram; the second one is based on sub-blocks of different sizes for color features;

the third one is based on sub-blocks of different sizes for the texture features; the fourth one is based on the median filter, median filter with edge extraction method and Laplacian filter using color histogram refinement method; and the fifth one is based on the integration of color and texture features using sub-block methods.

In the frequency domain to extract the quantized histogram texture features from DCT blocks the first approach is based on the median filter, median filter with edge extraction method and Laplacian filter; the second one is based on various distance metrics (for similarity measurement); the third one is based on different combinations of texture features; the fourth one is based on the integration of sub-block method with histograms of DCT blocks for texture features; and the fifth one is based on the integration of sub-block method with histograms of DCT blocks for color and texture features. The main objectives of this research work are:

- To develop an efficient hybrid CBIR in the spatial domain by using histogram, histogram refinement method and sub-block methods based on filters, different quantization schemes and sub-block of different sizes to extract color and texture features.
- To develop an efficient hybrid CBIR in the frequency domain by using the DCT blocks to extract quantized histogram statistical texture features based on filters, distance metrics and combination of features.
- To develop a near optimal hybrid CBIR by combining color features in the spatial and texture features in the frequency domains.
- To implement and evaluate the proposed approaches based on filters, different quantization schemes, sub-block methods using benchmark Corel database.

1.6 Scope of the Thesis

Improving the performance of CBIR as compared to the performances of the existing approaches in the literature; is the main scope of the thesis and it is based on the efficient indexing and retrieval of the images. The performance of CBIR is greatly affected by the high computational cost of the feature extraction for indexing of the images as well as by the inefficient image retrieval. The main purpose of the development and implementation of the proposed approaches is based on efficient indexing and retrieval in the spatial and frequency domains by using the statistical color and texture features. The retrieval is based on the overall similarity of images not on the object-based.

The proposed approaches are independent of the dimension of the query and target database images. Moreover any new image can be used as a query image other than the indexed database images to return similar images. The approaches give robust retrieval to the images with simple, bright and smooth textured background having less number of objects in foreground, especially one or two simple objects. Hence these approaches can be applied for the target identification and finger prints.

The proposed approaches are invariant to the rotation of the images. By changing the rotation of the image at different angles like 0°, 75°, 90° and 180°, the results of the retrieval are not so affected.

The retrieval of the images with complex background and multiple objects in foreground semantically is out of the scope of this research work.

1.7 Thesis Contribution

The main contribution of this thesis is the development of hybrid image retrieval in the spatial and frequency domains by combining various techniques in different approaches to extract color and texture features efficiently. The contributions of this thesis include:

- Development of an efficient hybrid CBIR in the spatial domain by using histogram, histogram refinement method and sub-block methods based on filters, different quantization schemes and sub-block of different sizes to extract color and texture features.
- Development of an efficient hybrid CBIR in the frequency domain by using the DCT blocks to extract quantized histogram statistical texture features based on filters, distance metrics and combination of features.
- Development of a near optimal hybrid CBIR by combining color features in the spatial and texture features in the frequency domains.

- Implementation and evaluation of the proposed approaches based on filters, different quantization schemes and sub-block methods using the benchmark Corel database.

The related works about: the different features, features extraction, retrieval domains, extraction of color and texture features in the spatial and frequency domains, distance metrics for similarity measurement, performance evaluator metrics and data set, is reviewed in Chapter 2.

1.8 Organization of the Thesis

This chapter introduces and explains the importance and applications of CBIR, problem statement, research objectives, scope and contributions of the research work in this thesis. The rest of thesis is organized as follows. In Chapter 2, we describe the methods and techniques used for CBIR. Commonly used image features, their types, and domains, extraction of color and texture features in the spatial and frequency domains are discussed with techniques used for their extraction. Similarity measurements for matching of features, performance evaluation measurements of research work and the benchmark dataset for testing of proposed approaches are discussed.

In Chapter 3, we elaborate the methodology of the proposed approaches in the spatial and frequency domains. In spatial domain the approaches based on: histogram, histogram refinement method and sub-blocks methods to extract color and texture features. In the frequency domain the quantized histogram texture features are extracted in the proposed approaches based on: filter methods, distance metrics and combination of features in both domains.

In Chapter 4, we discuss the implementation and the experimental results of the proposed approaches in spatial and frequency domains. Experimental results in spatial domain, are analyzed on the basis of different quantization schemes, filter methods and sub-blocks of different sizes, while in frequency domain, results are analyzed on the basis of different quantization schemes, filter methods, distance metrics, combination of features.

In Chapter 5, the retrieval performance of our proposed approaches is compared among themselves first and then the near optimal approach of them having combination of the color and texture features is compared with the state-of-the-art work in the literature using the Corel database.

In Chapter 6, we summarize our contributions, and discuss the future work.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW OF CBIR

2.1 Content-Based Image Retrieval (CBIR)

An image consists of rich contents like color, texture and shape. These contents can be used in matching of query image with the target images of database for retrieval of similar images. Thus retrieval of images by using the contents of images is called content-based image retrieval (CBIR). The features of an image are extracted by using certain techniques as a first step of CBIR. The extracted features are then represented in such a manner that they will be used in similarity (matching) measurement of the query image with the target images of database.

The development of an efficient and effective CBIR has two major challenging problems due to its nature of job to perform: (a) how to describe an image mathematically; (b) how to perform matching between two images using their extracted descriptions. The first challenge is due to the fact that the image is basically nothing more than an array of pixels. To describe the image for retrieval purposes, these pixels are represented mathematically. These mathematical descriptions can be then formulated in such a way that the similarity measurement can be determined to get a high retrieval of the similar images.

In recent years the improvement has been made to the construction of the mathematical description of image and based on these descriptions, new derived features of an image have been created. Along with the advance development of the mathematical formulation, new methods have been introduced for the similarity measurement. At present a very strong tendency has been introduced to CBIR for the extraction of mathematical features from an image, called the statistical methods. These statistical techniques extract features by formulating the distribution of the pixel values in an image. In this section, we will first review various image features and then the feature similarity methods.

A feature can be defined as: a value that represents a certain visual property of the image (Roumi, 2009). Color, texture and shape are widely used features of image. These existing features are divided into global and local features (Deselaers *et al.*, 2008).

2.1.1 Examples of Some CBIR Systems

CBIR systems are applied to various professions for the retrieval of the images with purposeful visual information. For example, face recognition, finger and foot prints in crime prevention for security, medical diagnosis to identify and retrieve the similar past images from the medical image databases to the input query patient image, trade mark identification, designing of new models in engineering etc.(Gudivada and Raghavan, 1995; Eakins and Graham, 1999). For the last decade many CBIR systems have been developed for demonstration, research, experimental and commercial purposes. Among existing CBIR systems, some are described below.

- **SIMPLIcity**

The main purpose of this image retrieval system is to retrieve images for picture libraries and biomedical image databases. In this system texture, color and shapes features are extracted. For the similarity measurement the Integrated Region Matching (IRM) metric is used and it also consists of methods for the classification of the images. In experiments this system is tested by using the large-scale picture libraries and a database of pathology images. This system provides fast retrieval with the high accuracy and it has high robustness to variations of an image (Wang *et al.*, 2001; Kosch and Maier, 2009) .

- **ImageFinder**

ImageFinder uses shape features of regions and gives dynamic platform for search in which users can make changes according to their requirements. However, it has overall poor retrieval performance for all the user queries (Kosch and Maier, 2009).

- **Caliph & Emir**

It consists of two systems, Caliph and Emir. Caliph employs annotation of the images while Emir searching of the images. They are using Color Layout and Edge Histogram features. Caliph describes semantic graphs for annotation to improve the search. Emir provides good visual similarity search. Emir extracts features in the form of MPEG-7, to be maintained by Caliph as MPEG-7 documents (Kosch and Maier, 2009).

- **VIPER/GIFT**

VIPER provides searching of the images and is a plug-in part of the GIFT system. For documents up to 80,000 features like terms in documents and for an image about 1000 features are extracted. Color and texture features are extracted. For color features extraction, histogram and sub-blocks of the images are used while for the texture features, the Gabor filters are used (Kosch and Maier, 2009).

- **SIMBA**

SIMBA uses color and texture features. The histogram features are invariant against the rotations and translations of the image. Kernel functions are used in the extraction of features. The size of the objects in an image is affected by the size of the kernel for the recognition in the images. The big kernel features are invariant to the rotation and translation of the bigger objects (Kosch and Maier, 2009).

- **Picture Finder**

Picture Finder uses color, texture and shape features along with the keywords. Polygons are characterized in an image by using the different size of shape, color and texture features. The similarity for search is affected by the position of the polygons. Hence two images will be more similar if the position of the polygons is same. However for the robust similarity measurement the size, color and texture features of the polygons are very important (Kosch and Maier, 2009).

- **QBIC (Query By Image Content)**

This CBIR system has been developed by IBM Almaden Research Centre. This system provides the flexibility to improve the queries graphically by changing the features of an image like texture, shape and color. The user queries can be an example image, draw sketches or select texture and color patterns. The Quadratic and weighted Euclidean Distances are used to compute the distance between the query and target database images (Niblack *et al.*, 1993).

- **VIR Image Engine**

This system is the extended version of the system developed by Virage Inc. This system provides the retrieval to the user query image by using the features of an image like color, texture and structure. For the extraction of features, pixel analysis process is performed in the images. distance function is used to compute the distance between the images (Bach *et al.*, 1996).

- **VisualSEEk**

This system has been developed by the Image and Advanced Television Lab, Columbia University. In this system, matching of the images is based on texture, color and spatial location features. In the query the user can draw the sketches of the regions with proper color, dimension, position, size, location and spatial relationships between the regions (Smith and Chang, 1997).

- **NeTra**

This system is based on the region-based features and developed in the University of California by the Department of Electrical and Computer Engineering. In this system the image is segmented into a number of regions and from those regions texture, color, shape and spatial location features are extracted (Netra, 1997; Ma and Netra, 1999).

- **MARS (Multimedia Analysis and Retrieval System)**

This system is based on the primitive features like color, spatial layout, texture and shape features. It was initially developed at the University of Illinois by the

Department of Computer Science. However, it is then further enhanced at the University of California by the Department of Information and Computer Science. Using the Boolean operators, it supports the formulation of the complex queries of the users. The desired query image can be direct or by example (Ortega *et al.*, 1997).

2.1.2 Global Features

The features which are computed from the overall image are called global features, which represent the characteristics of the whole image. For example in the color histogram technique, the whole grayscale image is divided into some number of bins like 16, 32 and 64, and in each bin mean of the color pixels distribution is computed. These calculated mean values of the all bins of an image are represented in a feature vector of color values. The global features are extracted to confine the overall properties of the images. For example an image is divided into sub-blocks using the color layout method and the color pixel values are calculated to get average values of all sub-blocks. These average values are represented in a vector form where each value indicates the location of sub-blocks. Since the global features are extracted from the overall image, not only the extraction of features is efficient but the computation of similarity is also efficient (Datta et al., 2008).

The color histogram (Swain and Ballard, 1991), color moments (Stricker and Orengo, 1995), edge histogram (Won *et al.*, 2002) and color correlograms (Huang *et al.*, 1997) are well known techniques for the extraction of the global features.

2.1.3 Local Features

Local features (Heng and Qing, 2008) are computed from the every block or patch of image when it is divided into some particular blocks or patches. Every block or patch of an image has unique visual information. The similarity can be measured by using the local features of the blocks to retrieve the image on the basis of some local visual information in the image.

In spatial domain, to increase the retrieval robustness, the local features are extracted such that using the neighborhood pixel values of a pixel to calculate a set of features but this approach will increase the computational cost. The computational complexity can be reduced by dividing the image into non-overlapping small sub-blocks and local features are calculated for all sub-blocks (Datta et al., 2008).

The main purpose of the local feature extraction is to recognize the particular objects in an image. The popular methods which are used for the extraction of the local features are: global search (Deselaers *et al.*, 2008), local signature (Mikolajczyk *et al.*, 2005) and local histograms (Deselaers *et al.*, 2005).

In global search method the top N nearest neighbor features are searched globally in the entire target database images for the query image features. The number of features of the target images is computed and sorted which have close matching with the query image. Thus if the target image has more number of features then it is most similar (Deselaers *et al.*, 2008). In the local signature method the local features are clustered for all the images and matching is performed by using mean and covariance of the clusters of the query image with target image (Mikolajczyk *et al.*, 2005). In the local histogram, all the features of the image are clustered to get histograms of all features in an image (Deselaers *et al.*, 2005).

2.2 The Most Commonly used Features for CBIR

The most commonly used features for CBIR are color, texture and shape which are described as:

2.2.1 Color Feature

Color information is the most prominent and simple visual feature of an image (Smeulders et al., 2000) since it is the dominant part of the human visual perception. It has characteristics of robustness in the background of the image, which has some complexity, and the variation of size and orientation of an image do not affect it (Park et al., 2007).

Color is the most useful feature due to the simple implementation, robust retrieval, efficient computation and need of low storage space, and used by about all CBIR systems. Color is represented by LUV and HSV color space, because they are near with the human perception (Muller *et al.*, 2004).

Initially, in the early days, the color feature is used in the matching of the images for the similarity measurement such that the pixel values of the query image is compared with the corresponding pixel value of the target image (Kato et al., 1991) but the drawback is that changes in the direction, noise and illumination of the images, caused a great dissimilarity between the images. This problem is solved by (Swain and Ballard, 1991) by introducing the color histogram for the first time. The color histogram is the most popular and widely used method for the extraction of color features of the images in CBIR (Hafner *et al.*, 1995). A color histogram extracts the global color features from the image. It is very simple and easy to compute. It has robustness and efficiency to the indexing images in a large database. The color histogram consists of the frequency of the occurrences of each pixel value which represents the color of an image in the spatial domain. For the extraction of the color features, the color histogram is divided into a set of bins of color and each pixel having a specific color, belongs to a color bin of that color. It has the characteristic that it represents the global information of an image (Swain and Ballard, 1991; Jin, 2009). These global features representations of an image are very useful in the queries in which the matching of the images is based on the overall appearance.

The color histogram is very fast in the computation of color features (Park et al., 2008). It has no effect on the small changes in the scenes. It is useful and widely used for the images which require invariance in the translation and rotation (Park et al., 2010). If a query requires retrieving the images with the same scenes but with the different circumstances of illuminations, then the color histogram is not a suitable technique for such queries. The reason is that the same histograms are generated for the images having different appearances due to the lack of spatial information in the histograms. In other words, an image with many very small green spots has a histogram similar with the histogram of the image which has a single green area (Jin, 2009). However, a color histogram does not provide the spatial information about the location of the pixels and the relationship of a pixel with the other neighboring pixels. To preserve the spatial information in a histogram, the color histogram refinement method is used to classify the pixels in the coherent and incoherent clusters. If a pixel belongs to a region of a similar color and considerable size, then it is known as coherent and in the other case it is known as incoherent. The coherent features like mean, variance, size, and major and minor axis lengths of different clusters are computed. This method is also called the color coherence vector (CCV) (Park et al., 2008).

The spatial relationship can be maintained by using the correlograms technique, which depicts the frequency of the number of occurrences of the two pixel values having a spatial distance. The color correlograms features are extracted by quantizing the image into bins and the frequency of the pair pixels is counted at the neighboring pixels (Huang et al., 1997).

Color features have been extracted using several methods other than the color histogram in an image retrieval like color moments and color sets. A method has been proposed by (Stricker and Orengo, 1995) using the color moments to triumph over the histogram quantization. In this method the statistically color features are computed such that the color distribution is characterized by the first moment (mean), second moment (variance) and third moment (skewness).

For combination of the color and spatial information, many methods have been developed for the queries in CBIR (Mustaffa *et al.*, 2008). As retrieval, based on the segmentation, has high computational complexity for large image database, to reduce the computational cost and get the spatial information, the image is divided into nine equal sub-blocks and color histogram are constructed for all sub-blocks (Gong *et al.*, 1996). A method is developed by (Stricker and Dimai, 1996), to split the image into five non-overlapping spatial regions to compute color features in all the regions and to be used in matching of the images. A novel algorithm is proposed by (Li, 2003) in which an image is divided into sub-blocks to get color features. Recently, various methods have been introduced in which different color features are extracted such as chromaticity moments based on the regular histograms (Paschos *et al.*, 2003) and fuzzy color histogram (Han and Ma, 2002). Corresponding region-based color features are extracted and used in matching of the images to retrieve similar images (Thomas *et al.*, 2008).

2.2.2 Texture Feature

The visual information of an image which has some repeated patterns of the pixel values in some proper arrangement is called the texture of the image (Tuceryan and Jain, 1993). The texture features can be seen easily in the natural images like flowers, leaves, petals, brick walls etc., separating one area from another. Having significant characteristics for image representation by texture, such as directionality, coarseness, roughness, smoothness, granularity etc can be very useful for the image retrieval. These properties can play a very important role in the classification of the images. Thus for the effective image retrieval, the most relevant and significant texture features can be extracted and then these features can be represented in an effective vector form to measure the similarity effectively (Baaziz et al., 2010). The texture features can be extracted either directly from the coefficients of the transformed images or by computing statistically from the coefficients (Li et al., 2000; Do and Vetterli, 2002).

Statistical second or high order moments are usually combined with the filtering methods like the Gabor filter or wavelet to extract the texture features. To extract one type of texture feature from one sub-band of a filtered image, the filter approach is used; while, to extract different texture features from many sub bands of an image for classification, statistical features are computed in all sub-bands (Hideyuki *et al.*, 1978; Park *et al.*, 2002). The statistical texture features which represent the properties of an image such as coarseness, constant, directionality, line-likeness, regularity and roughness, are computed and called Tamura features (Hideyuki, Shunji *et al.*, 1978). Regarding the human perception, the first three texture features are considered important after having gone through a set of experiments by (Deselaers *et al.*, 2008) and these three features are extracted by using the histogram method.

The Gabor filter consists of rows and columns of values which are used to compute important information from image data in different directions either by calculating the mean and standard deviation of filtered values or by quantizing the histogram into bins for each filter (Squire et al., 1999). For medical, aerial and images having texture patterns, texture features show good results in terms of retrieval; moreover for generic images, the texture features are combined with the other features for good image retrieval (Goldberger et al., 2003).

The coarseness and recurring patterns of the surface of the images are captured as the texture features for example flowers, leaves, petals; brick walls have textures features as pattern, coarseness and smoothness. These texture features play an important role in the specific domains of image retrieval like medical imaging and aerial imagery due to the close association with the human semantics. For the last decades the texture features have been used as important features like filter banks (Malik and Perona, 1990) and wavelet transforms (Unser, 1995) in the fields of computer graphics, computer vision and image processing (Haralick, 1979). The texture features in the image processing, are extracted by using directly the coefficients of transformed image or by computing the coefficients statistically using discrete cosine transformation (DCT) or Wavelet Transformation (Li et al., 2000; Do and Vetterli, 2002). In computer graphics and vision Markov statistical texture features are computed by using the wavelet coefficients in different scale and orientations of a transformed image (Portilla and Simoncelli, 2000) In early days of image retrieval using the texture features based on the texture descriptors of (Manjunath and Ma, 1996). Texture features are included in MPEG-7, computing important visual information in the standard numerical formats (Manjunath et al., 2001). For the retrieval of aerial images, a texture features thesaurus is proposed by (Ma and Manjunath, 1998). To extract the texture features using this thesaurus, statistical texture features are computed using the Gabor filters (Jain and Farrokhnia, 1990). Advanced textured region features are computed like invariant photometric- and affine-transformation features (Schaffalitzky and Zisserman, 2001). Advanced affine-invariant texture feature using interest point detection for sparsity, are extracted for texture recognition (Mikolajczyk and Schmid, 2004).

The texture is represented by a vector of computed features which are extracted from the image in the statistical texture approach, while in the structural texture approach the texture is represented by a vector of texture primitives and their placement rules (Haralick, 1979; Vilnrotter et al., 1986). The statistical texture features can be computed either directly from images using histograms (Stricker and Orengo, 1995) or co-occurrence matrices (Haralick, 1979) or by using the filter methods like Gabor filters (Jain and Healey, 1998) or transformations consist of wavelets (Pun and Lee, 2003; Jafari-Khouzani and Soltanian-Zadeh, 2005), wavelet packets (Laine and Fan, 1993), ridgelets, and curvelets (Semler and Dettori., 2006).

Separate color features computed by using histograms in CIE Lab space, are combined with the texture features computed by using wavelet (Liapis and Tziritas, 2004).

2.2.3 Shape Feature

When an image consists of natural objects, these objects can be identified by shape. The shape is the most apparent feature which can be extracted from the images and used for the retrieval of an object based on overall similarity. Shape features are clearer visual features than the texture features (Biederman, 1987). Shape features can be divided into two categories, region-based and boundary-based (Loncaric, 1998; Zhang and Lu, 2001). For the extraction of the region-based shape features, the image is fragmented into different regions and then the shape features like size, area, circularity, rectangularity and variance are computed from the regions (Lu and Sajjanhar, 1999; Park et al., 2008). Contour or boundary-based shape features represent the edges and perimeters of objects. Contour-based features are extracted by using the techniques like Wavelet Fourier descriptors (Zhang and Lu, 2004). The edges can also be detected by using a canny edge detector (John, 1986; Zhao et al., 2009) and a Harris edge detector (Harris and Stephens, 1988).

Shape is an important feature of the regions of the fragmented image and plays a significant role in the retrieval due its robust and efficient characterization. Over the decades there is a move because of the limitations of the typical modeling from the representation of global shape (Flickner *et al.*, 1995) to further local shape features (Mehrotra and Gary, 1995; Berretti *et al.*, 2000; Petrakis *et al.*, 2002).

The contours are simplified by extracting the shape features using discrete curve evolution (Latecki and Lakamper, 2000). The image is enhanced by using these simple contours which remove irrelevant and noisy shape features. For matching of the images to measure the similarity, a new shape features is proposed, which is called as shape context and quite robust to a number of geometric transformations (Belongie *et al.*, 2002).

An approach is proposed by (Berretti *et al.*, 2000), to extract the shape features, a number of segments are represented as curves and these curves are arranged into a metric tree (Ciaccia *et al.*, 1997), for matching of shape efficiently and the retrieval is called shape-based image retrieval.

An approach is proposed by (Petrakis *et al.*, 2002) which is called dynamic programming (DP) in which the shape features are extracted in sequences of concave and convex segments. However the computational cost of this approach is high due to the computation of moments and Fourier descriptors.

The performance of the CBIR in terms of effective retrieval is not good by using the shape features because it is very difficult to describe the shape features which lose some vital information and it is greatly affected by noise and occlusion (Loncaric, 1998). Shape is always combined with the other features like color and texture to retrieve the similar images in CBIR.

2.3 Retrieval Domains of CBIR

In order to extract the color, texture and shape features from the images for a similarity measurement, the visual information of the image needs to be represented in certain useful forms. The representation of the image data can be divided into two groups called spatial and transformed domains.

As the image is a two dimensional matrix of pixel values, hence in the spatial domain, the features are directly computed by using the pixel values of the image. The calculated features are represented in a feature space with fewer dimensions which is more efficient in terms of computational cost and storage space (Shahbahrami et al., 2008). In the spatial domain, the distribution of the pixel values representing color can be computed statistically.

The color histogram statistical features like mean, standard deviation, skewness, energy and entropy are extracted by using the probability distribution of the intensity levels of the image (Sergyan, 2008). The statistical color features like mean and standard deviation are computed simply row and column wise in all the three planes, Red, Green and Blue (RGB), of the color image to construct feature vectors with six dimensions. To get the similarity measurement, the distance is computed between the vectors of the query and target images of the database by using the Euclidean distance (Kekre and Patil, 2009).

The texture and color features are jointly used in the spatial domain for the retrieval of similar images. The color features are computed simply by using the histograms and the texture features are calculated by using the statistical histogram features like entropy, smoothness and uniformity (Thawari and Janwe, 2011). To get the spatial relationship among the pixels of the image in the spatial domain, the color histogram refinement method is used to get the shape features of the objects by detecting the region in the image and calculating the size, major and minor axes of length, and the variance (Park et al., 2008).

Nonetheless, the data of the image is transformed from the spatial domain to the frequency domain by using certain transformation techniques like the Discrete Cosine Transformation (DCT) and the Discrete Fourier Transformation (DFT). The spatial domain has sensitivity for variation and at present, most of the images are represented in a compressed format like JPEG and MPEG (Nezamabadi-pour and Saryazdi, 2005; Liu et al., 2007). That is why most of the researchers are giving attention to the transformed domain, and the DCT based image features are extracted widely for the image retrieval. DCT is used very commonly because it is a very important component of JPEG and MPEG compression and DCT coefficients can be reconstructed in reverse easily from the JPEG images. The low level features can be extracted directly in the compressed domain without decoding to the spatial domain to reduce the computational cost (Mandal et al., 1999). The DC and AC coefficients of the 8×8 DCT blocks of a grayscale image are represented in different directions which are mapped into the feature vectors of nine dimensions (Bae and Jung, 1997). The image in the YUV color space (Luma (Y) and two chrominance (UV) components) is divided into four blocks and in each block, only the Y channel is converted from the spatial to the DCT blocks to get horizontal, vertical and diagonal texture features in all of the blocks (Tsai et al., 2006). The DC and the first three AC coefficients of all 8×8 DCT blocks are selected in a zigzag order to construct a quantized histogram of 32 bins to be used as feature vectors for retrieval (Mohamed et al., 2009).

The color and texture features are extracted in the DCT domain using the energy coefficients of the DCT blocks. The color features are extracted by dividing each of the 8×8 DCT blocks into four sub-blocks and the mean is computed in each sub-block to get the mean values from the histogram. The texture features are extracted by computing the mean and standard deviation in the histograms of the selected coefficients of the blocks (Lu et al., 2006).

2.4 Extraction of Color and Texture Features in the Spatial and Frequency Domains

In this study of research, color and texture features are extracted in the spatial and frequency domains by combing different techniques.

2.4.1 Extraction of Color and Texture Features in the Spatial Domain

Color information is widely used for CBIR by researchers for retrieval. It is a very prominent and extensively studied feature. One reason for its importance is that it is invariant to the orientation and scaling of the image (Lei et al., 1999). Color information of an image can be extracted by using the different techniques but the mostly used and prominent technique is the color histogram. It is extensively used for CBIR.

A color histogram represents the frequency of the occurrences of each color in an image. It is divided into bins, each having a number of specific color values of pixels. It shows the global characteristic of an image. It has robustness in regards to the rotation and translation. The significant characteristic of the color histogram is that it can compute the color moments, mean and standard deviation of the images efficiently with a large number of pixels of a very huge database (Shahbahrami et al., 2008). Color histograms can be generated in all three of the components, Red, Green and Blue, of the RGB color image by quantizing the histograms of each color component, into 64 bins to get the color histogram features (Murala et al., 2009).

The normalized histograms are quantized into the 48 bins in each component of the RGB color image. Thus, for each image a feature vector of the total 48×3=144 features, is created. For the similarity measurements, the Euclidean distance is used to calculate the distance between the query image feature vector and the database image feature vectors. The images are ranked by the similarity distance values (Chakravarti and Meng, 2009).

The color features of an image are computed by using the histogram technique (Thawari and Janwe, 2011). The color histogram features are extracted from the images for CBIR to retrieve or classify the images in the database according to the user query image. The histogram features are the mean, standard deviation, skewness, energy and entropy where the mean reveals brightness, standard deviation indicates contrast, skewness shows the intensity level distribution about the mean, energy describes the distribution of the intensity levels in an image and entropy represents the distribution of the pixel values in the intensity levels (Sergyan, 2008).

A method is proposed by (Park et al., 2010) in which Global and local color features are extracted. The global color features are extracted by generating histograms in the RGB color space while for the local features the genetic algorithm (GA) is used in the HSV color space.

The main problem in a histogram is that the spatial information in the color histogram is not preserved and thus the same histograms will be extracted for the images with different looks. For this purpose the histogram technique has been modified to get an improved and refined histogram. This method is also called color histogram refinement. This refined histogram method divides the histogram into buckets of pixels and each bucket is divided into classes of pixels of same local properties. The histograms of query image are compared with the database images of pixels of the same local properties in the buckets (Liu et al., 2007).

The color histogram refinement method is used to get the color and shape features of the objects in an image. The histogram of the grayscale image is quantized into bins. In each bin similar colored connected regions are determined. The number of coherent and incoherent pixels in each region of each bin is determined. A pixel is coherent if it is present in the same colored region otherwise is incoherent. The number and average of the coherent and incoherent clusters are calculated. Additional features are also computed but only in the coherent clusters. These features consist of the sizes of largest, median and smallest clusters, the major axis length, minor axis length, ellipse angle and variances of the largest, median and smallest clusters. These features are not affected by the orientation of an image. For distance calculation, the Euclidean distance is used to retrieve the images (Park et al., 2008).

Most of the images consist of some noise and unwanted information. These should be removed from the images by using filters before processing for the retrieval. Different filter methods can be used for the removal of noise. A median filter is applied on images for enhancement as a preprocessing step. Though this filter improves the image quality, it creates another problem, that some amount of the edge information of the objects in the images is lost. This edge information is recovered by applying an edge extraction method. Then the histogram features are extracted from the enhanced filtered image by quantizing the histogram into bins and in each bin the average of the pixels is computed which are combined to form a feature vector for the retrieval of images. The results showed that the median filter with the edge extraction method gives good results (Zhao et al., 2009). In our research work some approaches are based on some filters like Laplacian and median filters.

To reduce the computational cost sub-blocks of images can be used instead of histogram. A method is proposed by (Qiu, 2003) in which block truncation coding (BTC) technique is used for improving the effectiveness of CBIR. In this method, the image is simply divided into non-overlapping blocks, especially in 4×4 pixels for high performance. The mean value of each block is computed and then the mean value is compared with each pixel of the block such that if the mean value is less than or equal to the pixel value, then the pixel position is replaced by 1, otherwise by 0. One mean value is calculated for the first condition and the second for another condition. In decoding for each block, the pixel position 1 is replaced by the first mean and 0 by the second mean value, which is used in the computation of the color features.

For an effective CBIR, the image is divided into equal sized sub blocks. Each block of the HSV color space image, is quantized to get the color histogram features of that block (Kavitha *et al.*, 2011).

Color moments represent the distribution of the color information in an image. Color moments have been used successfully in various CBIR systems for the retrieval of similar images, for example in QBIC (Veltkamp and Tanase, 2002). The statistical values, mean, variance and skewness are calculated using the color pixel values of the image to describe the color feature distribution in the image (Balamurugan *et al.*, 2010). Color moments characterize the color image to get the color features for retrieval of similar images. Color moments consist of the mean, which is the first-order moment; the variance, which is the second-order moment and the standard deviation, which is the third- order moment. These represent the distribution of the color pixel values in an image (Dubey *et al.*, 2010).

Red, green and blue are the three color components of the RGB color image. In each of the components of the image, the color moments, the mean, variance and standard deviation, are computed. These moments are calculated column-wise and row-wise in the three components to be used as the color features to retrieve similar images (Kekre and Patil, 2009).

In spatial domain to increase the retrieval robustness the local features are extracted such that using the neighborhood pixel values of a pixel to calculate a set of features but this approach will increase the computational cost. The computational complexity can be reduced by dividing the image into non-overlapping small sub-blocks and local features are calculated for all sub-blocks (Datta et al., 2008).

For combination of color and spatial information many methods have been developed for the queries in CBIR (Mustaffa et al., 2008). A method is developed by (Stricker and Dimai, 1996), to split the image into five non-overlapping spatial regions to compute color features in all the regions and to be used in matching of the images. A novel algorithm is proposed by (Li, 2003) in which an image is divided into sub-blocks to get color features. Recently, various methods have been introduced in which different color features are extracted such as chromaticity moments based on regular histograms (Paschos et al., 2003) and fuzzy color histogram (Han and Ma, 2002). Corresponding region-based color features are extracted and used in matching of the images to retrieve similar images (Thomas et al., 2008). As retrieval based on segmentation has high computational complexity for large image database, to reduce the computational cost and get the spatial information, the image is divided into nine equal sub-blocks and color histogram are constructed for all sub-blocks (Gong et al., 1996).

Color features have been extracted using several methods other than the color histogram in image retrieval like color moments and color sets. A method has been proposed by (Stricker and Orengo, 1995) using color moments to triumph over the histogram quantization. In this method statistically color features are computed such that the color distribution is characterized by the first moment (mean), second moment (variance) and third moment (skewness).

Another feature is the texture feature which is used for the retrieval of images; it can be defined as: the area of an image described by the spatial distribution of pixel values. Texture can be extracted by using the statistical texture moments in the co-occurrence matrix technique (Partio, 2002). Texture represents the visual characteristics of surfaces like wood and fabric. Texture may be rough, smooth, coarse and rippled. It consists of vital information about the structural management of surfaces and their relationships to the neighboring environment. Since the texture properties of the images contain useful information for the classification purposes, statistical texture features of the images can be computed to retrieve the similar images (Haralick et al., 1973).

The spatial neighborhood distribution of the pixel values describe an area in an image which is called texture and can be extracted by using the statistical texture moments in the co-occurrence matrix technique (Partio, 2002). Analysis of the texture is attractive and a useful area of research because it has importance in applications like medical image processing, defect detection and remote sensing. Texture is more useful in the classification of images than other features like color or shape. Texture can be mostly found in the natural images having woods, grass, water, trees etc (Park et al., 2005). Density, contrast, coarseness and uniformity are typical texture features (Shahbahrami et al., 2008). Statistical histogram texture moments are the mean, standard deviation, energy, entropy, skewness and kurtosis. These features are calculated by using the intensity levels of the images (Selvarajah and Kodituwakku, 2011).

2.4.2 Extraction of the Statistical Texture Features in the Frequency Domain

The availability of a huge number of the images is due to the advanced development in the image capturing devices, Internet and computer hardware. As a result, most of the images at present, are represented in a compressed format like JPEG (Nezamabadi-pour and Saryazdi, 2005; Liu *et al.*, 2007). JPEG(Joint Photographic Expert Group) is a compressed format of the images with good quality (Jeong, 1997).

In the compressed domain, the features can be extracted directly without decoding into pixels by using frequency transformation like Discrete Cosine Transformation (DCT) which is used as the component of the compression process (Mandal et al., 1999). In DCT, some information from the image is eliminated and some important information is left behind during compression, which can be used to play an important role in the retrieval of the similar images (Zhong and Defee, 2005).

There are various discrete transformation like discrete cosine transformation (DCT), discrete Fourier transformation (DFT) and discrete wavelet transformation. These transformations will be prominent domains for retrieval of compressed images like JPEG. The selection will be made on the performance of transformation. In this work of research we have selected DCT for feature indexing and retrieval due to some advantages over DWT such that DCT is less expensive for hardware or software implementation than DWT, for instance the two-dimensional 8×8 DCT blocks in an efficient algorithm, involves only 54 multiplications (Feig, 1990) where as in DWT each coefficient requires at least one multiplication which increases the computational cost.

DCT is a cosine part of Fourier transformation while the DWT requires more computation in time frequency. So it is acknowledged by (Shen, 2013) that the computational cost of DWT is higher than DCT, the regions of image become blurred and noise is produced near edges of regions by using the larger DWT basis function or wavelet filters, the compression time of DWT is longer than DCT, at low compression the quality of DWT is lower than JPEG. Therefore we focus on the performance of DCT in this research work.

The DC and AC coefficients of the 8×8 DCT transformed blocks are represented in nine different directions which represent the nine feature vectors of the texture features and the grayscale level distribution in the image (Bae and Jung, 1997). In the YUV color space, the texture features are extracted such that the image is divided into four blocks and only the Y component in each block is transformed in the DCT coefficients to get vertical, horizontal and diagonal features in all of the blocks for the image retrieval (Tsai *et al.*, 2006). The DC and some of the AC coefficients are used directionally to get energy histograms which are represented as feature vectors to retrieve similar images and the approach is tested with a medium sized database (Lay and Guan, 1999). The DC vector is combined with a nine AC coefficient distribution vector to get the feature vector of the texture features of the JPEG format images. The AC coefficients define the texture information (Shan and Liu, 2009). The statistical texture features are extracted from the images in the compressed domain by computing the mean and standard deviation moments using the DCT coefficients (Feng and Jiang, 2003).

In the JPEG compressed format, the texture features are extracted by computing the central moments of the second and third order using the DCT coefficients. These features are used to form a feature vector to retrieve the similar images (Vailaya *et al.*, 1998). The quantized histograms are extracted from the DCT coefficients in the approach of (Mohamed *et al.*, 2009) such that the DC and the first three AC coefficients are selected in a zigzag order from the transformed 8×8 DCT blocks of the JPEG format images and then histograms of these coefficients are constructed with a 32 bins quantization. These histograms are used as a feature vector for retrieval. This method is tested using the animal dataset of the Corel database.

The histogram statistical Texture-Pattern is constructed using AC coefficients of each DCT block of the image and used for the image retrieval (Bai *et al.*, 2012). The histogram texture features are extracted directly from the DCT coefficients for image retrieval (Fan and Wang, 2002).

To retrieve similar images for the query image from the database, the distance metric is used for matching. To measure the distance for the similarity between the query and database images the distance metrics like Manhattan Distance (L1 metric), Euclidean Distance (L2 metric) and the Vector Cosine Angle Distance (VCAD) are used (Hafner *et al.*, 1995).

The features of the image are represented in the feature vector form which represents the object. For the similarity measurement, various methods are used to compare two feature vectors. In comparison, the distance metric measures the difference between the two vectors of the images and a small difference means that two images are the most similar. The similarity measure metric measures the similarity between the two vectors of the images and a large similarity means that two images are closely related (Sergyan, 2008).

2.4.3 Combination of Features in CBIR

The effectiveness of the CBIR can be analyzed by combining various texture features in different combinations because the individual texture features cannot describe the image completely enough to retrieve similar images.

Only a single feature among the different features like color, texture and shape, have been used in most of the early work on CBIR. However, satisfactory results are difficult to obtain by using a single feature since an image consists of different visual information. In order to get high performance of retrieval, in recent research work combination of visual features have been used (Liapis and Tziritas, 2004; Vadivel et al., 2004; Chun, 2005). In approach of (Liapis and Tziritas, 2004), color features are extracted by using 1-D(dimensional) or 2-D histograms of the CIELab chromaticity coordinates while texture features are extracted by computing variance of the discrete wavelet frames. In approach of (Vadivel et al., 2004), the color features are extracted by using color histogram and texture features are extracted by using Haar or Daubechies' Wavelet moment. The dimension of the feature vectors by combining various features in these approaches was ignored. However there was no guarantee that the retrieval accuracy would be high with low dimension of feature vectors (Chun, 2005). A novel retrieval framework is proposed by (Deng et al., 2001; Hiremath and Pujari, 2007), in which color, texture and shape features are combined. The image is divided into non- overlapping sub-blocks of equal size. The color features are extracted by computing color moments, texture features by computing moments using Gabor filter and shape features are extracted by capturing edges of the objects in the images using gradient vector flow fields.

An experimental comparison of the statistical features, such as skewness, color variance and cross correlation is performed using the color histogram (Sharma *et al.*, 2011). For the last three decades, the study of the texture features, such as the MPEG7 edge histogram, relational invariant feature histogram, global texture features, Gabor features and Tamura texture histogram have not fully described the texture properties of images. To overcome this issue, various texture features have been combined in different combinations to get high performance in terms of image retrieval (Deselaers *et al.*, 2007).

In (Veltkamp and Tanase, 2002) the HSV color space is used because this color space is closer to the human visual perception. For this purpose, the RGB color images are converted into HSV color images. Each component H, S and V is partitioned into 96 bins of the histogram. Due to the large number of computations of the histogram for the three color components, the efficiency of this process is low. The statistical texture values such as the mean, standard deviation, smoothness, third moment, fourth moment, uniformity, and entropy are calculated in each bin of the histogram in each component of the HSV. For each component 96×7=672 features and for the three components 672×3=2016 features are calculated.

A method is proposed by (Hiremath and Pujari, 2007), in which the texture, color and shape features are fused together and extracted in a non-overlapping partitioned image by using the Gabor filter, the statistical color moments and the Gradient vector flow fields. In the method of (Murala *et al.*, 2009), the color and texture features are combined to retrieve similar images. For the color, the mean and standard deviation are computed in a histogram of 64 bins in each channel of the RGB color image, to get a total of 192 features. For the texture features, the mean and standard deviation are computed in sub bands of the Gabor Wavelet Transform image with the three scales and four orientations to get a feature vector of 48 features. In (Thawari and Janwe, 2011) the HSV color space is used with three color channels, H, S and V. The histogram of each channel is quantized into 96 blocks, and each block has a dimension of 32×32 pixels. The statistical texture moments of mean, standard deviation, skew, kurtosis, energy, entropy and smoothness are calculated in each bin of the histogram. The total 96×7×3=2016 features are computed. Thus, this process of feature extraction involves a large number of computations which increase computational cost. The method has used 500 images of the Corel database for testing. In the approach of (Kavitha *et al.*, 2011), the color and texture features are also combined. The HSV color space image is divided into sub-blocks. The color features are calculated by quantizing the histograms of each block. The texture features are calculated by using the grey level co-occurrence matrix. In the approach of (Soman *et al.*, 2011), the color and texture features are extracted by computing the color moments of mean, standard deviation and skewness in 8×8 blocks of the three components of the RGB image and computing the DC and AC coefficients in 9 directions in 8×8 DCT blocks. The CBIR approach proposed by (Singha and Hemachandran, 2012) is based on the combination of the texture and color features in which color features are extracted by using the histogram while texture features are extracted by using the Haar Wavelet Transformation to get vertical, horizontal and diagonal coefficients.

2.4.4 Summary and Limitations in the Related Works

Limitations in the related works, for the extraction of color and texture features in the spatial and frequency domains are summarized in Table 2.1. It has been summarized in the table that approaches with S#:1 to 3 using histogram for the extraction of color features, approaches with S#:4 to 9 using the coefficients of DCT transformed blocks in different techniques of histogram, co-occurrence matrix and sub-blocks, for the extraction of texture features and approaches with S#: 10 to 20 using different techniques to extract and combine color and texture features. It has been concluded from the approaches in the Table that the CBIR has two major limitations in terms of computational cost of feature extraction and retrieval efficiency. Therefore, these limitations motivated us to develop efficient and effective CBIR using hybrid methods with color and texture features.

Table 2.1 Summary of the related works for the extraction of color and texture features in the spatial and frequency domains.

S#	Author	Features	Techniques/Work	Limitations
1	(Chakravarti and Meng, 2009)	Color	Color Histogram of RGB color components	The algorithm is less efficient in retrieval and also attempts to analyze color information by studying each RGB component histogram separately. Thus, the color analysis does not provide necessarily similarity of colors.
2	(Sergyan, 2008)	Color	Color histogram features	Histogram features are generated from the image histogram fast and the comparison of these features is computationally fast and efficient using only 200 images.
3	(Zhao et al., 2009)	Color	Color histogram	The approach is less efficient in terms of computational cost, however the retrieval efficiency is only 28% precision for some query images
4	(Selvarajah and Kodituwakku, 2011)	Texture	First Order S autistics, Autocorrelation (AC), Gray Level Run Length Matrices (GLRLM), Gray Level Co-occurrence Matrix(GLCM), Gabor Transform and 2 D wavelet Transform	Minimum efficiency of individual feature is 34% while combination of all features is 78%. However features combination indicates somewhat computational complexity but has robust retrieval.
5	(Shan and Liu, 2009)	Texture	the DC vector and AC distribution entropy are computed by using the distribution of AC and DC coefficients	Approach has high retrieval efficiency. However indicates somewhat computational issue.
6	(Tsai *et al.*, 2006)	Texture	In the YUV color space, the texture features are extracted by dividing into four blocks and only the Y component in each block is transformed in the DCT coefficients to get vertical, horizontal and diagonal AC coefficients as features in all of the blocks	Computationally less expensive, however, the performance is often poorer as the features from different sub-blocks may be correlated.
7	(Mohamed *et al.*, 2009)	Texture	Quantized histogram texture features are extracted from DCT coefficients over the all	Computationally less expensive and has good retrieval efficiency for a small

			Image blocks by using the DC and the first three AC's coefficients	dataset only. Results are poor for the images which have the similar complex background to the object itself.
8	(Lay and Guan, 1999).	Texture	The DC and some of the AC coefficients are used directionally to get energy histograms which are represented as feature vectors	Computationally less expensive but retrieval efficiency is comparatively less.
9	(Deselaers et al., 2007).	discussed a large variety of features for image retrieval	Different techniques used	It has been shown that, despite more than 30 years in research on texture descriptors, still none of the texture features presented can convey a complete description of the texture properties of an image. Therefore a combination of different texture features will usually lead to best results.
10	(Hiremath and Pujari, 2007)	Color, texture and shape	Texture, color and shape features are fused together and extracted in a non-overlapping partitioned image by using the Gabor filter, the statistical color moments and the Gradient vector flow fields.	Approach's retrieval efficiency is 55% precision while computational cost indicates comparatively less expensive.
11	(Soman et al., 2011)	Color and texture	In the approach of, the color and texture features are extracted by computing the color moments of mean, standard deviation and skewness in 8×8 blocks of the three components of the RGB image and computing DC and AC coefficients in 9 directions in DCT blocks.	Approach's retrieval efficiency is 57% precision while there is some computational issue due to the extraction in the three components of RGB image and then in the DCT blocks.
12	(Singha and Hemachandran, 2012)	Color and texture	combination of the texture and color features in which color features are extracted by using histogram while texture features are extracted by using the Haar Wavelet Transformation to get vertical, horizontal and diagonal coefficients.	There is some computational issue due to the extraction in the three components of HSV and then Haar Wavelet Transformation coefficients while retrieval efficiency is better with 76% precision.
13	(Lu et al., 2006)	color and texture	Statistical Color and texture, based on DCT coefficients, are computed.	Retrieval efficiency is comparatively is good but has less computationally cost.

14	(Alnihoud, 2012)	Color and shape	Color and shape features are extracted based on the SOM (self-organizing map). A Fuzzy Color Histogram (FCH) and the object Model Algorithm (to get the edge of the objects).	Good retrieval efficiency of 74% precision while algorithm indicates some computational complexity.
15	(Shahbahrami et al., 2008)	Color and Textures	Color histogram, color moments, and color coherence vector features. The co-occurrence matrices and discrete wavelet transform features have been used for texture descriptors	Using several techniques for feature extraction it shows that there is some computational issue.
16	(Murala et al., 2009)	Color and Textures	Color Histogram and Gabor Transformation	Retrieval efficiency is 64.76% precision while approach indicates that there is some computational issue due to high dimensional feature vector.
17	(Thawari and Janwe, 2011).	Color and Textures	Color and statistical texture features of Color Histogram of RGB color components	Computations and feature vector with high dimension indicates somewhat computational issue while algorithm is tested by using a small dataset with very few query images.
18	(Park et al., 2008).	Set of features with color, and shape of objects.	Histogram Refinement Method	This approach has high retrieval performance but commutation of several features indicates some computational issue for the creation of feature database.
19	(Kavitha *et al.*, 2011)	Color and texture	Sub-block method, color histogram and gray-level co-occurrence matrix from each block	Retrieval efficiency is 50% precision while approach indicates that there is some computational issue due to using various techniques for extraction of features.
20	(Dubey *et al.*, 2010)	Color and texture	combining the color Histogram, Color Moment, co-occurrence matrix, and edge Histogram descriptor features	Retrieval efficiency is 27% while approach indicates that there is some computational issue due to using various techniques for extraction of features.

2.5 Similarity Measurement

Similarity measuring is rather easier said than done. It is relatively complex to measure between the two images. It is a challenging task in CBIR. The similarity measurement shows the power of the correlation between the features of two images. It distinguishes one feature from another between the two images. It utilizes the extracted features of the images in such a way that a single value is generated which shows the desired subjective similarity between the two images and the similarity is high if the value is higher (Arevalillo-Herráeza et al., 2008).

After features extraction, the there was a question that how these features would be indexed efficiently and matched effectively for retrieval. Then different methods for similarity measurement were introduced by the researchers and grouped like feature-based matching (Swain and Ballard, 1991), salient (geometric hashing) feature matching (Wolfson and Rigoutsos, 1997), object-silhouette-based matching (Bimbo and Pala, 1997), structural (hierarchically ordered sets of features) feature matching (Wilson and Hancock, 1997), matching at the semantic level (Fagin, 1997) and learning-based approaches for matching (Webe et al., 2000; Wu et al., 2000).

To imitate the needs of the users the similarity measurement is improved step wise by the feedback of the users and an approach was introduced called relevance feedback (RF) which was used by (Rui et al., 1997) in MARS system.

To measure the similarity between the query and target images a method was proposed using the earth mover's distance (EMD) (Rubner et al., 2000) to compute the distance between images using their feature vectors. Another distance metric used by (Li et al., 2000) is the IRM (integrated region matching) distance which is also useful matching-based distance and this distance metric uses the most similar highest priority (MSHP) principle for the matching of regions.

Once the features are extracted from the image then the features are represented in a feature vector with N-dimensions. The feature vectors are used to index the images into a database. To measure the similarity between the query images and the target images of the database, the distance is computed between the feature vectors of the query and the target images. The main problem in CBIR is to determine the distance between the images. The distance value reveals the desired perception about the images of humans. The similar images have small distance values and dissimilar images have large distance values. There are various distance metrics which can be used to measure the distance between the features vectors of images, for example the Sum of the Absolute Difference (SAD) (Kekre and Mishra, 2011; Selvarajah and Kodituwakku, 2011), the Sum of the Squared of the Absolute Difference (SSAD) (Selvarajah and Kodituwakku, 2011), the Euclidean distance, the City block distance, the Canberra distance, the Maximum values and Minkowski distance (Vadivel et al., 2003; Zhang and Lu, 2003; Cha, 2007; Sergyan, 2008).

Let Q and T be the feature vectors of the query and target images having n number of features such that $Q= \{q_1 , q_2, \ldots, q_n\}$ and $T=\{t_1 , t_2, \ldots, t_n\}$ where q_i and t_i are the features of the query and target images. The Euclidean distance D can be used to measure the distance between Q and T feature vectors.

$$D(Q,T) = \sqrt{\sum_{i=1}^{n}\left(\left|Q_i - T_i\right|\right)^2} \tag{2.1}$$

The above mentioned distance metrics are discussed in chapter 4.

2.6 Performance Evaluation

The effectiveness of the image retrieval is based on the performance of the feature extraction and the similarity measurement. In this section, the performance metrics which have been adopted not only to evaluate the effectiveness of the image retrieval but also to make sure of the stability of the results are described. In order to evaluate the retrieval performance of CBIR, three measurements have been used: precision, recall (Thawari and Janwe, 2011) and F- score (Sudhakar et al., 2011; Jacob and Srinivasagan, 2013).

Before defining precision and recall first it is necessary to discuss and explain abstract diagram of these measurements. Let ***A*** be the set of images in the image database, ***q*** be the user query image to retrieve the similar images from ***A*,** and ***B*** be the set of retrieved images according to the query image ***q*** as show in Fig. 2.1.

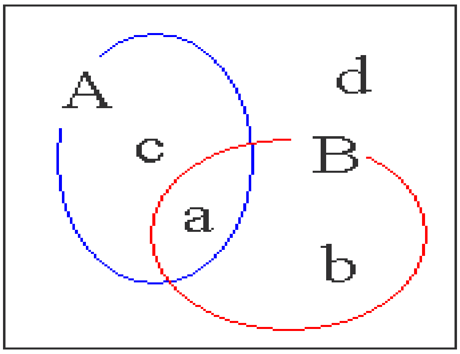


Figure 2.1 Abstract diagram of the performance evaluation measurements for CBIR.

There are three other sets ***a***, ***b*** and ***c*** as shown in Fig. 2.1. Where ***a*** denotes the retrieved most relevant images to the query image ***q***, ***b*** denotes the irrelevant images to the query image and ***c*** denotes the relevant images in set ***A*** and ***d*** denotes the irrelevant images which are not retrieved. Consider the above diagram and the proposed sets of images then the precision and recall can be defined as follows:

- **Precision**

The precision in image retrieval can be defined as: the measurement of the retrieved relevant images ***a*** to the query total retrieved images[3] ***B***.

$$Precision = \frac{a}{a+b} = \frac{a}{B} \tag{2.2}$$

- **Recall**

The recall in image retrieval can be defined as: the measurement of the retrieved relevant images ***a*** to the total relevant images in database[4] ***A***.

$$Recall = \frac{a}{a+c} = \frac{a}{A} \tag{2.3}$$

- **F-Score[5]**

The precision and recall measure the accuracy of the image retrieval with relevancy to the query and database images and always two values are computed to show the effectiveness of the image retrieval. However, these two measurements cannot give

---

[3] http://en.wikipedia.org/wiki/Precision_and_recall
[4] http://en.wikipedia.org/wiki/Precision_and_recall
[5] http://aimotion.blogspot.com/2011/05/evaluating-recommender-systems.html

complete view of the effective image retrieval. Hence, they can be combined to give a single value that describes the accuracy of image retrieval and this combination is called F-Score or F-measure to measure accuracy. Both precision and recall measurements are combined to compute the F-score (Sudhakar et al., 2011; Jacob and Srinivasagan, 2013) and it is also called as a weighted average or the harmonic mean of the precision and recall. F-Score can be defined as:

$$\mathrm{F-Score} = 2 \times \frac{\mathrm{precision} \times \mathrm{recall}}{\mathrm{precision} + \mathrm{recall}} \tag{2.4}$$

The F-Score value is a single value that indicates an overall effectiveness of the image retrieval.

For example, a CBIR method for a query image retrieves totally ***B*=**10 images with ***a*=**8 relevant images out of totally ***A*=**30 relevant images in the database. Then the precision is ***a* ÷ *B*** = 8 ÷10= 80%, recall is ***a* ÷ *A*** =8 ÷ 30 =27% and F-Score is 2× (80×27) ÷ (80+27) =40.37%. Thus this shows that only recall and precision cannot measure the effectiveness of the CBIR, a single valued F-Score must also be computed to show the overall performance in terms of retrieval.

2.7 Benchmark Image Datasets

In the literature, the benchmark image data sets are proposed in order to compare the performance of the different image features for the retrieval systems (Deselaers et al., 2008). In this research work, the Corel dataset (Wang et al., 2001) is used to test the proposed approaches and the comparison of the optimal proposed approach with the other approaches in literature is also based on same the dataset.

- **Corel Dataset**

The Corel (Wang et al., 2001) dataset consists of 1000 images from the Corel stock photo database. The 1000 images are divided into 10 categories: People, Beach, Building, Buses, Dinosaurs, Elephants, Horses, Roses, Mountains, and Foods. Each category consists of 100 images. Figure 2.2 shows an example of the images from each image category. In the retrieval of this work’s experiments, an image of the same category as the query image will have a positive and images of different categories as negative. Hence, the Corel dataset is used for evaluating features and methods for image matching.

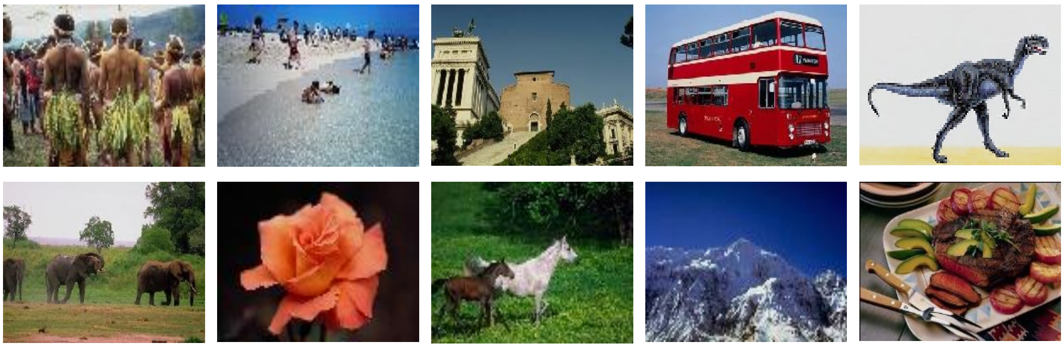


Figure 2.2 Images of each category from the Corel dataset.

2.8 Chapter Summary

In this chapter, some of the essentials required for this research have been presented. The chapter started with the concept of content-based image retrieval, types of features, and various basic features like color, texture and shape. It also presented techniques of extraction. The domains are discussed where the features are extracted, which affect the retrieval of the images. The extracted features are represented in vector form. The similarity measurement discussed using the feature vectors for the matching of images. The performance measurements, which are used to evaluate the performance of this research work, are discussed. The benchmark data set is described which is used to test our approaches and the comparison in Chapter 5 with the other methods is based on the same dataset.

CHAPTER 3

METHODOLOGY

In order to improve the efficiency of CBIR, in this study, various approaches are proposed in which different techniques are fused together in the spatial and frequency domains to design and implement CBIR system using hybrid methods with color and texture features as shown in Fig. 3.1

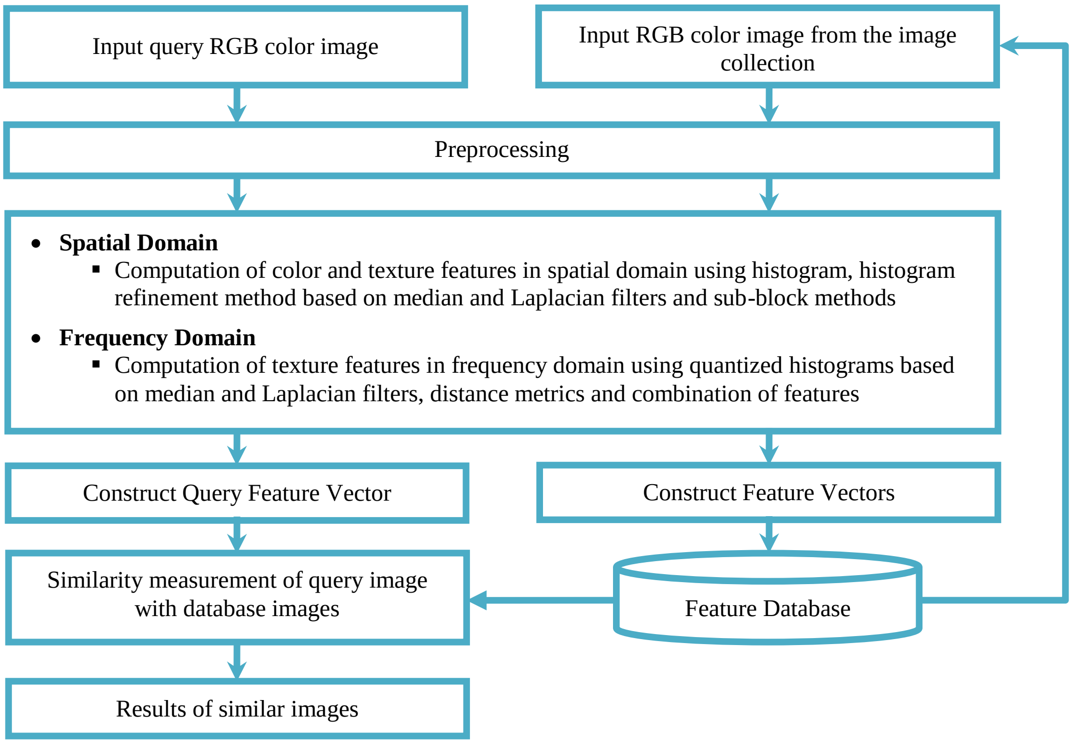


Figure 3.1 Overview of the entire CBIR using hybrid methods with color and textures features in the proposed approaches in the spatial and frequency domains.

In the spatial domain the first approach is based on Laplacian filter using color histogram; the second one is based on median filter, median filter with edge extraction method and Laplacian filter using color histogram refinement method, third one is based on sub-blocks of different sizes for color features; the fourth one is based on sub-blocks of different sizes for texture features; and the fifth one is based on the integration of color and texture features using sub-block methods.

In the frequency domain to extract the quantized histogram texture features from DCT blocks the first approach is based on median filter, median filter with edge extraction method and Laplacian filter; the second one is based on various distance metrics (for similarity measurement); the third one is based on different combinations of texture features; the fourth one is based on the combination of sub-block method with histograms of DCT blocks for texture features; and the fifth one is based on the combination of sub-block method with histograms of DCT blocks for color and texture features.

3.1 Color and Statistical Texture Features Extraction in the Spatial Domain

To extract the color and texture features from an image for the efficient retrieval of similar images different techniques are combined in various approaches in a spatial domain. In the spatial domain various approaches are proposed in which the statistical color histogram features are computed using the pixel distribution of the Laplacian filtered sharpened image based on the different quantization schemes. However, color histogram does not provide the spatial information. The solution is by using the histogram refinement method in which the statistical features of the regions in histogram bins of the filtered image are extracted. This approach gives efficient retrieval but it has high computational cost, which is reduced by dividing the image into sub-blocks of different sizes, to extract the local color and texture features of image and also to get local information instead of using complex segmentation. To improve further the performance, the color and texture features are combined using sub-block methods due to the less computational cost and good local information.

To improve the retrieval performance of the CBIR, color and texture features are combined in various approaches in the spatial domain. For this purpose the following contributions are mainly focused towards the efficient and effective CBIR:

- Extraction of the color features of the histograms and their quantization into a set of different numbers of bins based on Laplacian, median and median with extraction techniques to achieve the effective image retrieval. The effectiveness is analyzed based on the set of different quantization schemes of histogram in the context of the filters.

- Utilization of the sub-block method to extract the color and statistical texture features instead of the histogram to reduce the computational complexity. The performance is analyzed individually for both features on the basis of the sub-block methods of different sizes.

- Combination of color and texture features using the sub-block methods to get robust performance in terms of retrieval and reduced computational cost.

3.1.1 Analysis of the Quantized Color Histogram Features Based on the Laplacian Filter (Approach-1)

In this proposed Approach-1, the main contribution is to analyze and show the performance of the quantized histogram color features based on the Laplacian filter and different quantization schemes. The proposed approach is started with the conversion of the RGB color image into grayscale image and then gets a sharpened image by using the Laplacian filter. The first and second order color moments are extracted from the histograms by quantizing them into a set of different quantization schemes and these features are represented in a feature vector. The feature vectors of all of the images are constructed and stored in a database. These vectors are used in the matching of the query image with the database target images to retrieve similar images.

3.1.1.1 Preprocessing

The preprocessing involves the conversion of an image from RGB color space to grayscale image and then it is further enhanced by using Laplacian filters.

3.1.1.1.1 Conversion of RGB Color Image into Grayscale Image

As the RGB color image consists of three color components, red, green and blue, and each component is a two dimensional matrix of pixel values having values from 0 to 256, therefore the computational cost of the feature extraction would be high. To reduce the computational cost, the RGB color image is converted into grayscale

(Murala et al., 2009). The proposed approach starts with the conversion of the input RGB image into grayscale as shown in Fig. 3.2-A and 3.2-B-a. The grayscale image is then converted into a histogram equalized (HE) image *f*, to get the enhanced image with equal intensity levels so as to get a high contrast image as shown in Fig. 3.2-B-b.

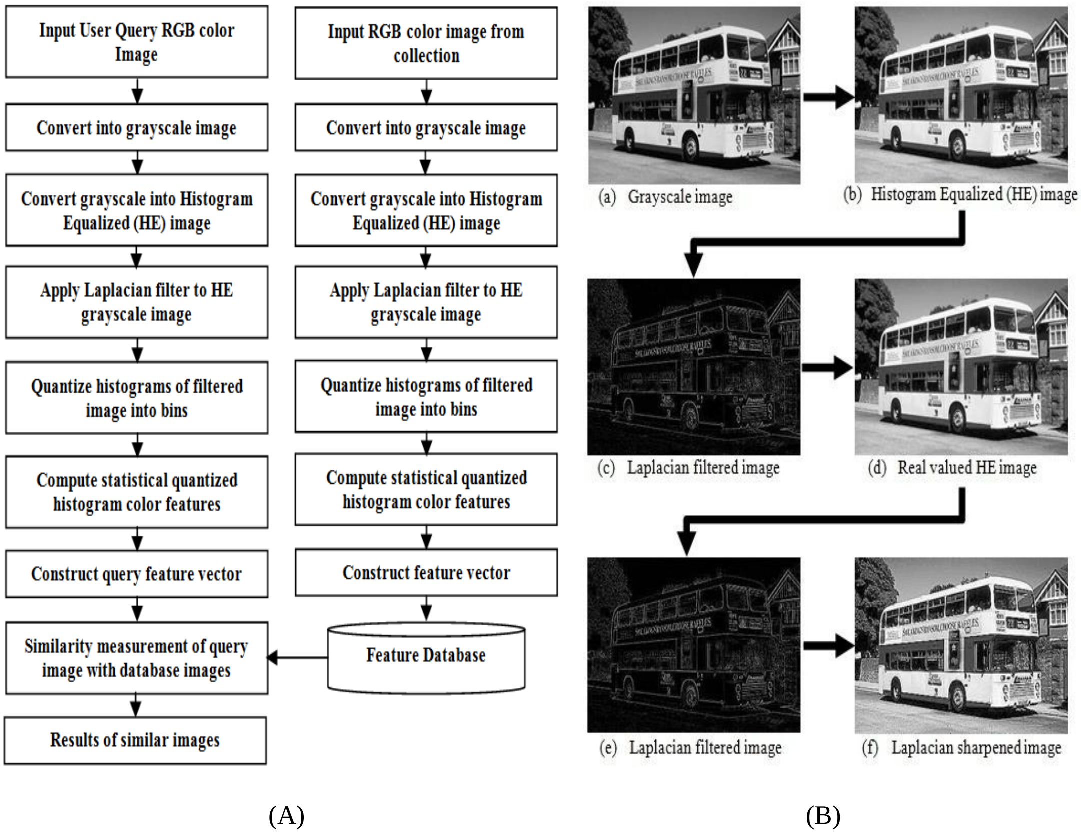


Figure 3.2 (A) Block diagram of the proposed Approach-1(B) Sharpening process using Laplacian filter.

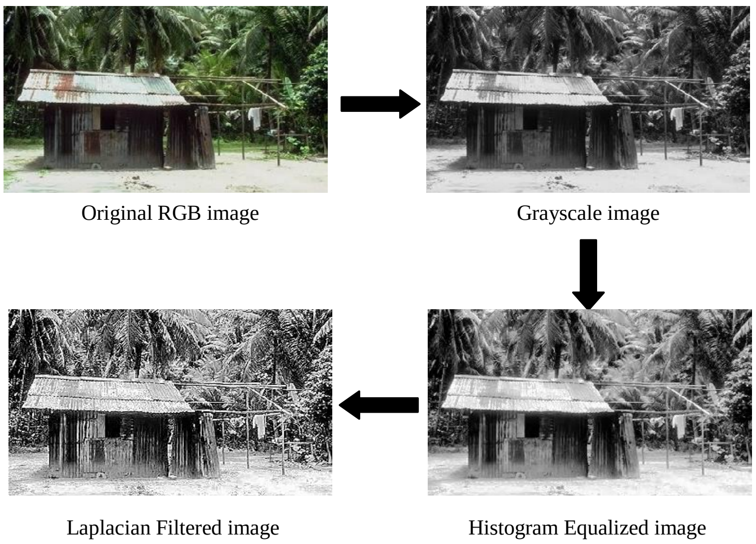


Figure 3.3 Preprocessing of the proposed Approach-1 using Laplacian filter.

Then Laplacian filter is applied to get a more enhanced sharpened image. The preprocessing steps using the Laplacian filter are shown in Fig. 3.3.

3.1.1.1.2 Laplacian Filter

Images mostly consist of noise and unwanted information which effects the matching of images in CBIR. Therefore it is very essential to remove the noise and unwanted information in image by performing effective noise reduction process before performing high-level processing steps to obtain multi-resolution images by using different filters like median, Laplacian and Gaussian filters(Demigny, 2002).

In the proposed approach we use Laplacian filter which is mainly used for the extraction of the strong edges of the objects in the images. In our work the Laplacian filter uses a window of values which are convoluted with image pixel values. These values are further processed to get enhanced and sharpened image to improve the retrieval of images because oftenly images with noise and unwanted information lead to inefficient retrieval of images in CBIR.

In the Laplacian filter, a window or mask with some values works with the values of the image pixels in the neighborhood. The values in the filter window are called filter coefficients. The result of this filter is the sum of the products of the filter coefficients and the corresponding image pixel values. This filter gives an image with strong edges (Gonzalez et al., 2004).

Let $f(x, y)$ be an original image and $\nabla^2 f(x, y)$ be a Laplacian image such that

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \tag{3.1}$$

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y) \tag{3.2}$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y) \tag{3.3}$$

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] - 4f(x, y) \tag{3.4}$$

To get the filtered image, all points (x, y) in Eq. 3.4 can be convoluted with a 3×3 mask as seen in Fig. 3.4.

0	1	0
1	-4	1
0	1	0

Figure 3.4 3×3 mask of the Laplacian filter.

The image *f*, is filtered with the Laplacian filter to get a filtered image g_1, with the edges of objects in the image as show in Fig. 3.2-B-c. But all of the pixel values in g_1 are positive and these values must be negative because of the negative value -4, at the center of the mask as shown in Fig. 3.4. For this purpose, the histogram equalized image *f*, is converted into the real valued image f_2, as shown in Fig. 3.2-B-d. This image f_2, is again filtered with the Laplacian filter to get image g_2 with the edge information as shown in Fig. 3.2-B-e. But during the filtering process of image g_2, some amount of the information is lost. To restore this information and get an enhanced and sharpened image *g*, the Laplacian filtered image g_2, is subtracted from the real valued image f_2 and calculated as (Gonzalez et al., 2004):

$$g = f_2 - g_2 \tag{3.5}$$

In Eq. 3.5, *g* is the sharpened and enhanced image with detailed information as shown in Fig. 3.2-B-f. This process is also called sharpening of the image (Gonzalez et al., 2004). The features are extracted from image *g* for retrieval and analysis.

3.1.1.2 Histogram Quantization

Color is the most prominent and important feature of an image because it is the dominant part of the human visual perception. It is widely used in CBIR to retrieve images. For this purpose, various color techniques have been used. Among these techniques, the color histogram is the most popular and widely used technique.

The histogram is defined as the frequencies of the color pixels in the sharpened grayscale image. Quantization is a process in which the histogram is divided into a number of bins to reduce the number of bins by selecting color pixels which have very close similarity with each other and enclosing them in the same bin. The grayscale image by default has 256 maximum number of bins (Park et al., 2008).

Histogram has the characteristic that it represents the global information of the image (Swain and Ballard, 1991; Jin, 2009). This global information representation of the image is very useful in the queries in which the matching of the images is based on the whole appearance. The color histograms are very fast in the computation of the features (Park *et al*., 2008). It has no effect on the small changes in the scenes. It is useful and widely used for the images which require invariance in the translation and rotation (Park *et al*., 2010). However, for a query which requires retrieving the images with the same scenes but with different appearances of illuminations, the color histogram is not a suitable technique. Spatial information in the color histogram is not maintained due to which the same histograms will be extracted for the images with diverse appearances. In other words, an image with many very small green spots has a histogram similar with the histogram of the image which has a single large green area (Jin, 2009).

To preserve the spatial information in a histogram, an algorithm is proposed by (Liu *et al*., 2007) in which the color histogram refinement method is used so that pixels of the same color are classified into coherent and incoherent clusters. This method is also called the color coherence vector (CCV).

The value of each pixel indicates a specific color that can be represented in the various color spaces of the three components, red, green and blue (RGB), and Hue, Saturation and Value (HSV). Each component, R, G and B, in the RGB color space consists of 0 to 255 pixel values or intensity levels. The computational cost will be high for the extraction of the color histogram features from all the three components. To reduce the computational speed, the RGB color image is converted into a single component grayscale image of only 0 to 255 levels.

In the proposed Approach-1, all of the 256 levels of the grayscale image are not used and it is further reduced to certain levels to reduce the computational cost. The process to divide the image into levels or bins of frequencies of the same color is called quantization. In this approach, the filtered grayscale image of 256 levels is quantized into different numbers of bins so as to reduce the computations.

Hence in this proposed approach the histogram of the sharpened filtered image is quantized into different quantization schemes. The quantization process is carried out separately for quantization schemes of 4, 8, 16, 32, 64 and 128 bins.

The histogram is then quantized into L (4, 8, 16, 32, 64 and 128) bins such that:

$$H = \{h(b_1), h(b_2) \ldots h(b_L)\} \tag{3.6}$$

Where $h(b_i)$ is the frequency of the pixel values in bin b_i, for $i=1,2,..L$ and H is the histogram of L bins.

3.1.1.3 Feature Extraction

For feature extraction, the statistical color moments are considered useful for retrieval of similar images. These color moments provide the information about the intensity level distribution in the image. The first-order moment is the mean and the second order moment is the standard deviation. The mean represents the brightness of an image and the standard deviation represents the contrast. The dark image has a low mean and the bright image has a high mean. The low contrast image has a low standard deviation while the high contrast image has a high standard deviation. The mean and standard deviation are calculated in each bin using the distribution of pixels.

The statistical color features' mean and standard deviation are calculated in the histogram bins of $\boldsymbol{H}$. Let $\boldsymbol{\mu_j}$ be the mean and $\boldsymbol{\sigma_j}$ be the standard deviation in a particular bin $\boldsymbol{j}$, where $j=1, 2, 3\ldots, L$, and then these two features are calculated by using the statistical measurements (Wang *et al.*, 2001; Jia and Wang, 2003) as:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ji} \tag{3.7}$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ji} - \mu_j)^2} \tag{3.8}$$

Where x_{ji} is the pixel value of pixel i in bin j and N is the total number of pixels in each bin.

After the calculation of these color features, the feature vector FV of these values is constructed as:

$$FV = \{\mu_1, \mu_2 \ldots \mu_L, \sigma_1, \sigma_2 \ldots \sigma_L\} \quad (3.9)$$

The feature vectors (*FVs*) of all of the images are constructed and are stored in a database. Algorithm 1 is used to extract color features. The feature vector of the user query is constructed in the same way using Eq. 3.7 to 3.9 and compared with the feature vectors of the database for similarity and retrieval of relevant images as shown in Fig. 3.2-A.

Algorithm 1 Feature Extraction Algorithm of Approach-1

Input: Input image ***Img_file***, Number of bins ***B***

Output: Feature vector ***fv***

1. Read and convert the ***Img_file*** into 2D matrix
   - 1.1 *Img* = ***imgread***(*Img_file*)
2. Convert RGB ***Img*** into grayscale image
   - 2.1 *Img_gray* = ***RGB_to_Gray*** (*Img)*
3. Convert grayscale *Img_gray* into Histogram Equalized image
   - 3.1 *Img_he* = ***hist_Eq*** (*Img_gray)*
4. Apply Laplacian filter to *Img_he*
   - 4.1. *S_img* = ***Laplacian_Filter*** (*Img_he*)
5. Construct histograms and quantize into ***B*** bins
   - 5.1. For b = 1 to ***B***
   - 5.2. *h(b)* = ***Histogram***(*S_img*, b)
   - 5.3 End for
6. Compute the features of mean and standard deviation
   - 6.1. For b = 1 to ***B***
   - 6.2. *m(b)* = ***mean***(*h(b))*
   - 6.3. *stddev(b)* = ***std***(*h(b))*
   - 6.4 End for
7. Initialize *K=1*
8. Construct feature vector ***fv*** of features
   - 8.1. For b = 1 to ***B***
   - 8.2. *fv(1, K) = m(b)*
   - 8.3. *fv(1, K+1) = stddev(b)*
   - 8.4. *K = K + 2*

8.5 End for

9. Return feature vector *fv*

    9.1 *Return fv*

3.1.2 Color Feature Analysis for CBIR Based on Median, Median with Edge Extraction Method and Laplacian Filters using the Color Histogram Refinement Method (Approach-2)

In this section, a CBIR Approach-2 is proposed which is based on median and Laplacian filters to reduce the noise and provide enhanced sharpened images with more detail information using the color histogram refinement method to overcome the problem of retaining of spatial information which is not provided by standard histogram. The color histogram is divided into bins. The number of regions is determined in each bin. The statistical color moments of mean and standard deviation are calculated in each bin by using the areas of the regions to get a feature vector which is used for image retrieval.

The proposed Approach-2 is based on the analysis of the statistical color histogram features using median, median with edge extraction method and Laplacian filters. The statistical features are extracted in histograms using the spatial information for each filter separately. Before applying the histogram to the image for the extraction of the features, preprocessing is performed. Step wise and entire block diagrams of the proposed Approach-2 are shown in Fig. 3.5 and Fig. 3.6.

3.1.2.1 Preprocessing

Preprocessing consists of conversion from RGB to grayscale image and the filters method to enhance image before using for features extraction.

3.1.2.1.1 Conversion of RGB Color Image into Grayscale Image

The RGB color image is converted into grayscale and then the grayscale image is then converted into a histogram equalized (HE) image to get the enhanced image as discussed in section 3.1.1.1.1 and shown in Fig. 3.5.

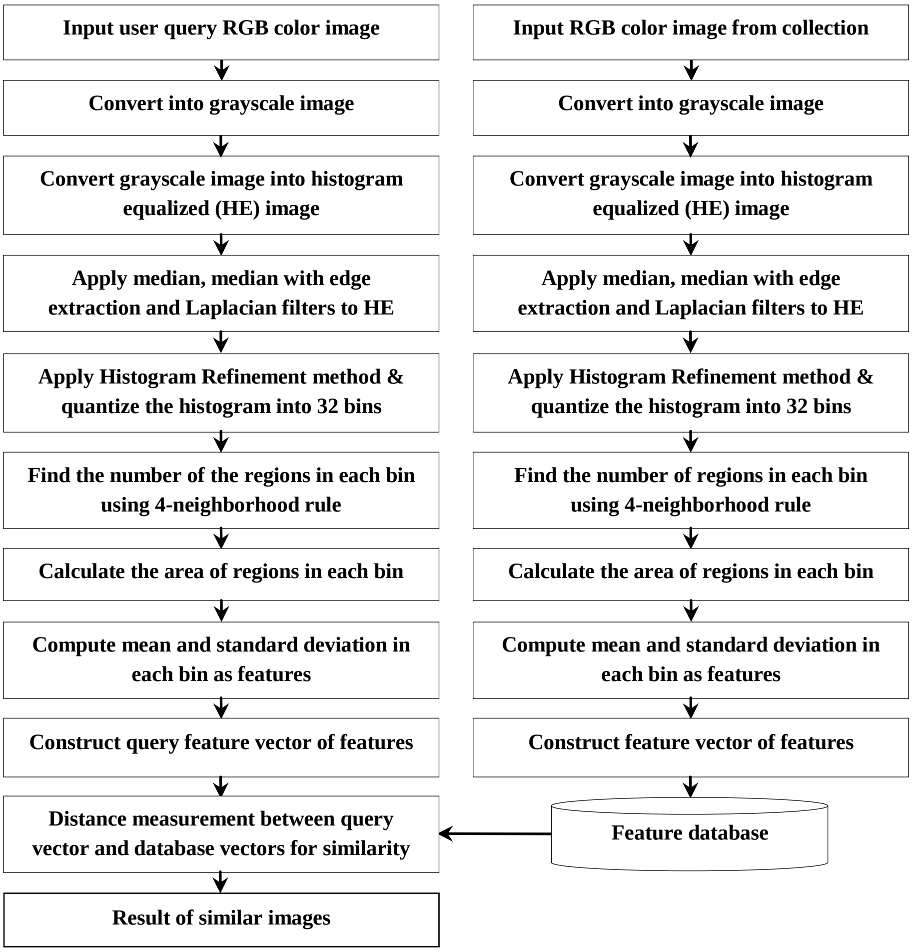


Figure 3.5 Step wise block diagram of the proposed Approach-2.

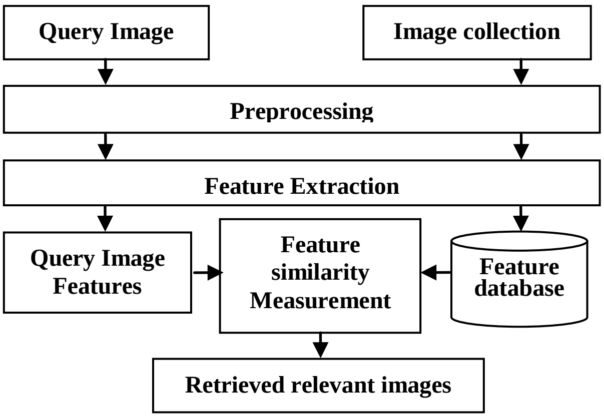


Figure 3.6 Block diagram of the entire proposed Approach-2's process.

In the proposed approach we use median and Laplacian filters. In our work these filters use a window of values which are convoluted with the image pixel values. These values are further processed to get enhanced and sharpened image to improve the retrieval of images because oftenly images with noise and unwanted information lead to inefficient retrieval of images in CBIR.

3.1.2.1.2 Median Filter

Images can contain noise. Therefore, when applying processing techniques on images to extract color features, the images needed to be preprocessed to remove unwanted information and to get enhanced images with the relevant information only. Median filter is based on neighborhood operations. It consists of a window which is encompassed over an image to put into order (rank) the pixels in the image area and then replace the central pixel with the determined values.

The median filter replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel (Gonzalez *et al.*, 2004). This filtered image is then used for the feature extraction.

3.1.2.1.3 Median with Edge Extraction Method

Even though median filtering removes the noise from images, some black specks are still left around the border. These black points are due to the default padding of zeros (0's). Some amount of information in the image, like edge information, is lost (Gonzalez et al., 2004). To restore the edge information of the median filtered image, a technique called canny edge detection is used to determine the edge information in the image before applying the median filter (Shan and Liu, 2009). This is the most powerful edge detector. This technique detects two edge points, strong and weak, using two threshold values T1 and T2 such that $T1<T2$. If the pixel values are greater than T2, then the edge values are strong and if the pixel values are in between T1 and T2, then these are called weak edge pixels. At the end, the canny technique connects the weak edges to the strong edges by using an 8-connection. The edge detection technique is used to determine the edges before applying the median filter to the

image. The edge information of the median filtered image is restored by the already extracted edge information (Gonzalez et al., 2004). Thus, the features are extracted from the median filtered image with the edge extraction method.

3.1.2.1.4 Laplacian Filter

The Laplacian filter is discussed in detail in section 3.1.1.1.2.

3.1.2.2 Histogram Refinement Method

Histogram quantization has been discussed in section 3.1.1.2 and the main disadvantage of the color histogram is that spatial information is not maintained due to which the same histograms will be extracted for the images with diverse appearances. In other words, an image with many very small green spots has a histogram similar with the histogram of the image which has a single large green area (Jin, 2009).

To preserve the spatial information in a histogram, an algorithm is proposed by (Liu et al., 2007) in which the color histogram refinement method is used so that pixels of the same color are classified into coherent and incoherent clusters to get the connected regions. The pixels in the regions have spatial correlation in their neighborhoods. Different properties of the regions can be extracted as features. This method is also called the color coherence vector (CCV).

The histogram is then quantized using different quantization schemes with L bins such that:

$$H= \{h(b_1), h(b_2) \ldots .h\ (b_L) \tag{3.10}$$

where $h(b_i)$ is the frequency of the pixel values in bin $\boldsymbol{b_i}$ and $\boldsymbol{H}$ is the histogram of the $\boldsymbol{L}$ bins.

After quantization of enhanced filtered image into number of bins then in the bins the number of connected regions is determined using 4-nieghborhood. These regions can be used to extract the features.

3.1.2.3 Feature Extraction

For feature extraction, the color histogram refinement method (CHRM) is used. The color histogram is quantized into different number of bins and each bin is divided into connected regions of pixels. The number of regions in each bin is determined. Then, the area of each region is calculated. Two color moments are used to calculate the features. The first-order moment is the mean and the second order moment is the standard deviation. The mean and standard deviation are calculated in each bin using the areas of the regions.

The statistical color features' mean and standard deviation are calculated in the histogram bins of $\boldsymbol{H}$. Let $\boldsymbol{\mu_j}$ be the mean and $\boldsymbol{\sigma_j}$ be the standard deviation in a particular bin $\boldsymbol{j}$, where $j=1, 2, 3..., L$ bins, and then these two features are calculated by using the statistical measurements (Wang *et al.*, 2001; Jia and Wang, 2003) as:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} A_{ji} \tag{3.11}$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(A_{ji} - \mu_j)^2} \tag{3.12}$$

Where $\boldsymbol{A_{ji}}$ is the area of the ***i**th* region in ***j**th* bin and $\boldsymbol{N}$ is the total number of regions in each bin $\boldsymbol{j}$.

After the calculation of the color features, the feature vector *FV* is constructed as:

$$FV= \{\mu_1,\ \mu_2 .... \ \mu_L,\ \ \sigma_1, \sigma_2 .... \ \sigma_L\} \tag{3.13}$$

For all of the images in the database, the feature vectors (*FVs*) are computed and stored in the database, using Algorithm 2. The feature vector of the user's query image is constructed in the same way and compared with the feature vectors of the database for similarity and retrieval of relevant images as show in Fig. 3.5. Algorithm 2 is used to extract the color histogram features based on the filters.

Algorithm 2 Feature Extraction Algorithm of Approach-2

Input: Input image ***Img_file***, Number of bins ***B***

Output: Feature vector ***fv***

1. Read and convert the ***Img_file*** into 2D matrix
   - 1.1 *Img* = ***imgread***(*Img_file*)
2. Convert RGB ***Img*** into grayscale image
   - 2.1 *Img_gray* = ***RGB_to_Gray*** (*Img)*
3. Convert grayscale *Img_gray* into Histogram Equalized image
   - 3.1 *Img_he* = ***hist_Eq*** (*Img_gray)*
4. Apply Median and Laplacian filters to *Img_he*
   - 4.1. *F_img* =***Filters*** (*Img_he*)
5. Construct histograms and quantize into ***B*** bins and create regions ***N*** of objects
   - 5.1 For b= 1 to ***B***
   - 5.2 *Q_ hist(b)* =***Histogram***(*F_img*, b)
   - 5.3 *Region(b)* =***Create_Regions*** (*F_img*, *N*)
   - 5.4 End for
6. Initialize *K=1*
7. Compute the features and construct feature vector ***fv***
   - 7.1 For b= 1 to ***B***
   - 7.2 *fv(K)* = ***mean****(****Area****(Region(b)))*
   - 7.3. *fv(K+1)* = ***std***(***Area****(Region(b)))*
   - 7.4 *K = K + 2*
   - 7.5 End for
8. Return feature vector *fv*
   - 8.1 *Return fv*

3.1.3 Features Analysis for CBIR Based on the Color Moments using the Block Methods (Approach-3)

We propose an approach for CBIR that is based on statistical color moments using the sub-block methods of different sizes. Efficient and effective retrieval of similar images from a database is an active area of research. Without efficient feature extraction and proper indexing structures, similar image retrieval is time consuming because the query image is compared with all of the images of the database. The computational cost of feature extraction will be increased when the database is large. The problems of image retrieval which have been studied widely in the past are given: the reduction of the computational cost of feature extraction, the proper representation of features and the similarity measurement of the most similar images. To approach these issues and to get effective image retrieval, the statistical features are extracted from a grayscale image in the proposed approach by dividing the image into sub-blocks of different sizes. In the proposed block methods simply the color pixel values are computed using their distribution in each block. In each block, two color moments, mean and standard deviation, are computed using the pixel values to get the feature vectors of the different dimensions. The calculated features characterize the local information of blocks. The local features are combined in vector to describe the overall image. Using the proposed sub-block methods the computational cost is reduced as compared to complex segmentation which leads to efficient indexing of feature database. The matching of images is based on the local information of blocks which gives efficient retrieval of images. After various experiments, the results are analyzed and the performance is measured in terms of precision, recall and F-Score.

3.1.3.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into grayscale image to reduce computational cost as shown in Fig. 3.7 and it has been discussed in section 3.1.1.1.1.

3.1.3.2 Block Division

The grayscale image in the next step as shown in Fig. 3.7, is divided into non-overlapping sub-blocks of different sizes such as, Whole-Image-as-One-Block, 2-Blocks-Column-Wise, 2-Blocks-Row-Wise, 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks. Each block is a 2-dimensional matrix of 0 to 256 values. These values in each block will be used in the computations of the color moments to retrieve the similar images from the image database.

Let the image I be divided into L number of blocks such that:

$$ImgB = \{b_{1,}\ b_2\ ....b_L\} \tag{3.14}$$

Where b_i is the block of image I

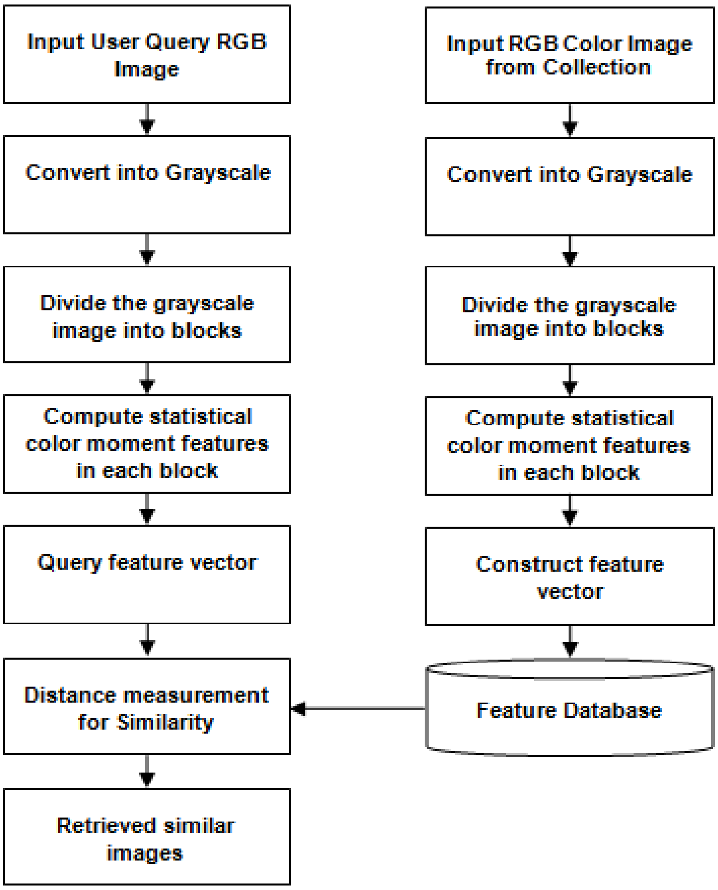


Figure 3.7 Block diagram of the proposed Approach-3.

3.1.3.3 Feature Extraction

For extraction of color features, the color moments is widely used because it has invariance to the rotation, scaling and translation of an image (Kodituwakku and Selvarajah, 2010). The proposed approach is also based on color moments which include the first order moment, mean, and the second order moment, standard deviation. These statistical features are computed in each block by using the pixel values.

The two statistical color moments are extracted from the sub-blocks of the grayscale image for all of the proposed different sizes of the block methods. These features are extracted by using the pixel values of the blocks from 0 to 256. The block diagram of the proposed approach is shown in Fig. 3.7.

Different numbers of color features are computed using sub-block methods of different sizes, for example for the 4×4 block method, total 4×4×2=32 features; for the 8×8 block method, total 8×8×2=128 features; for the 16×16 block method, total 16×16×2=512 features and so on, are calculated.

Let the mean be denoted by μ and the standard deviation by σ which are calculated using the x_{ji} pixel value, for $j = 1, 2,..., L$ blocks and for $i = 1,2,...,N$ pixels , then these values can be calculated (Bannour *et al.*, 2009; Kodituwakku and Selvarajah, 2010) as:

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} x_{ji} \tag{3.15}$$

$$\sigma_j = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{ji} - \mu_j)^2} \tag{3.16}$$

Where x_{ji} is the pixel value of *ith* pixel in block j and N is the total number of pixels in each block j.

After the calculation of these color features in all sub-blocks L then they are combined together to construct a feature vector *FV* as:

$$FV= \{\mu_1, \mu_2 ... \mu_L, \sigma_1, \sigma_2 ... \sigma_L\} \tag{3.17}$$

The feature vectors (*FVs*) of all of the images are constructed in the first step of the proposed approach as shown in Fig. 3.7 and are stored in the database. The feature vector *FV* is also calculated for the user query image by using the same approach in the second step as shown in Fig. 3.7. This query feature vector is compared with all of the feature vectors in the database to retrieve similar images. Algorithm 3 extracts the features of the sub-blocks to get feature vector *FV*.

Algorithm 3 Feature Extraction Algorithm of Approach-3

Input: Input image ***Img_file***, Size of blocks ***L=R×C***

Output: Feature vector ***fv***

1. Read and convert the ***Img_file*** into 2D matrix
    1.1 *Img* = ***imgread*** (*Img_file*);
2. Convert RGB ***Img*** into grayscale image
    2.1 *Img_gray* = ***RGB_to_Gray*** (*Img);*
3. Divide the image in sub-blocks
    3.1. For b = 1 to ***L***
    3.2. *B(b)* = ***Block_Conversion***(*Img_gray*, b);
    3.3 End for
4. Initialize *K=1;*
5. Construct feature vector ***fv*** by computing mean and standard deviation features
    5.1. For b= 1 to ***L***
    5.2. *fv(1, K)* = ***mean***(*B(b));*
    5.3. *fv(1, K+1)* = ***stddev***(*B(b));*
    5.4. *K = K + 2;*
    5.5 End for
6. Return feature vector *fv*
    6.1 ***Return fv***

3.1.4 Features Analysis for CBIR Based on the Statistical Texture Features using the Block Methods (Approach-4)

In order to reduce the computational cost and improve the retrieval efficiency, an approach Approach-4 is proposed which is based on the statistical texture features which are calculated in the non-overlapping sub-blocks of the image. The texture features are mean, standard deviation, skewness, flatness, uniformity, randomness and smoothness. For the feature extraction, the grayscale image is divided into non-overlapping blocks of different sizes like 2×2, 4×4, 8×8 etc. The block is used as a model of the probability distribution of the intensity levels. The statistical texture features are calculated by using the intensity level distribution in each block of the image. A feature vector is constructed by using the calculated features and used in the image retrieval.

3.1.4.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into grayscale image to reduce computation as shown in Fig. 3.8 and it has been discussed in section 3.1.1.1.1.

3.1.4.2 Block Division

The division of the grayscale image into sub-blocks of different sizes is discussed in section 3.1.3.2 and these steps are also shown in Fig. 3.8. The pixel values in each sub-block are used in computation of the statistical texture features to retrieve the similar images from the image database. Let the image I be divided into L number of sub-blocks such that:

$$ImgB = \{b_{1,} b_2 \ldots. b_L\} \tag{3.18}$$

Where b_i is the block of image I

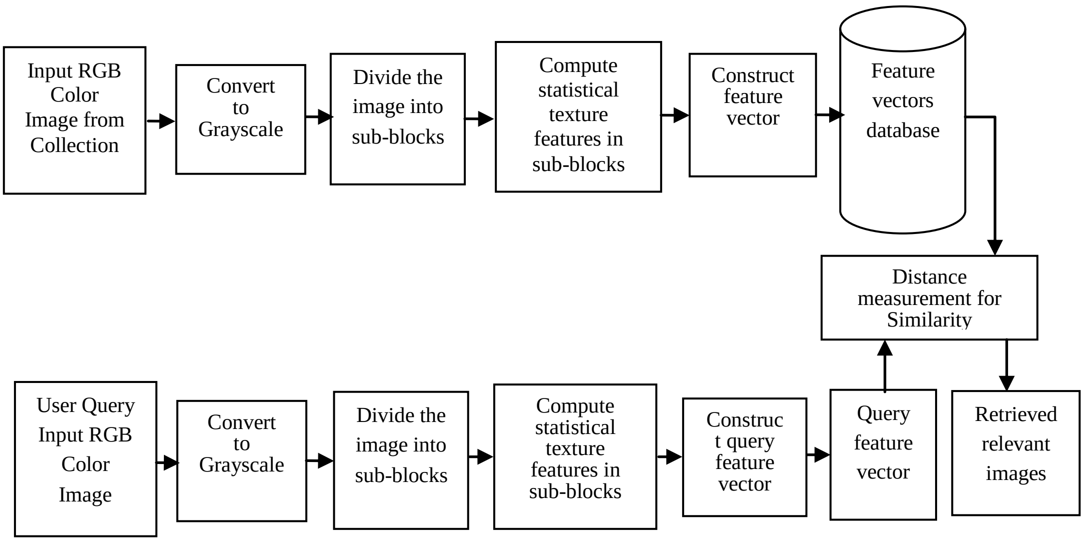


Figure 3.8 Block diagram of the proposed Approach-4.

3.1.4.3 Feature Extraction

The statistical texture features are considered useful for the classification and retrieval of similar images. These texture features provide information about the properties of the intensity level distribution in the image like uniformity, smoothness, flatness and contrast. The statistical texture features of the mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the sub-blocks of image *I*. Let $P(b_i)$ be the probability distribution of block b_i of image *I* such as:

$$B(b_i) = \sum_{j=1}^{br \times bc} x_{ij} \tag{3.19}$$

Where x_{ij} is the pixel value of *jth* pixel in the block *i* and $br \times bc$ is the size of block *i*.

$$P(b_i) = \frac{B(b_i)}{M \times N} \tag{3.20}$$

Where $B(b_i)$ is the sum of the pixels in block b_i, $M \times N$ is the total of the number of pixels in image *I* and $P(b_i)$ is the probability distribution of the intensity levels in block b_i. The texture features based on the block probability $P(b)$ are: mean, standard deviation, skewness, relative flatness or kurtosis, uniformity or energy, entropy and smoothness.

The mean is the texture feature that represents something about the brightness of the image. The mean measures the average value of the intensity values. If the mean is high, then it means that the image is bright and if low, then the image is dark. The mean is defined (Sergyan, 2008; Selvarajah and Kodituwakku, 2011) as:

$$mean = \sum_{b=1}^{L} bP(b) \tag{3.21}$$

The standard deviation is the second order moment and it shows the contrast of the gray level intensities. The low value of the standard deviation indicates low contrast and the high value shows the high contrast of the image. This can be computed (Selvarajah and Kodituwakku, 2011; Thawari and Janwe, 2011) as:

$$stddev = \sqrt{\sum_{b=1}^{L} (b - mean)^2 P(b)} \tag{3.22}$$

The third order moment is the skewness and it shows the skewness of the intensity values. It is the measurement of the inequality of the intensity level distribution about the mean. The value will be positive or negative of the skewness. The negative value indicates that a large number of intensity values are on the right side of the mean and the skewness of the tail of the intensity values is towards the left side of the distribution or the tail on the left side is longer than the right side. The positive value indicates that a large number of the intensity values are on the left side of the mean and the skewness of the tail of the intensity values is towards the right side of the distribution or the tail on right side is longer than the left side. The zero value indicates that the distribution of the intensity values is relatively equal on both sides of the mean. The skewness[6] is defined (Suresh et al., 2008; Selvarajah and Kodituwakku, 2011; Kekre and Sonawan, 2012) as:

$$SKEW = \frac{1}{(stddev)^3} \sum_{b=1}^{L} (b - mean)^3 P(b) \tag{3.23}$$

The fourth order moment is the kurtosis (flatness) and is used to measure the peak of the distribution of the intensity values around the mean. The high value of the kurtosis indicates that the peak of the distribution is sharp and the tail is longer and fat. The low value of the kurtosis indicates that the peak of the distribution is rounded and the tail is shorter and thinner. Kurtosis[7] is be defined (Suresh et al., 2008; Selvarajah and Kodituwakku, 2011; Kekre and Sonawan, 2012) as:

$$kurtosis = \frac{1}{(stddev)^4} \sum_{b=1}^{L} (b - mean)^4 P(b) \tag{3.24}$$

The energy feature measures the uniformity of the intensity level distribution. If the value is high, then the distribution is to a small number of intensity levels. Energy is defined (Sergyan, 2008; Selvarajah and Kodituwakku, 2011) as:

$$ENERGY = \sum_{b=1}^{L} [P(b)]^2 \tag{3.25}$$

The entropy measures the randomness of the distribution of the coefficient values over the intensity levels. If the value of the entropy is high, then the distribution is among more intensity levels in the image. This measurement is the inverse of energy.

---

[6] http://en.wikipedia.org/wiki/Skewness, Last vist on March 2012
[7] http://en.wikipedia.org/wiki/Kurtosis, last vist on March 2012

A simple image has low entropy while a complex image has high entropy. Entropy is defined (Sergyan, 2008; Selvarajah and Kodituwakku, 2011) as:

$$ENTROPY = -\sum_{b=1}^{L} P(b)\log_2[P(b)] \tag{3.26}$$

The smoothness texture is measured by using the standard deviation value. It is defined (Thawari and Janwe, 2011) as:

$$SM = 1 - \frac{1}{1+(stddev)^2} \tag{3.27}$$

After the calculation of these texture features, the feature vector *FV* of these values is constructed as:

$$FV = \{mean, stddev, SKEW, kurtosis, ENERGY, ENTROPY, SM\} \tag{3.28}$$

The feature vectors (*FVs*) of all of the images are constructed and stored to create a feature database. Algorithm 4 is used to extract the texture features in the blocks. The block diagram of the approach is shown in Fig. 3.8.

Algorithm 4 Feature Extraction Algorithm of Approach-4

Input: Input image ***Img_file***, Number of blocks ***L=R×C*** , Size of image ***S=M×N***

Output: Feature vector ***fv***

1. Read and convert the ***Img_file*** into 2D matrix
   - 1.1 *Img* = ***imgread*** (*Img_file*);
2. Convert RGB ***Img*** into grayscale image
   - 2.1 *Img_gray* = ***RGB_to_Gray*** (*Img);*
3. Divide the image in sub-blocks
   - 3.1. For b= 1 to ***L***
   - 3.2. *B(b)* = ***Blocks_Conversion*** (*Img_gray*, b);
   - 3.3 End for
4. Get the probability distribution of the pixels in the sub-blocks
   - 4.1. For b= 1 to ***L***
   - 4.2. *P(b)* = ***Sum****(B(b))/S*;
   - 4.3 End for

---

5. Calculate the statistical texture features

5.1 $mean = \sum_{b=1}^{L} bP(b)$

5.2 $stddev = \sqrt{\sum_{b=1}^{L} (b - mean)^2 P(b)}$

5.3 $SKEW = \frac{1}{(stddev)^3} \sum_{b=1}^{L} (b - mean)^3 P(b)$

5.4 $kurtosis = \frac{1}{(stddev)^4} \sum_{b=1}^{L} (b - mean)^4 P(b)$

5.5 $ENERGY = \sum_{b=1}^{L} [P(b)]^2$

5.6 $ENTROPY = -\sum_{b=1}^{L} P(b) \log_2 [P(b)]$

5.7 $SM = 1 - \frac{1}{1 + (stddev)^2}$

6. Construct the feature vector *fv* that combines the statistical texture features

6.1 *fv* = [mean *stddev SKEW kurtosis ENERGY ENTROPY SM*]

7. Return feature vector *fv*

7.1 Return ***fv***

3.1.5 Combination of the Color and Texture Features for CBIR using the Blocks Methods (Approach-5)

In this section, an approach Approach-5 is proposed for CBIR in which the color and texture features of images are combined for the retrieval of similar images using sub-block methods. Color and texture features are extracted in Approach-3 and Approach-4 using 9 different sub-block methods. In this proposed approach an attempt has been made to combine the color and texture features using the optimum 8×8 sub-block method with efficient retrieval and low computational cost, along with other sub-block methods of different sizes, of the proposed Approach-3 and approach-4. The statistical color moments of the mean and standard deviation, and the texture features of mean, standard deviation, skewness, flatness, energy, entropy and smoothness, are calculated in non-overlapping sub-blocks of the grayscale image. For feature extraction, the image is divided into non-overlapping sub-blocks of different sizes like

2×2, 4×4, 8×8 etc. The statistical texture features are calculated by using the intensity level distribution in each block of the image. A feature vector is constructed by combining the local color and texture features to retrieve the similar images.

3.1.5.1 Preprocessing

The proposed Approach-5 starts with the conversion of the input RGB color image into grayscale image to reduce the computational cost as shown in Fig. 3.9 and it has been discussed in section 3.1.1.1.1.

3.1.5.2 Block Division

The block division of a grayscale image is discussed in section 3.1.3.2. The values in each block are used in the computation of the color and texture features to retrieve the similar images from the image database as shown in Fig. 3.9. Let the image I be divided into L number of sub-blocks such that

$$IB = \{b_1, b_2 \ldots. b_L\} \tag{3.29}$$

Where b_i is the block of image, I.

3.1.5.3 Feature Extraction

The statistical color moments are extracted by computing the mean and standard deviation of the pixel values in the sub-blocks of the grayscale image for all of the proposed different sizes of the sub-block methods as discussed in section 3.1.3.3, using Eq. 3.15 and Eq. 3.16 to get the feature vector FVc using Eq. 3.17.

The statistical texture features, mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the blocks of IB using Eq. 3.19 and 3.20.

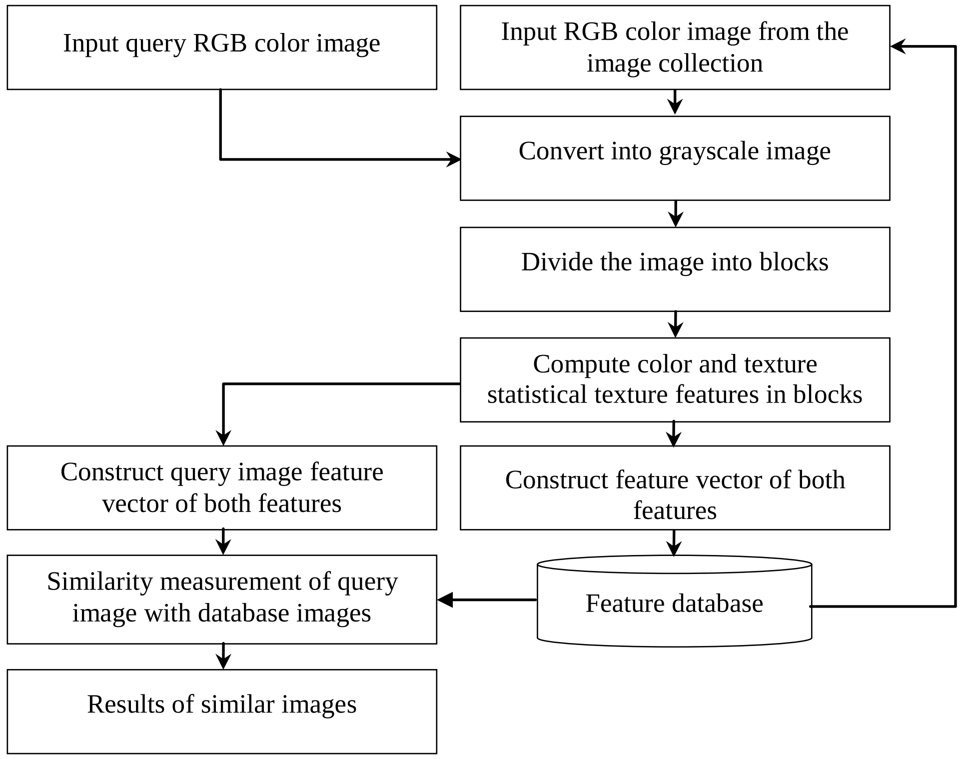


Figure 3.9 Block diagram of the proposed Approach-5.

The texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness, which have been discussed in section 3.1.4.3, are computed using Eq. 3.21 to 3.27.

After the calculation of these texture features, the texture feature vector *FVt* of these values is constructed using Eq. 3.28.

Color feature vector *FVc* is combined with the texture feature vector *FVt* to get an combined feature vector *FV* such as:

$$FV = FVc + FVt \tag{3.30}$$

The feature vectors (*FVs*) of all of the images are constructed and stored to create a feature database. Algorithm 5 is used to extract the color and texture features to get the feature vector *FV*. The block diagram of the algorithm is shown in Fig. 3.9.

Algorithm 5 Feature Extraction Algorithm of Approach-5

Input: Input image ***Img_file***, Size of blocks $\boldsymbol{L=R\times C}$, Size of image $\boldsymbol{S=M\times N}$

Output: Feature vector ***fv***

1. Call Algorithm 3 to compute color features using sub-block method to get feature vector $\boldsymbol{fv_c}$
2. Call Algorithm 4 to compute texture features using sub-block method to get feature vector $\boldsymbol{fv_t}$
3. Combine both feature vectors to get feature vector ***fv***

   3.1 $\boldsymbol{fv} = [\ \boldsymbol{fv_c}\quad \boldsymbol{fv_t}\ ]$
4. Return feature vector *fv*

   4.1 *Return* ***fv***

3.2 Statistical Texture Features Extraction in the Frequency Domain

Retrieval of the similar images in CBIR is based on the features of images. Features of the images are extracted and stored, which are then used in comparison with the given query example image's features to search for the desired similar images. The identification and extraction of the appropriate image features is a challenging issue in CBIR. The storage space and manipulation of the images are the other issues caused by the availability of a huge number of images due to the advanced development in the image capture devices, Internet and computer hardware. Consequently, most of the images at present, are represented in a compressed format like JPEG (Nezamabadi-pour and Saryazdi, 2005; Liu *et al.*, 2007). JPEG(Joint Photographic Expert Group) is a compressed format of images with good quality (Jeong, 1997), where it can been seen in Fig. 3.10 that an image of a rose with dimension of 384×256 in the compressed JPEG format with a size of 18.9 KB and Bitmap without compression with a size of 288 KB formats. There is no such apparent difference visually but there exist huge difference in size.

In the compressed domain, the features can be extracted directly without decoding into pixels by using frequency transformation like Discrete Cosine Transformation (DCT) which is used as the component of the compression process (Mandal *et al.*, 1999). In DCT, some information from the image is eliminated and some important

JPEG, size 18.9 KB Bitmap, size 288 KB

Figure 3.10 An image of rose with JPEG and Bitmap formats.

information is left behind during compression, which can be used to play an important role in the retrieval of similar images (Zhong and Defee, 2005).

There are various discrete transformation like discrete cosine transformation (DCT), discrete Fourier transformation (DFT) and discrete wavelet transformation. These transformations will be prominent domains for retrieval of compressed images like JPEG. The selection will be made on the performance of transformation. In this work of research we have selected DCT for feature indexing and retrieval due to some advantages over DWT such that DCT is less expensive for hardware or software implementation than DWT, for instance the two-dimensional 8×8 DCT blocks in an efficient algorithm, involves only 54 multiplications (Feig, 1990) where as in DWT each coefficient requires at least one multiplication which increases the computational cost.

DCT is a cosine part of Fourier transformation while the DWT requires more computation in time frequency. So it is acknowledged by (Shen, 2013) that the computational cost of DWT is higher than DCT, the regions of image become blurred and noise is produced near edges of regions by using the larger DWT basis function or wavelet filters, the compression time of DWT is longer than DCT, at low compression the quality of DWT is lower than JPEG. Therefore we focus on the performance of DCT in this research work.

In our work, statistical quantized histogram texture features are extracted from the grayscale image in the frequency domain, using hybrid techniques to develop an efficient and effective CBIR to improve the retrieval performance. We proposed various approaches in which the grayscale image is transformed into non-overlapping 8×8 DCT blocks. The first element of each block is called DC, which is the average of the intensity values of the blocks and the remaining are called AC coefficients corresponding to the frequencies. The AC coefficients describe the texture information (Shan and Liu, 2009). The statistical texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated in all of the blocks using the block coefficients in various approaches using different techniques. After the feature extraction and creation of the feature database, the similarity is measured between the query image and the database images. For this purpose, the Euclidean distance is used to calculate the distance between the two feature vectors of the images to retrieve relevant images.

3.2.1 Quantized Histogram Texture Features Based on Median, Median with Edge Extraction Method and Laplacian Filters (Approach-6)

In this proposed Approach-6, our main contribution is to perform the experimental comparative analysis of the statistical quantized histogram texture features for the effective image retrieval in the DCT domain based on the median and Laplacian filters. The approach starts with an 8×8 DCT block transformation of the filtered histogram equalized grayscale image. The histograms of the DC and the first three AC coefficients are constructed by quantizing them into 32 bins. Then, the statistical texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the histogram bins of all of the blocks. These features construct a feature vector to retrieve similar images from the database. These vectors are used in a similarity measurement to compare the query image vector with the database vectors. The comparison of the results of the quantized histogram texture features based on the median, median with edge extraction and Laplacian filters, is demonstrated and give the optimal performance in terms of image retrieval in the DCT domain.

3.2.1.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into a single component grayscale image to reduce the computational cost because a color image consists of Red, Green and Red components (Anjum and Javed, 2007). The grayscale image is converted into a histogram equalized (HE) image, to make the image's intensity levels equal to get a high contrast image as shown in Fig 3.11. Then median, edge extraction and Laplacian filters are applied to HE image respectively.

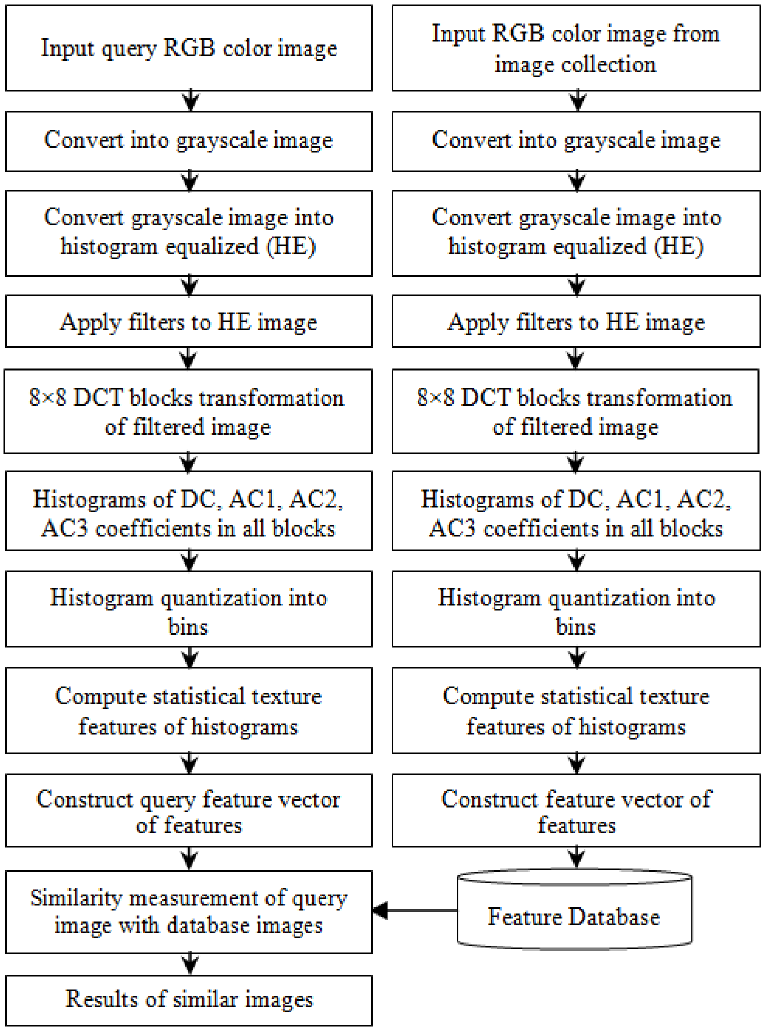


Figure 3.11 Block diagram of the proposed Approach-6.

The preprocessing and filter methods including median filter, median filter with edge extraction and Laplacian filter, have been discussed in detail in section 3.1.2.1 and are applied to the grayscale image to get a more enhanced and sharpened image.

3.2.1.2 DCT Block Transformation

The filtered image is divided into non-overlapping 8×8 blocks. Then, all of these blocks are transformed into DCT blocks in the frequency domain. The 8×8 DCT block transformation is simpler and quicker. By taking large block size like 64×64, then the computational cost and loss of information will be high. Moreover, by taking small block size like 4×4 then the compression will be low. Therefore 8×8 block size is an optimal choice for DCT transformation. Each block consists of values called coefficients. These coefficients are represented in a matrix of two dimensions of rows and columns. Let the two dimensional DCT block with a size of $N \times N$ for $x,y=0,1,2,\ldots,N-1$ is calculated as:

$$F(u,v) = \frac{1}{\sqrt{2N}} c(u)c(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \times$$
$$\cos\left[\frac{(2y+1)v\pi}{2N}\right] \tag{3.31}$$
$$c(u) = \begin{cases} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{cases}$$

Where *F(u, v)* is the transformed block and *f(x, y)* is the element of the block.

The first uppermost DCT coefficient in the DCT block is *F(0,0)* in Eq. 3.31, it is also called the DC coefficient and it represents the average intensity value of a block. The DC coefficient is also described as the energy of the block. The other coefficients of the DCT blocks are called AC coefficients, which correspond to the different frequencies (co sinusoidal).

After the DCT transformation, the DC coefficients of all of the blocks and the first three AC coefficients (AC1, AC2 and AC3) are selected in a zigzag order as shown in Fig. 3.12. The DC and AC coefficients of all the blocks are used to construct the histograms.

3.2.1.3 Histogram Quantization

The number of occurrences of the DC coefficients in all the DCT blocks is called the DC histogram and the number of occurrences of the AC coefficients is called AC histogram. Let the DC histograms be quantized into L (4, 8, 16 or 32) bins then the DC histogram H_{DC} can be calculated as:

$$H_{DC}= \{h(b_1)_{DC}, h(b_2)_{DC} ....h (b_L)_{DC} \} \tag{3.32}$$

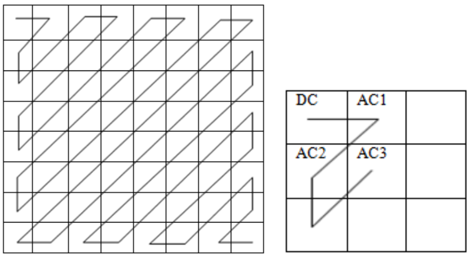


Figure 3.12 Selection of 8×8 DCT block coefficients in zigzag order.

Where $h(b_i)_{DC}$ is the frequency of the DC coefficients in bin b_i and H_{DC} is the histogram of L bins.

The AC histograms for the selected AC coefficients are also quantized into L bins and histograms are computed as:

$$H_{AC1}= \{h(b_1)_{AC1}, h(b_2)_{AC1}, ..., h(b_L)_{AC1}\} \tag{3.33}$$

$$H_{AC2}= \{h(b_1)_{AC2}, h(b_2)_{AC2}, ..., h(b_L)_{AC2}\} \tag{3.34}$$

$$H_{AC3}= \{h(b_1)_{AC3}, h (b_2)_{AC3}, ..., h(b_L)_{AC3}\} \tag{3.35}$$

Where $h(b_i)_{AC1}$, $h(b_i)_{AC2}$ and $h(b_i)_{AC3}$ are the frequencies of AC1, AC2 and AC3 coefficients in histograms H_{AC1}, H_{AC2} and H_{AC3} using the histogram quantization scheme of L bins.

Quantization schemes with different number of bins from 4 to 32 have been discussed in sections 3.1.1.2. Here in this proposed approach different quantization schemes are analyzed in the frequency domain based on the filtered values of images for the extraction of the texture features.

3.2.1.4 Feature Extraction

The statistical texture features: mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the histogram bins of the histograms H_{DC}, H_{AC1}, H_{AC2}, *and* H_{AC3} using Eq. 3.31 to 3.35.

Let $P(b)$ be the probability distribution of bin b in each of the four histograms H using Eq. 3.31 to 3.35 with L bins then it is calculated as:

$$P(b) = \frac{H(b)}{M} \tag{3.36}$$

Where M is the total number of blocks in image I.

The texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness, which have been discussed in section 3.1.4.3, are computed using the block probability $P(b)$ and Eq. 3.21 to 3.27.

After the calculation of these texture features, the feature vector fv of these values is constructed using Eq. 3.28.

The feature vectors (fv_s) are calculated for all of the histograms using Eq. 3.28 such that feature vector fv_{HDC} is calculated for the histogram H_{DC}, fv_{HAC1} for H_{AC1}, fv_{HAC2} for H_{AC2} *and* fv_{HAC3} for H_{AC3}. The four feature vectors are combined to get a single feature vector FV of the features as:

$$FV = [fv_{HDC}, fv_{HAC1}, fv_{HAC2}, fv_{HAC3}] \tag{3.37}$$

The feature vectors (*FVs)* of all of the images are constructed and stored to create a feature database. Algorithm 6 is used to extract the quantized histogram texture features based on median and Laplacian filters in the DCT blocks. The feature vector of the user query is also constructed in the same way and compared with the feature vectors of the database for similarity and retrieval of relevant images. The feature extraction process is shown in the block diagram of the proposed approach in Fig. 3.11.

Algorithm 6 Feature Extraction Algorithm of Approach-6

Input: Input image ***Img_file***, Number of bins ***Bins, filter_type,*** Blocks ***B***

Output: Feature vector ***fv***

1. Read and convert the ***Img_file*** into 2D matrix
    1.1 *Img* = ***imgread***(*Img_file*)
2. Convert RGB *Img* into grayscale image
    2.1 *Img_gray* = ***RGB_to_Gray*** (*Img)*

3. Convert grayscale *Img_gray* into Histogram Equalized image

   3.1 *Img_he* = ***hist_Eq*** (*Img_gray)*

4. Apply filters to *Img_he*

   4.1. *F_img* =***Filters*** (*Img_he,* ***filter_type***);

5. conversion of *F_img* into 8×8 DCT blocks

   5.1 For b= 1 to ***B***

   5.2 *dct_blk (b)*=***DCT_Trans formation***(*F_img*, *b*)

   5.3 End for

6. Get the DC, AC1, AC2, AC3 coefficients of all blocks by DC(*dct_blk (Bi)*), AC1(*dct_blk (Bi)*), AC2(*dct_blk (Bi)*) and AC3(*dct_blk (Bi)*) for i=1 to ***B***

7. Get the quantized histograms of ***Bins*** bins by

   7.1 DC_h =***Histogram***(DC, ***Bins***)

   7.2 AC1_h =***Histogram***(AC1, ***Bins***)

   7.3 AC2_h =***Histogram***(AC2, ***Bins***)

   7.4 AC3_h =***Histogram***(AC3, ***Bins***)

8. Get the probability distribution of histograms $P(b)_{DC}$ = DC_h(b)/B, $P(b)_{AC1}$ = AC1_h(b)/B, $P(b)_{AC2}$ = AC2_h(b)/B, $P(b)_{AC3}$ = AC3_h(b)/B for b=1 to ***Bins.***

9. Calculate statistical texture features for DC, AC1, AC2 and AC3 coefficients separately to get four feature vectors

   9.1 $$mean = \sum_{b=1}^{Bins} bP(b)_{DC}$$

   9.2 $$stddev = \sqrt{\sum_{b=1}^{Bins} (b - mean)^2 P(b)_{DC}}$$

   9.3 $$SKEW = \frac{1}{(stddev)^3} \sum_{b=1}^{Bins} (b - mean)^3 P(b)_{DC}$$

   9.4 $$kurtosis = \frac{1}{(stddev)^4} \sum_{b=1}^{Bins} (b - mean)^4 P(b)_{DC}$$

   9.5 $$ENERGY = \sum_{b=1}^{Bins} [P(b)_{DC}]^2$$

9.6 $ENTROPY = -\sum_{b=1}^{Bins} P(b)_{DC} \log_2[P(b)_{DC}]$

9.7 $SM = 1 - \frac{1}{1+(stddev)^2}$

9.8 fv_{HDC} = *[mean stddev SKEW kurtosis ENERGY ENTROPY SM]*

9.9 Similarly calculate the feature vectors fv_{HAC1} fv_{HAC2} and fv_{HAC3} using steps 9.1 to 9.8

10. Combine feature vectors of coefficients to get combined feature vector

10.1 $fv = [fv_{HDC}\ fv_{HAC1}\ fv_{HAC2}\ fv_{HAC3}]$

11. Return feature vector *fv*

11.1. *Return* ***fv***

3.2.2 Analysis of the Distance Metrics in CBIR using the Statistical Quantized Histogram Texture Features in the Frequency Domain (Approach-7)

In this proposed Approach-7, our main contribution is to show the performance of the image retrieval by extracting quantized histogram statistical texture features efficiently and by matching the query image with the database images effectively using various distance metrics in the DCT domain. The proposed method is started with the non-overlapping 8×8 DCT block transformation of the grayscale image. The histograms of the DC and the first three AC coefficients are constructed. The statistical texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the coefficients in different quantization bins of the histograms of all the blocks. The computed features are used to measure the similarity between the query image and the database images by using various distance metrics. The performance is analyzed on the basis of the results of various distance metrics using different quantization schemes in the DCT domain.

3.2.2.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into grayscale image to reduce computation as shown in Fig. 3.13 and it has been discussed in section 3.1.1.1.1.

3.2.2.2 DCT Block Transformation

The grayscale image transformation into DCT blocks has been discussed in 3.2.1.2.

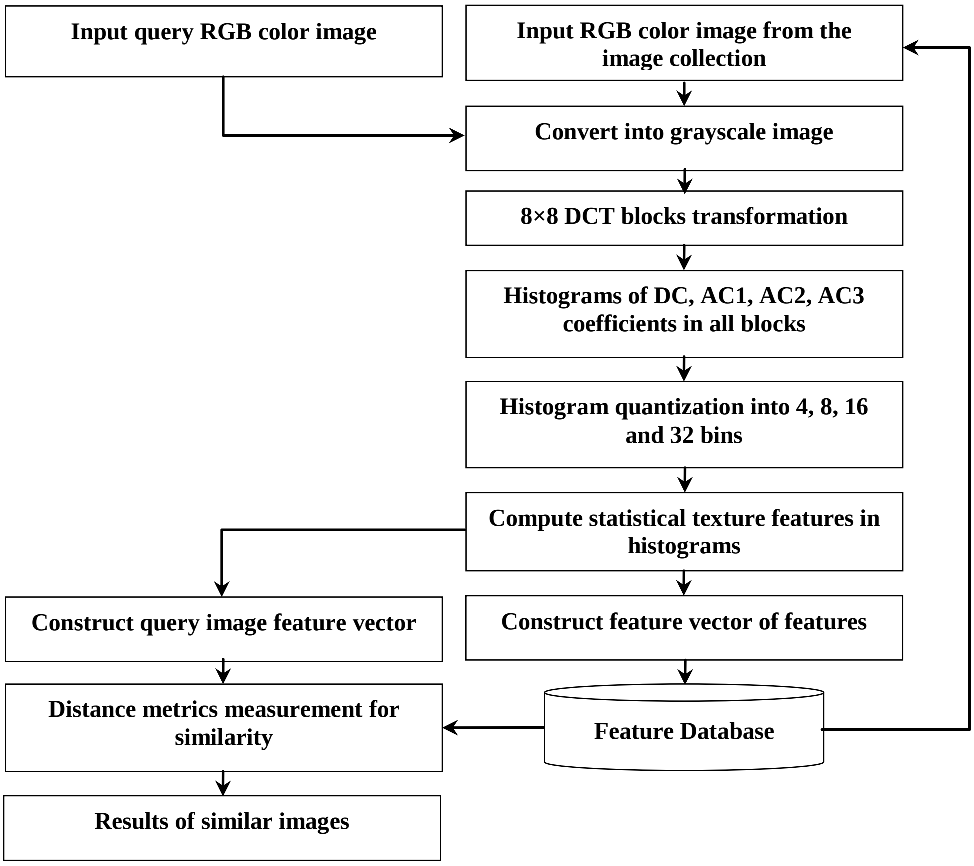


Figure 3.13 Step wise block diagram of the proposed Approach-7.

3.2.2.3 Histogram Quantization

The histograms of the DC and AC coefficients are constructed and quantized into L bins. In this proposed approach, the histograms are quantized into 4, 8, 16 and 32 bins. The histogram construction and quantization have been discussed in section 3.2.1.3, in which the DC quantized histogram, H_{DC}, and AC histograms, H_{AC1}, H_{AC2}

and $H_{AC3,}$ are generated using Eq. 3.32 to 3.35. The proposed statistical texture features are extracted in these histograms.

3.2.2.4 Feature Extraction

The first issue in CBIR is to extract the features of an image efficiently and then represent them in a particular form to be used effectively in matching of the images. The statistical texture features are considered useful for the classification and retrieval of similar images. The statistical texture features are extracted in the proposed method. The proposed texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the histogram bins of the histograms of the DC, AC1, AC2 and AC3 coefficients.

The texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness, which have been discussed in section 3.2.1.4, are computed using the DCT block probability $P(b)$ to create the feature vectors, fv_{HDC}, fv_{HAC1}, fv_{HAC2} *and* fv_{HAC3}, of the histograms: H_{DC}, H_{AC1}, H_{AC2} *and* $H_{AC3.}$. These feature vectors are combined to get a single feature vector (*FV*) of the features using Eq. 3.37.

The feature vectors (*FVs*) of all of the images are constructed and stored to create a feature database. Algorithm 7 is used to extract the texture features in the DCT blocks. The feature vector of the user query is also constructed in the same way and compared with the feature vectors of the database for the similarity and retrieval of the relevant images. The block diagram of the method is shown in Fig. 3.13.

Algorithm 7 Feature Extraction Algorithm of Approach-7

Input: Input image ***Img_file***, Number of bins ***Bins,*** Blocks ***B***

Output: Feature vector ***fv***

1. Read and convert the ***Img_file*** into 2D matrix
    - 1.1 *Img* = ***imgread***(*Img_file*);
2. Convert RGB *Img* into grayscale image
    - 2.1 *Img_gray* = ***RGB_to_Gray*** (*Img);*

---

3. Transformation of *Img_gray* into 8×8 DCT blocks

   5.1 For b= 1 to ***B***

   5.2 *dct_blk (b)*=***DCT_Transformation***(*Img_gray*, *b*)

   5.1 End for

4. Get the DC, AC1, AC2, AC3 coefficients of blocks by DC(*dct_blk (Bi)*), AC1(*dct_blk (Bi)*), AC2(*dct_blk (Bi)*) and AC3(*dct_blk (Bi)*) for i=1 to ***B***

5. Get the quantized histograms of bins ***Bins*** by

   5.1 DC_h =***Histogram***(DC, ***Bins***)

   5.2 AC1_h =***Histogram***(AC1, ***Bins***)

   5.3 AC2_h =***Histogram***(AC2, ***Bins***)

   5.4 AC3_h =***Histogram***(AC3, ***Bins***)

6. Get the probability distribution of histograms $P(b)_{DC}$ = DC_h(b)/B, $P(b)_{AC1}$ = AC1_h(b)/B, $P(b)_{AC2}$ = AC2_h(b)/B, $P(b)_{AC3}$ = AC3_h(b)/B for b=1 to ***Bins.***

7. Calculate the statistical texture features in the histograms DC_h, AC1_h, AC2_h and AC3_h for b=1 to ***Bins*** using the *P(b)* to get feature vector

   7.1 $$mean = \sum_{b=1}^{Bins} bP(b)_{DC}$$

   7.2 $$stddev = \sqrt{\sum_{b=1}^{Bins} (b - mean)^2 P(b)_{DC}}$$

   7.3 $$SKEW = \frac{1}{(stddev)^3} \sum_{b=1}^{Bins} (b - mean)^3 P(b)_{DC}$$

   7.4 $$kurtosis = \frac{1}{(stddev)^4} \sum_{b=1}^{Bins} (b - mean)^4 P(b)_{DC}$$

   7.5 $$ENERGY = \sum_{b=1}^{Bins} [P(b)_{DC}]^2$$

   7.6 $$ENTROPY = -\sum_{b=1}^{Bins} P(b)_{DC} \log_2 [P(b)_{DC}]$$

   7.7 $$SM = 1 - \frac{1}{1 + (stddev)^2}$$

8. Construct the feature vector of calculated features

   8.1 fv_{DC}=[*mean stddev SKEW kurtosis ENERGY ENTROPY SM*]

9. Similarly calculate the feature vectors fv_{AC1} fv_{AC2} and fv_{AC3} using steps 7.1

to 7.7.

10. Combine feature vectors of coefficients to get combined feature vector

    10.1 $fv = [fv_{DC}\ fv_{AC1}\ fv_{AC2}\ fv_{AC3}]$

11. Return feature vector *fv*

    11.1 *Return* ***fv***

3.2.2.5 Distance Metrics

Once the feature database of the images is created with the feature vectors in the first step of the approach as shown in Fig. 3.13 using Algorithms 7, then the user can give an image as a query image to retrieve the similar images from the database. In the second step of the approach, the feature vector of the query image is computed by using the same steps as discussed in section 3.2.1.4.

The similarity measurement is the second issue in CBIR in which the query image is compared with the target images of the database. To measure the similarity between the query image and the target database images, the difference is calculated between the query feature vector and the target database feature vectors by using the distance metrics. The small difference between the two feature vectors indicates the large similarity and small distance. The vectors of the images with small distance are most similar to the query images. The distance metrics which are included in this work are the Sum of Absolute Difference (SAD), Sum of Squared of Absolute Differences (SSAD) (Selvarajah and Kodituwakku, 2011), Euclidean distance, City block distance, Canberra distance, Maximum value metric and Minkowski distance (Sergyan, 2008).

Let the query feature vector is represented by ***Q*** and the target feature vector by ***T*** of the database for the seven distance metrics to calculate the difference between the two vectors for similarity:

3.2.2.5.1 Sum of Absolute Difference (SAD)

The Sum of Absolute Difference (SAD)[8] is a very straightforward distance metric and extensively used for computing the distance between the images in CBIR to get the similarity. In this metric, the sum of the differences of the absolute values of the two feature vectors are calculated (Selvarajah and Kodituwakku, 2011). The similarity is decided on the computed value of the distance. This distance metric can be calculated as:

$$D(Q,T) = \sum_{i=1}^{n} \left(|Q_i| - |T_i| \right) \tag{3.38}$$

Where n is the number of features, $i=1, 2\ldots, n$. Both images are the same for $D(Q, T)=0$ and the small value of D shows the relevant image to the query image. Image retrieval is performed by using Algorithm 8.

The distance metric SAD is a simple method to search for the similar images in the database to the query image, automatically, but it can be sensitive and untrustworthy towards the consequences of the background issues of the image, such as variations in size, color, illumination and the direction of light[9].

3.2.2.5.2 Sum of Squared Absolute Difference (SSAD)

In this distance metric, the sum of the squared differences of the absolute values of the two feature vectors are calculated. This distance metric can be calculated (Selvarajah and Kodituwakku, 2011) as:

$$D(Q,T) = \sum_{i=1}^{n} \left(|Q_i| - |T_i| \right)^2 \tag{3.39}$$

It has some computational complexity due to the square of the differences as compared to SAD. However, squaring always gives a positive value but it highlights a big difference. The distance metric SSAD can be used in the spatial as well as in the

---

[8] http://en.wikipedia.org/wiki/Sum_of_absolute_differences, last visit on September 18, 2012.
[9] http://en.wikipedia.org/wiki/Sum_of_absolute_differences, last visit on September 18, 2012.

transformed domains but in the transform domain the calculated value depends upon the quality of the compression[10].

3.2.2.5.3 Euclidean Distance

This distance metric is most commonly used for the similarity measurement in image retrieval because of its effectiveness. It measures the distance between the two vectors of images by calculating the square root of the sum of the squared absolute differences and it can be calculated (Wang *et al.*, 2005; Sergyan, 2008) as:

$$D(Q,T)=\sqrt{\sum_{i=1}^{n}\left(\left|Q_i-T_i\right|\right)^2} \tag{3.40}$$

3.2.2.5.4 City Block Distance

This distance metric is also called Manhattan distance. This distance metric is computed by the sum of the absolute of differences between the two feature vectors of the images and can be calculated[11] (Sergyan, 2008) as:

$$D(Q,T)=\sum_{i=1}^{n}\left|Q_i-T_i\right| \tag{3.41}$$

3.2.2.5.5 Canberra Distance

The city block distance metric gives large value for the two similar images which create dissimilarity between the similar images. Hence, each feature pair difference is normalized by dividing it by the sum of a pair of features. This metric is used for the numerical measurement of the distance between the query and database feature vectors and can be calculated (Sergyan, 2008) as:

$$D(Q,T)=\sum_{i=1}^{n}\frac{\left|Q_i-T_i\right|}{\left|Q_i\right|+\left|T_i\right|} \tag{3.42}$$

---

[10] http://siddhantahuja.wordpress.com/tag/sum-of-squared-differences/ , last visit on September 18, 2012.
[11] http://people.revoledu.com/kardi/tutorial/Similarity/QuantitativeVariables.html , last visit on September 18, 2012.

The value of this method is arranged in ascending order such that the top most shows high similarity. It has a similarity with the city block distance metric. It has a good effect for the data which are spread about the origin (Schulz, 2007).

3.2.2.5.6 Maximum Value Distance

This distance metric is also called the Chebyshev distance. This distance is used to get the largest value of the absolute differences of a pair of features of the feature vectors and can be calculated (Sergyan, 2008) as:

$$D(Q,T) = \max\left\{\left|Q_1 - T_1\right|, \left|Q_2 - T_2\right|, \ldots, \left|Q_n - T_n\right|\right\} \tag{3.43}$$

The distance value is the maximum of the difference of the features of the pair of images, which shows the maximum dissimilarity of the two images.

3.2.2.5.7 Minkowski Distance

The generalized form of the distance can be defined as:

$$D(Q,T) = \left[\sum_{i=1}^{n}\left(\left|Q_i - T_i\right|\right)^p\right]^{1/p} \tag{3.44}$$

Where p is a positive integer.

This generalized form gives other distance metrics for positive values of p, for example $p=1$ gives the city block distance and $p=2$ give the Euclidean distance. In this work, for the comparison of the distance metrics, we also take $p=3$ as the Minkowski distance. Algorithm 8 is used to retrieve images by using, one by one, the distance metrics in Eq. 3.38 to 3.44.

Algorithm 8 Image Retrieval Algorithm of Approach-7

Input: A query image Q, target images $T_1, \ldots, T_N$

Output: A sorted list of target images

1. Select a query image Q

1.1. *Img_file* = ***get_image***(*Q*)

2. Extract ***n*** number of features of *Q* to create feature vector ***fv***

   2.1. *Qfv* =***Feature_Extraction*** (*Img_file*) *//call Algorithm 9*

3. For each target image T_i

4. Access the feature vector of target image T_i from feature database *fv_DB*

   4.1. *Tfv = fv_DB(* T_i*)*

5. Calculate the sum of Absolute Difference (SAD) for *n* number of features and get the distance value.

   5.1. $D(Q, T_i) = \sum_{j=1}^{n}\left(\left|Qfv_j\right| - \left|Tfv_j\right|\right)$

6. End for

7. Rank all the target images according to $D(Q, T_i)$.

8. Similarly calculate the distance values for the following distance metrics by repeating the steps from 1 to 7

   8.1. Sum of Squared Absolute difference (SSAD*)*

      8.1.1. $D(Q, T_i) = \sum_{j=1}^{n}\left(\left|Qfv_j\right| - \left|Tfv_j\right|\right)^2$

   8.2. Euclidean distance

      8.2.1. $D(Q, T_i) = \sqrt{\sum_{j=1}^{n}\left(\left|Qfv_j - Tfv_j\right|\right)^2}$

   8.3. City block distance

      8.3.1. $D(Q, T_i) = \sum_{j=1}^{n}\left|Qfv_j - Tfv_j\right|$

   8.4. Canberra distance

      8.4.1. $D(Q, T_i) = \sum_{j=1}^{n}\frac{\left|Qfv_j - Tfv_j\right|}{\left|Qfv_j\right| + \left|Tfv_j\right|}$

   8.5. Maximum value distance

      8.5.1. $D(Q, T_i) = \max\left\{\left|Qfv_1 - Tfv_1\right|, \left|Qfv_2 - Tfv_2\right|, \ldots, \left|Qfv_n - Tfv_n\right|\right\}$

   8.6. Minkowski distance

      8.6.1. $D(Q, T_i) = \left[\sum_{j=1}^{n}\left(\left|Qfv_j - Tfv_j\right|\right)^3\right]^{1/3}$

3.2.3 Experimental Analysis of the Combination of the Statistical Quantized Histogram Texture Features in the Frequency Domain (Approach-8)

In this proposed Approach-8, the main contribution is to show the performance of the image retrieval by the combination of different quantized histogram texture features in the frequency (DCT) domain.

3.2.3.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into grayscale image to reduce the computational cost as shown in Fig. 3.13 and it has been discussed in section 3.1.1.1.1.

3.2.3.2 DCT Block Transformation

In this approach, we start with the non-overlapping 8×8 DCT block transformation of a grayscale image as shown in the block diagram in Fig. 3.13 and that is discussed in section 3.2.1.2 using Eq. 3.31. After the DCT transformation, the DC and AC coefficients of all of the blocks are selected in a zigzag order as shown in Fig. 3.13. All of these DC and AC coefficients are used to construct histograms.

3.2.3.3 Histogram Quantization

The histograms of the DC and AC coefficients are constructed and quantized into $L=32$ bins. The histogram construction and quantization have been discussed in section 3.2.1.3, in which the DC quantized histogram H_{DC} and AC histograms, H_{AC1}, H_{AC2} *and* H_{AC3} are generated using Eq. 3.32 to 3.35.

3.2.3.4 Feature Extraction

The statistical texture features of mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated in different combinations like single, two, three and four feature combinations by using the probability distribution of the coefficients in the histogram bins of the histograms of the coefficients.

The texture features, which have been discussed in detail in section 3.1.4.3, are computed using the DCT block probability $P(b)$ to create the feature vectors fv_{HDC}, fv_{HAC1}, fv_{HAC2} and fv_{HAC3}, of the histograms H_{DC}, H_{AC1}, H_{AC2} *and* H_{AC3}. These feature vectors are combined to get a single feature vector (*FV*) of the features using Eq. 3.37. The size of the feature vectors varies due to the different combinations of features; for example, by considering the mean and standard deviation, then each feature vector of the DC and the first three AC coefficients will have two features. Then, these four feature vectors with two features are combined to get a single feature vector of eight features.

The feature vectors (*FVs*) of all of the images are constructed and stored to create a feature database using Algorithms 7 with different combinations of features. The block diagram of the method is shown in Fig. 3.13.

3.2.4 Analysis of the Experimental Performance of the Combination of the Statistical Texture Features in the Spatial and Frequency Domains for CBIR (Approach-9)

In this section, an approach Approach-9 is proposed in which the statistical texture features, mean, standard deviation, skewness, flatness, energy, entropy and smoothness, of the grayscale images in the spatial domain are combined with the statistical texture features in the DCT domain for the retrieval of similar images in CBIR. The retrieval by texture features in the spatial domain using the sub-block methods have been discussed in section 3.1.4 using the 9 different sub-block methods, separately. In the frequency domain, the statistical texture features have been used for the retrieval of similar images and are discussed in section 3.2.1 and 3.2.3.

The statistical texture features are extracted from the sub-blocks of an image in the spatial domain in Approach-4. In order to further improve the retrieval performance of CBIR, the statistical texture features are extracted from the DCT transformed blocks of the images in the frequency domain in Approach-7. The features are extracted from the quantized histograms using only the DC and first three AC coefficients of all of the blocks. The performance is not only effective in retrieval but also efficient in the computational cost. Since both Approach-4 and Approach-7 are effective and efficient in retrieval as well as in computation, therefore the feature vectors of the two approaches using 8×8 blocks, are combined together to further improve the retrieval performance of CBIR effectively.

3.2.4.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into grayscale image to reduce the computational cost as shown in Fig. 3.14 and it has been discussed in section 3.1.1.1.1.

3.2.4.2 Feature Extraction

The proposed approach starts with the conversion of the RGB color image into a grayscale image. Then, the texture features are extracted from the grayscale image. The feature extraction is carried out in two steps: Extraction in the spatial domain using the block method and in the DCT domain using the histogram method as shown in Fig. 3.14.

3.2.4.2.1 Texture Feature Extraction using Block Method in the Spatial Domain

The feature extraction in the spatial domain using sub-block methods has been discussed in section 3.1.4.3; that is using simple sub-blocks of image, is a simpler and quicker method with good retrieval accuracy as compared to the segmentation of the images. The grayscale image is divided into 8×8 non-overlapping L sub-blocks. Each block is in a 2-dimensional matrix of 0 to 256 values. The statistical texture features, mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the pixel values in all of the blocks using Eq. 3.19 to 3.27. After the calculation of these texture features, the texture feature vector FV_p is constructed using Eq. 3.28, as shown in Fig. 3.14. In the next step, the texture features are calculated in the DCT domain.

3.2.4.2.2 Texture Feature Extraction using Histogram Method in the Frequency Domain

The statistical texture feature extraction using the histogram method in the DCT domain has been discussed in Approach-7 and sections 3.2.2.4. The same grayscale image is divided into non-overlapping 8×8 blocks. Then, all of these blocks are transformed into DCT blocks in the frequency domain. Each block is in a 2-dimensional matrix. After the DCT transformation, the DC and the first three AC coefficients of all of the blocks are selected in a zigzag order and are used to construct the histograms H_{DC}, H_{AC1}, H_{AC2} and H_{AC3}, *as* shown in Fig. 3.14. The histograms are quantized into 32 bins.

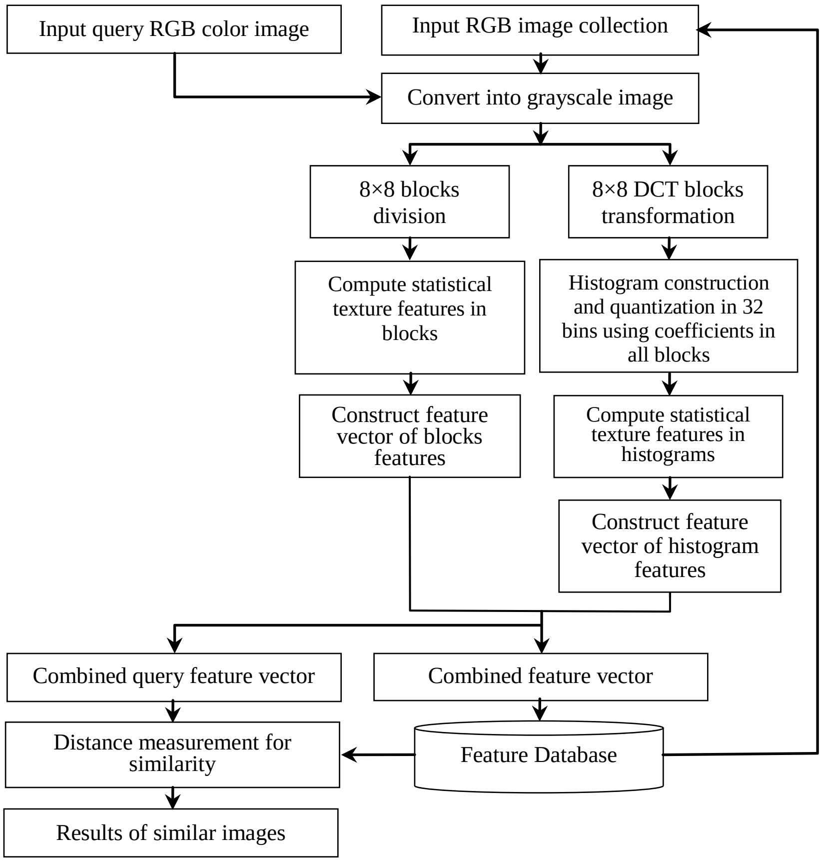


Figure 3.14 Block diagram of the proposed Approach-9 based on the combination of the texture features in the spatial and frequency domains.

The statistical texture features, mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the histogram bins of the histograms H_{DC}, H_{AC1}, H_{AC2} and H_{AC3}, to create the feature vectors fv_{HDC}, fv_{HAC1}, fv_{HAC2} and fv_{HAC3} as discussed in Approach-6 and section 3.2.1.4. These features vectors are combined to get a single feature vector (FV_d) using Eq. 3.37.

After the calculation of the feature vectors FV_p in the spatial domain and FV_d in the frequency domain, are combined to get a combined feature vector, FV, such as:

$$FV = FV_p + FV_d \tag{3.45}$$

The feature vectors (FVs) of all of the images are constructed and stored to create a feature database using Algorithms 9. The feature vector of the user query is also constructed in the same way and compared with the feature vectors of the database for similarity and the retrieval of relevant images. The block diagram of the proposed approach is shown in Fig. 3.14.

Algorithm 9 Feature Extraction Algorithm of Approach-9

Input: Input image ***Img_file***, Size of blocks ***L=8×8***, Size of image ***S=M×N***

Output: Feature vector ***fv***

1. Call Algorithm 4 to compute texture features using 8×8 sub-block method to get feature vector $\boldsymbol{fv_p}$
2. Call Algorithm 7 to compute texture features using 8×8 DCT block method to get feature vector $\boldsymbol{fv_d}$
3. Combine both feature vectors to get feature vector ***fv***

   3.1 $\boldsymbol{fv} = [\ \boldsymbol{fv_p} \quad \boldsymbol{fv_d}\ ]$
4. Return feature vector *fv*

   4.1 ***Return fv***

3.2.5 Analysis of the Experimental Combination of the Color and Texture Features in the Spatial and Frequency Domains (Approach-10)

As discussed in Approach-9 that the statistical texture features of the grayscale images in the spatial and frequency domains are combined to improve the performance of retrieval. In order to further improve the retrieval performance of CBIR, an approach, Approach-10 is proposed, in which the statistical color features in the spatial domain are combined with the statistical texture features of the grayscale images in the frequency domain. The color features which have been extracted in the spatial domain using the 8×8 block method and discussed in Approach-3. In the frequency domain, the statistical quantized histogram texture features are extracted using the 8×8 DCT blocks, which has been discussed in Approach-7.

In this proposed Approach-10, the statistical color features in the spatial domain are combined with the texture features in frequency domain. It has been shown in Approach-3 that the 8×8 sub-block method is not only effective in the retrieval with but it is also efficient in the computation of the color features extraction. First, the color feature vector of the features is constructed using the 8×8 sub-blocks of the grayscale image. Then the quantized histogram texture feature vector is created using the 8×8 DCT blocks of the same image. Then, these two feature vectors are combined to get a net feature vector of both domains to retrieve similar images.

3.2.5.1 Preprocessing

The proposed approach starts with the conversion of the input RGB color image into grayscale image to reduce the computational cost as shown in Fig. 3.15 and it has been discussed in section 3.1.1.1.1.

3.2.5.2 Feature Extraction

The color and texture features are extracted from the grayscale image. In order to construct the feature vectors, the features are extracted in two steps. In the first step, the color features are extracted in the spatial domain using sub-block method and in the second step the texture features are extracted in the frequency domain using the quantized histograms of the DCT blocks as shown in Fig. 3.15.

3.2.5.2.1 Color Feature Extraction using Block Method in the Spatial Domain

The color feature extraction in the spatial domain using the 8×8 sub-block method has been discussed in section 3.1.3.3 of Approach-3. The grayscale image is divided into 8×8 non-overlapping 64 sub-blocks. Each block is in a 2-dimensional matrix of 0 to 256 values. The statistical color features first order moment, mean, and second order moment, standard deviation, are calculated by using the probability distribution of the pixel values in all of the blocks. After the calculation of these color features, the color feature vector, FV_c, is constructed as discussed in section 3.1.3.3 and as shown in Fig. 3.15.

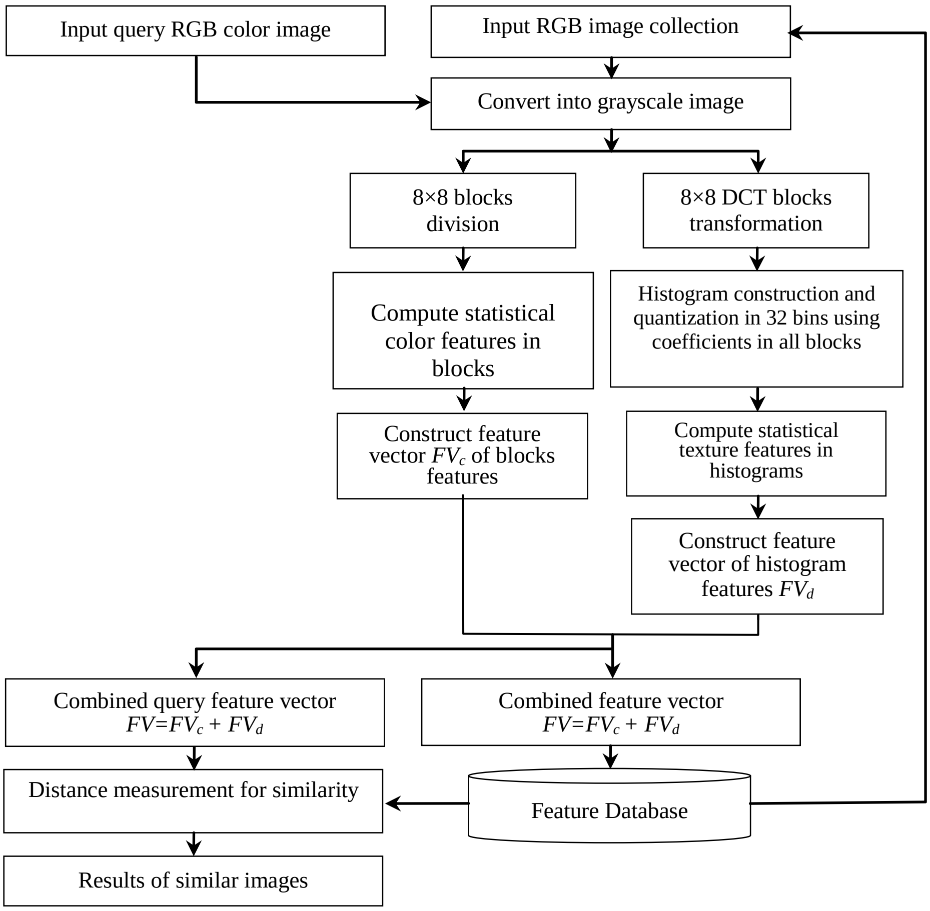


Figure 3.15 Block diagram of the proposed Approach-10 based on the combination of the color and texture features in the spatial and frequency domains.

3.2.5.2.2 Texture Features Extraction using Histogram Method in the Frequency Domain

The statistical quantized histogram texture features using 32 bins quantization scheme are extracted from the 8×8 DCT blocks of the grayscale image using the probability distribution of the coefficients of the blocks in the frequency domain, which have been discussed in section 3.2.1.2 to section 3.2.1.4 to construct the feature vector, FV_d.

After calculation of the color feature vector FV_c, in the spatial domain and the texture feature FV_d, in the frequency domain, are combined to get a combined feature vector FV, such as:

$$FV = FV_c + FV_d \tag{3.46}$$

The feature vectors (*FVs*) of all of the images are constructed and stored to create a feature database using Algorithms 10. The feature vector of the user query is also constructed in the same way and compared with the feature vectors of the database for the similarity and retrieval of relevant images. The block diagram of the proposed approach is shown in Fig. 3.15. Algorithm 10 is used to extract the features.

Algorithm 10 Feature Extraction Algorithm of Approach-10

Input: Input image ***Img_file***, Number of blocks $\boldsymbol{L=8\times8}$, Size of image $\boldsymbol{S=M\times N}$

Output: Feature vector $\boldsymbol{fv}$

1. Call Algorithm 3 to compute the color features using 8×8 sub-block method to get a feature vector $\boldsymbol{fv_c}$
2. Call Algorithm 7 to compute the texture features of histograms using 8×8 DCT blocks to get a feature vector $\boldsymbol{fv_d}$
3. Combine both feature vectors to get a feature vector $\boldsymbol{fv}$

   3.1 $\boldsymbol{fv} = [\ \boldsymbol{fv_c}\quad \boldsymbol{fv_d}\ ]$
4. Return feature vector $\boldsymbol{fv}$

   4.1 *Return* $\boldsymbol{fv}$

3.3 Similarity Measurement

Once the feature database of the images is created with the feature vectors, the user can give an example image as a query to retrieve the similar images from the database. The feature vector of the query image is also computed in the second step of the same approach.

To measure the similarity between the query image and the database images, the difference is calculated between the query feature vector and the database feature vectors by using the distance metric. A small difference between the two feature vectors indicates a large similarity and a small distance. The vectors of the images with a small distance are most similar to the query images. The distance metric, which has been included in this work, is the Euclidean distance. This distance metric is most commonly used for the similarity measurement in image retrieval because of its efficiency and effectiveness. It measures the distance between the two vectors of the images by calculating the square root of the sum of the squared absolute differences.

Let Q and T be the feature vectors of the query and target images, having ***n*** number of features such that $Q=\{q_1, q_2, \ldots, q_n\}$ and $T=\{t_1, t_2, \ldots, t_n\}$ where q_i and t_i are the features of the query and target images. The Euclidean distance D can be used to measure the distance between the Q and T feature vectors.

$$D(Q,T) = \sqrt{\sum_{i=1}^{n} \left(\left|Q_i - T_i\right|\right)^2} \tag{3.47}$$

Both images are the same for $D = 0$ and the small value of D shows the most relevant image to the query image.

3.4 Image Retrieval

For the retrieval of images, all the proposed approaches are performed in two steps.

- ***Step-1: Feature Database Creation***

In the first step, all of the images are acquired, one after another, from the collection of the images for the feature extraction. The extracted features are stored in a database in the form of feature vectors to create a feature database as shown in Fig. 3.1. The computation time of features database creation is noted for each proposed approach separately to get the computational cost of the feature database. The feature database is created once and it can be used as a source to provide similar images to any query image. The query image can be of the same database or any new image from any other source with any dimension. Algorithm 11 is used to create feature database.

Algorithm 11 Feature Database Creation Algorithm

Input: Image database with images, $I_{1,} \ldots, I_N$

Output: Feature database ***fv_DB*** creation

1. **For** each input image I_i **do**
2. Get image file from database
   2.1 *Img_file* = ***get_image***(I_i)
3. Extract features to create feature vector ***fv***
   3.1 *fv* = ***Feature_ Extraction*** (*Img_file*) //Call Feature Extraction Algorithm
4. Store the feature vector *fv* in a feature database ***fv_DB***
   4.1 *fv*_DB*(i)* =*fv*
5. End for

- ***Step-2: Retrieval by Query Image***

In the second step, the user browses the database images and selects any image as a query image to search and retrieve relevant images from the feature database by using the same algorithm. The feature vector of the query image is constructed and compared with the feature vectors of the database by computing the similarities using the Euclidean distance metric. The relevant images are displayed to the user according to the query image as shown in Fig. 3.1. Algorithm 12 is used for the retrieval of relevant images.

Algorithm 12 Image Retrieval Algorithm

Input: A query image Q, target images, $T_{1,} \ldots, T_N$

Output: A sorted list of target images

1. Select a query image Q

1.1 *Img_file* = ***get_image***(*Q*);

2. Extract *n* number of features of *Q* to create feature vector ***fv***

    2.1 *Qfv* =***Feature_Extraction*** (Img_file) // Call F. E. Algorithm

3. For each target image T_i

4. Access the feature vector of target image T_i from feature database *fv*_DB

    4.1 *Tfv (i)* = *fv*_DB*(* T_i*)*

5. Calculate Euclidean distance

    5.1 $D(Q, T_i) = \sqrt{\sum_{j=1}^{n} \left(\left| Qfv_j - Tfv_j \right| \right)^2}$

6. End for

7. Rank all the target images according to $D(Q, T_i)$.

3.5 Chapter Summary

In this chapter, methodology of different approaches for the developing of CBIR, have been discussed by extracting the color and texture features in the spatial and frequency domains to improve the retrieval performance of images.

In the spatial domain, for the extraction of the color features, the color histogram, color histogram refinement method and sub-block methods are used; while for the extraction of texture features, the sub-block methods are used. These features are extracted by computing the pixel values of a two dimensional matrix of pixels of an image in the spatial domain. The color standard histogram technique for a grayscale Laplacian filtered sharpened image gives good performance, especially for the 32 bins quantization scheme. However the standard histogram has lack of spatial information. To get the spatial information, the color histogram refinement method is used. The performance of this approach is based on the analysis of the features of the median and Laplacian filtered images using different quantization schemes. But this approach has high computational cost. To reduce the computational cost and to improve further the performance of the CBIR, in another two approaches the image is divided into non-overlapping sub-blocks to extract the local statistical color and texture features in the sub-blocks of the grayscale image. The color and texture features are extracted jointly from the non-overlapping blocks of the grayscale image and the performance is analyzed on the basis of different block sizes. The approaches in spatial domain mainly have focused on the performances of the color and texture features individually and in a combination of both in terms of efficiency and effectiveness for the retrieval of similar images.

In the frequency domain, various approaches are proposed to extract statistical quantized histogram texture features to improve the performance of the image retrieval in terms of retrieval and computational costs in the frequency (DCT) domain based on the median, median with edge extraction method and Laplacian filters, distance metrics, combination of features, combination of texture features in both domains and combination of color features in the spatial and the texture features in the frequency domains. The main contribution in the frequency domain is the incremental efficiency of the proposed approaches by utilizing the significant energy of the coefficients of the DCT blocks. The distribution of the coefficients of the blocks is used statistically to compute the texture features in different combination in various approaches.

CHAPTER 4

RESULTS AND DISCUSSION

In this chapter the experimental results of the implementation of all the proposed designed approaches from Approach-1 to Approach-10 in the spatial and frequency domains which have been discussed in Chapter 3, are discussed and analyzed to show the improvement in the retrieval performance of CBIR to achieve the main objective of the this study of research.

4.1 Experimental Setup

All the proposed approaches have been implemented using MATLAB 7.1 software**,** computer with Intel (R) Core (TM)^2 Dua processor, CPU with speed of 2.20GHz and memory RAM (random Access memory) with the storage space of 2 GB.

The proposed CBIR algorithms are tested by using the Corel image database which is provided by (Wang *et al.*, 2001) and discussed in section 2.7.

Evaluation criteria for CBIR depend upon the benchmarking which is still an issue for the researchers. However, there are recommendations issued by the technical committee from the International Association for Pattern Recognition (IAPR) regarding benchmarking of image retrieval (An *et al.*, 2011). Guidelines based on the recommendations for implementation of CBIR algorithm are given by (Park *et al*., 2008):

1. Usage of the free available image database for researchers without any constraint.
2. All the database images are loaded in main memory for the retrieval evaluation to reduce the hardware dependency.
3. The total number of images for testing in the Corel database is 1000.

4. The format of the images is JPEG.
5. The database consists of ten different types of image categories.
6. From all the categories a set of about 30 evaluation image queries are used.
7. For each image query total 9 images are retrieved as thumbnails as a result.
8. The evaluating measurements are precision, recall and F-Score.

In this thesis the experimental setup is same for all the proposed approaches and using the above same guidelines. Each algorithm is performed in two steps as discussed in section 3.4. In the first step, the features database is created for all images using the proposed approach and in the second step the user can select any image from the browsing window of user interface. The targeted images are displayed to the user in windows having 9 images separately as thumbnails. Important thing is that the user can give any image as a query image whether from the same database or any new image from any other outside source with any dimension.

Precision, recall and F-Score measurements are computed to show the retrieval accuracy of the query image among the returned images. These measurements are calculated for the top 9 relevant images retrieved in response of a query example image (Park et al., 2008) and are displayed to the user as thumbnails.

The purpose of implementation of the proposed algorithms is to analyze the impact of the combination of different techniques by extracting of color and texture features in the spatial and frequency domains for the robust retrieval of images.

4.2 Results and Discussion of the Proposed Approaches in the Spatial Domain

In this section the results of all the proposed approaches in the spatial domain, discussed in Chapter 3, are discussed:

4.2.1 Analysis of the Quantized Color Histogram Features Based on the Laplacian Filter (Approch-1)

In Approach-1 the retrieval results of the CBIR by using the quantized histogram features are analyzed based on Laplacian filter and different quantization schemes.

4.2.1.1 Results and Discussion

In the experiments, the two steps; feature database creation and image retrieval, of the proposed approach are performed for all of the proposed quantization schemes of 4, 8, 16, 32, 64, and 128 bins separately, using the Corel dataset images. The performance is measured in terms of average precision, recall and F-Score for these images as shown in Tables 4.1 to 4.3.

The results in Table 4.1 show the precision of the all categories of images for the quantization schemes of the histograms with different numbers of bins, which includes 4, 8, 16, 32, 64 and 128 bins, using color features. The categories of dinosaurs, buses and people give good results especially for 32 bins and the overall average precision is 51%. Table 4.2 also shows good results in terms of the recall for roses, buses and food using histograms 32 bins and the overall average recall is 72%.

Table 4.1 Average precision of the query images from the entire image categories for different quantization schemes.

Categories	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**64 Bins**	**128 Bins**	**Average**
Dinosaurs	67	67	89	**89**	**87**	**83**	**80**
Buses	44	44	56	**67**	**51**	**49**	**52**
People	38	50	51	**60**	**50**	**48**	**50**
Beaches	44	44	56	59	55	50	**51**
Buildings	44	44	56	60	53	51	**51**
Roses	33	56	56	58	54	50	**51**
Horses	44	67	44	46	41	38	**47**
Elephants	44	33	56	59	50	45	**48**
Mountains	33	44	44	45	38	35	**40**
Foods	33	33	44	46	35	33	**37**
Average	42	48	55	59	51	48	51

Table 4.3 shows the F-Score of all of the image categories for the quantization of the histograms of the Laplacian sharpened image in different numbers of bins, which

includes 4, 8, 16, 32, 64 and 128 bins, using color features. The F-Score shows the overall performance of the proposed algorithm for the entire image categories using histograms. It can be seen from Table 4.3 and Fig. 4.1 that the performance in terms of the average F-Score is better for the categories: dinosaurs, buses and roses than other categories especially for 32 bins, and the overall average F-Score is 59%.

Table 4.2 Average recall of the query images from the entire image categories for different quantization schemes.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	64 Bins	128 Bins	Average
Roses	75	71	83	**83**	**80**	**76**	**78**
Buses	80	80	71	**75**	**70**	**68**	**74**
Foods	75	75	80	**75**	**77**	**65**	**75**
Mountains	60	80	80	80	79	75	**76**
Beaches	67	80	71	71	70	69	**71**
Buildings	67	80	71	71	69	64	**70**
Dinosaurs	75	60	73	80	76	74	**73**
Horses	57	67	80	80	78	72	**72**
Elephants	67	60	83	71	77	73	**72**
People	60	67	50	70	58	55	**60**
Average	68	72	74	76	73	69	72

Table 4.3 Average F-Score of the query images from the entire image categories for different quantization schemes.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	64 Bins	128 Bins	Average
Dinosaurs	71	69	86	86	83	79	**79**
Buses	57	57	63	71	59	57	**60**
Roses	50	60	62	67	61	55	**59**
Beaches	51	57	66	68	65	60	**61**
Buildings	53	57	63	65	60	59	**59**
Horses	44	66	63	64	61	56	**59**
Elephants	55	63	55	58	53	50	**56**
People	50	44	66	68	61	55	**57**
Mountains	44	51	58	55	51	47	**51**
Foods	43	44	47	56	44	41	**46**
Average	52	57	63	66	60	56	59

Figure 4.2 shows the average F-score for the different quantization schemes. The incremental F-Score from 52% to 66% for 4 to 32 bins quantization, which shows that dividing the enhanced sharpened image into more bins, gives more color information with a bright color image as compared with less numbers of bins. In Fig. 4.2 it can be

seen that the results in terms of average F-Score for the quantization schemes from 4 to 32 bins is incremental. As the quantization bins are increasing the retrieval accuracy is also increasing from 52% to 66% average F-score. We stop the quantization at 32 bins to locate the optimum quantization scheme because the accuracy is decreasing as the number of bins increasing from 64 to 128. It has been noticed that quantization schemes with large number of bins do not necessarily lead to a better accuracy and on another side these lead to a much less efficient search. Hence the selection of a quantization scheme must be taken after a careful study and analysis.

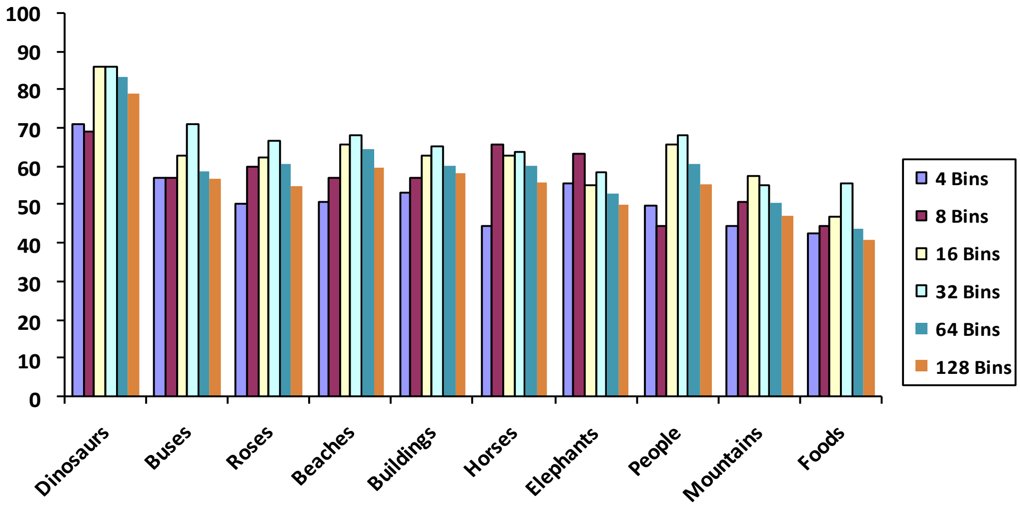


Figure 4.1 F-Score of the query images from the entire image categories in different quantization schemes.

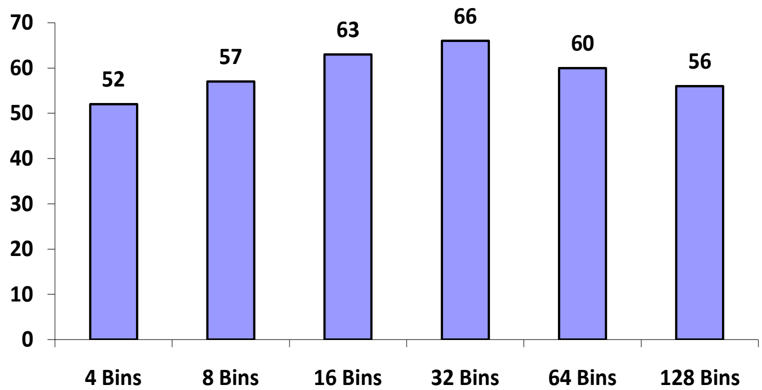


Figure 4.2 Average F-Score of the query images from the entire image categories for different quantization schemes.

The results show that the retrieval accuracy certainly affected by the selection of the proper quantization scheme after thorough experiments. The difference of the retrieval accuracy in terms of F-Score, between the optimal and maximum quantization schemes is decreasing. Accordingly using the quantization scheme with 256 bins is not recommended because it will give an inefficient retrieval of similar images. As a result it is distinguished that on one side using quantization scheme of bins with large number of values, describes the image with more detail information while on another side the matching distance between images is increasing which leads to a less effective image retrieval.

Clearly quantization in less number of bins reducing computational cost but also reducing the information concerning with the content of images. However histogram with large number of bins contains more information and give more discrimination power but it will increase the computational cost which leads to inefficient indexing of images with features to create features database. Quantization in large number of bins, histogram contains more bins with zero frequency of values. It is concluded that quantization in less and large number of bins have some disadvantages regarding retrieval. Then a question is raised that which number of quantization in bins will be optimal as concerned with effective retrieval of images. We cannot surely give the optimal quantization scheme depends upon the image type but we cay that the near to the optimal quantization scheme obtained using the sharpened Laplacian filtered image in our experimental work, is 32 bins.

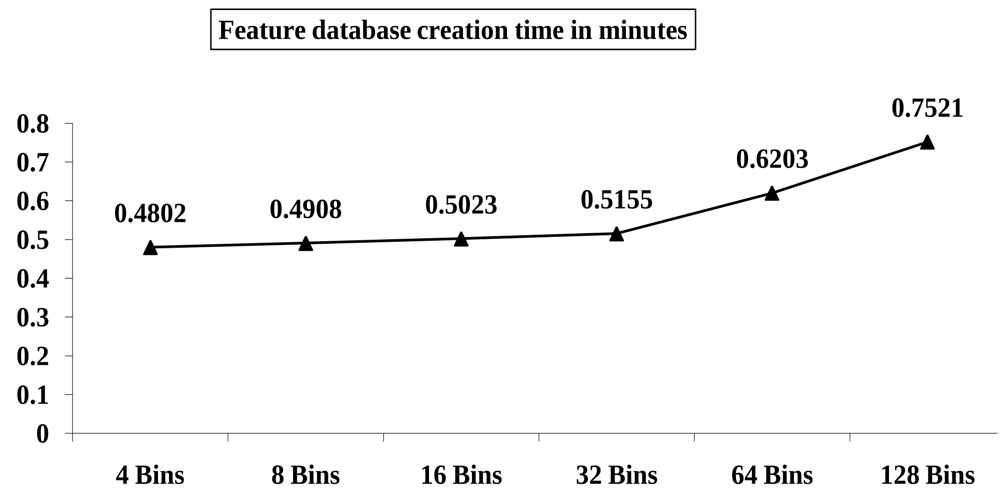


Figure 4.3 Time taken by the creation of feature databases for different quantization schemes.

Figure 4.3 shows the computational cost in terms processing time taken by the feature database creation. It is clear from the graph that as the number of bins in quantization scheme is increased the time also increased but the time difference is in seconds. It can be seen that the algorithm gives good results for the 32 bins quantization scheme in terms of retrieval and less effective results for more than 32 bins quantization scheme; however, the computational cost for the feature extraction to create the feature database using the 32 bin quantization, is higher than other bins with a slight difference in seconds.

Figure 4.4 (a-c) shows the results of the user queries. Each figure consists of a query image and the retrieved similar images from the database by using the proposed approach. The top single image is the query image and the below 9 are the relevant images. The results show that the proposed approach has good retrieval accuracy.

(a) (b) (c)

Figure 4.4 Query image results of (a) dinosaurs, (b) buses and (c) people using 32 bins quantization scheme in Approach-1.

4.2.1.2 Summary

The quantization of the histograms into bins has an impact on the performance of CBIR. In the proposed approach, the Laplacian sharpened grayscale image is used for feature extraction because the energy is compensated in the sharpened method, which is lost by the Laplacian filter in the preprocessing of the grayscale image to get a sharpened and enhanced image without noise. In the sharpening process using the Laplacian filter, not only the noise is reduced but the information is also preserved which gives a precise matching of images. The sharpened image is quantized into schemes with different number of bins like 4, 8, 16, 32, 64 and 128 bins. The statistical color features are extracted from the bins and represented in the feature vectors. These vectors are used in the similarity measurement for the retrieval of similar images. From the results, it is concluded that the quantization of the histograms into 32 bins gives the good performance in terms of F-Score in the retrieval of similar images as well as in the processing time for the creation of the feature database. After performing the experiments by using the images of all the categories as query images to get the results for the analysis and evaluation, proposed Approach-1 gives good performance in terms of effectiveness.

4.2.2 Color Feature Analysis for CBIR Based on Median, Median with Edge Extraction Method and Laplacian Filters using the Color Histogram Refinement Method (Approach-2)

In Approach-2 the retrieval results of the CBIR by using histogram refinement method to extract color features are analyzed based on different filters and different quantization schemes.

4.2.2.1 Results and Discussion

In the experiments, the two steps of the proposed algorithm are performed for three filter methods separately by using the quantization schemes of 4, 8, 16 and 32 bins, separately. The results are analyzed on the basis of three filter methods against different quantization schemes. The performance is measured in terms of the F-Score using the precision and recall for all of the images of the Corel database as shown in Tables 4.4 to 4.6.

The results in Table 4.4 show the average F-Score of the entire image categories against the quantization of the histograms in 4, 8, 16 and 32 bins using the median filter method by extracting color features. The F-Score shows the overall performance of the proposed algorithm for the query images of the entire categories. It can also be seen that the average F-Score is high for the categories of dinosaurs, horses and roses with 100%, 82% and 79% F-Scores, respectively. The 32 bins quantization gives good results as compared to other bins which is 70% average F-Score. The overall F-Score is 65%. This shows that the histogram refinement method with 32 bins gives good results using the color information of the median filtered images with good brightness and contrast.

Table 4.4 Average F-Score of the query images from the entire image categories for different quantization schemes using the median filter.

Categories	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**Average**
Dinosaurs	100	100	100	**100**	**100**
Horses	74	86	77	**90**	**82**
Roses	89	72	74	80	79
Elephants	61	64	70	71	66
Buses	53	61	66	69	62
Beaches	45	53	65	68	58
People	44	54	61	62	55
Buildings	47	55	67	50	55
Mountains	40	44	47	60	48
Foods	36	40	43	51	42
Average	59	63	**67**	**70**	65

Table 4.5 Average F-Score of the query images from the entire image categories for different quantization schemes using the median filter with edge extraction method.

Categories	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**Average**
Dinosaurs	97	100	100	**100**	**99**
Roses	73	82	89	**90**	**84**
Horses	68	78	79	**86**	78
Buses	58	68	67	74	67
Elephants	57	61	72	74	66
Buildings	53	55	68	51	57
People	41	59	54	61	54
Mountains	44	44	59	67	54
Beaches	49	52	51	56	52
Foods	41	42	51	60	49
Average	58	64	**69**	**72**	66

Table 4.5 shows the F-Score of the query images with different quantization schemes that includes 4, 8, 16 and 32 bins using the median with edge extraction filtered images. It can be seen from Table 4.5 that the performance in terms of the average F-Score for the categories of dinosaurs with 99%, roses with 84% and horses with 78%, is better than other categories, especially for 32 bins, and the overall

average F-Score is 66%. The average F-score for the different histogram bins is incremental from 58% to 72% for 4 to 32 bins quantization which shows that dividing the enhanced image into more bins give more color information by replacing the edge information of the median filtered image with a brighter color image as compared with a less number of bins.

Table 4.6 shows the average F-Score of the Laplacian sharpened images against the quantization of the histograms in different number of bins that includes 4, 8, 16 and 32 bins, using the color histogram features. It can be seen from Table 4.6 that the performance in terms of the average F-Score for the categories of dinosaurs with 100%, roses with 92% and horses with 90%, is better than the other categories, especially for 32 bins, and the overall average F-Score is 70%. The average F-score of the different quantization schemes is incremental from 64% to 75% for 4 to 32 bins quantizations which shows again that dividing the Laplacian sharpened and enhanced image into more bins gives more color information with a brighter color image as compared with a less number of bins.

Table 4.6 Average F-Score of the query images from the entire image categories for different quantization schemes using the Laplacian filter.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	Average
Dinosaurs	100	100	100	**100**	**100**
Roses	89	91	92	**95**	**92**
Horses	88	88	90	**94**	90
Elephants	59	75	75	79	72
Buses	53	68	69	77	67
Buildings	50	55	58	79	61
People	55	58	61	66	60
Beaches	51	57	60	59	57
Mountains	51	44	59	56	53
Foods	45	43	54	48	48
Average	64	68	**72**	**75**	70

Table 4.7 shows the comparison of the results of the proposed approach on the basis of the three filter methods: median, median with edge extraction and Laplacian filters, in terms of the average F-Score. The average F-Score based on the median, median with edge extraction and Laplacian filters are 65%, 66% and 70%. The overall

average F-Score is 67%. The F-Score using the Laplacian filter is 75% in the 32 bins histogram quantization and average is 70% which is better than other filter methods using quantization bins other than 32 bins. This shows that the Laplacian sharpened image has more energy for the retrieval of similar images.

Table 4.7 Average F-Score of the filters methods using different quantization schemes.

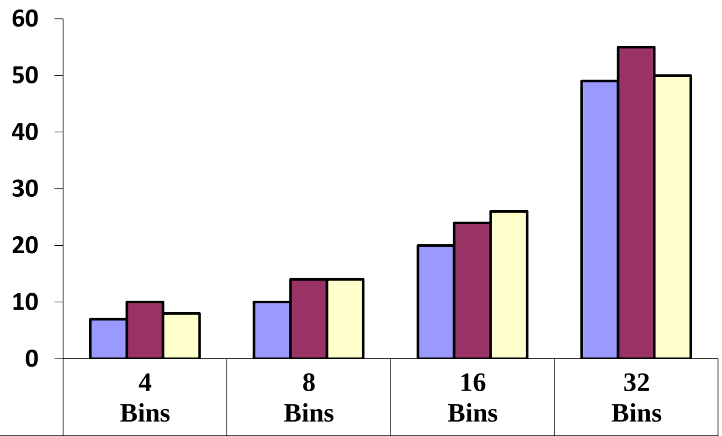
Filter Methods	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**F-Score Average**
Median filter	59	63	67	70	65
Median with Edge Extraction	60	64	69	72	66
Laplacian filter	64	68	72	**75**	**70**
Average	61	65	69	**72**	67

These filter methods provide enhanced images with more significant color information in the histograms; furthermore, color information plays an important role in the retrieval of the similar images. The color information can be extracted in the form of the statistical color features which provide the best performance in terms of retrieval, especially for the Laplacian filter as compared to other filter methods. The proposed approach has also provided improved performance for the median and median with edge extraction methods.

Figure 4.5 shows the computational cost for the creation of the feature database, in terms of processing time of the different filters in different quantization schemes. It can be seen that the time also increasing when the number of histogram quantization bins is increased. Since the 32 bins gives better performance in retrieval, it is considered the optimum quantization even though it takes more time in minutes than other bins like 4, 8 and 16 bins.

4.2.2.2 Summary

The performance of the Approach-2 is based on the different filter methods with different quantization schemes using the color histogram refinement method for feature extraction. Median, median with edge extraction and Laplacian filter methods are applied on grayscale images for noise removal before applying the histogram method. During the median filtration, edge information is lost which is restored by the



	4 Bins	8 Bins	16 Bins	32 Bins
Median filter	7	10	20	49
Median with Edge Extraction	10	14	24	55
Laplacian filter	8	14	26	50

Figure 4.5 Feature database creation time in minutes of the filter methods using different quantization schemes.

edge detection method while in the Laplacian method; some amount of information is also lost which is restored by subtracting the Laplacian image from the grayscale image to get a more enhanced sharpened image. The statistical features of mean and standard deviation of the quantized histograms are calculated using the spatial information of the connected regions. These statistical features are used for the retrieval of the relevant images. These features do not depend upon the orientation of the image. The performance is analyzed on the basis of the three filter methods using the spatial information of the histograms by quantization in different numbers of bins. The results show that the algorithm provides good results based on the Laplacian filter with 75% average F-Score.

4.2.3 Features Analysis for CBIR Based on the Color Moments using the Block Methods (Approach-3)

In Approach-3 the retrieval results of the CBIR by using sub-blocks of grayscale image to extract color features by computing color moments, are analyzed based on sub-blocks of different sizes.

4.2.3.1 Results and discussion

In the experiments, the two steps of the algorithm are performed for all of the proposed sub-blocks of different sizes: Whole-Image-as-One-Block, 2-Blocks-Column-Wise, 2-Blocks-Row-Wise, and 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks. Query images from the entire image categories of the Corel database are used and the performance of the proposed approach is measured by calculating the average precision, recall and F-Score for all of the proposed sub-block methods. The results are shown in Tables 4.8 to 4.10.

Table 4.8 shows the average precision in percentage for the query images of all of the categories against 9 proposed different sub-block methods by calculating two color moments; these are the mean and the standard deviation. The results in terms of precision of the proposed approach are good for all of the query images of categories, especially for dinosaurs, roses and elephants. These three categories provide good local color information in sub-blocks especially using 4×4, 8×8 and 16×16 block methods because these sub-block methods give better results of 62%, 66% and 63% precision as compared to the other sub-block methods.

Table 4.8 Average precision of the query images from the entire image categories for different sub-block methods.

Categories	**Whole Image as One-Block**	**2 Blocks Columns Wise**	**2-Blocks Row Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	**Average**
Dinosaurs	100	100	100	100	100	100	100	100	100	**100**
Roses	100	79	89	89	89	91	100	100	100	**93**
Elephants	33	56	78	67	78	78	78	78	78	**69**
Horses	44	50	43	54	58	70	66	60	54	**55**
Buses	44	67	44	56	47	67	56	44	56	**53**
People	56	56	56	56	56	56	44	44	56	**53**
Buildings	56	44	56	44	45	56	56	56	44	**51**
Beaches	56	56	44	44	56	44	44	44	44	**48**
Mountains	44	41	33	33	44	56	44	44	44	**43**
Foods	33	33	33	33	44	44	44	44	33	**38**
Average	**57**	**58**	**58**	**58**	**62**	**66**	**63**	**61**	**61**	**60**

Table 4.9 shows the average recall in percentage for all of the categories against 9 different block methods by using statistical color features, the mean and the standard

deviation. The proposed approach gives good results in terms of recall for all of the query image's categories and the optimum results is obtained for dinosaurs, buildings and beaches. The results in terms of recall for the block methods are also good and the optimal for the 4×4, 8×8 and 16×16 methods.

Table 4.9 Average recall of the query images from the entire image categories for different sub-block methods.

Categories	**Whole Image as One-Block**	**2 Blocks Columns Wise**	**2-Blocks Row Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	**Average**
Dinosaurs	100	100	100	100	100	**100**	100	100	100	**100**
Buildings	100	67	83	67	67	**71**	71	83	80	**77**
Beaches	71	71	67	67	83	**80**	80	80	77	**75**
Buses	57	76	77	83	67	100	83	57	71	75
Mountains	80	75	60	65	80	83	67	80	75	74
Foods	60	75	75	60	80	80	80	80	75	74
Roses	90	57	77	79	78	62	68	65	64	71
People	56	71	71	71	71	71	80	67	75	70
Horses	50	75	60	80	67	84	80	67	62	69
Elephants	50	64	64	65	64	70	70	70	70	65
Average	71	73	73	74	**76**	**80**	**78**	75	75	75

Table 4.10 shows the F-Score of all of the categories of images against 9 different sub-block methods used for the extraction of color features from a grayscale image. The F-Score shows the overall performance of the proposed approach for the query images of all of the categories using the sub-block methods. It can be seen from Table 4.10 that the performance in terms of the average F-Score is better for the categories: dinosaurs, roses and elephants with 100%, 80% and 67%, respectively.

It can be seen that the different block methods have been used for image retrieval to obtain the optimum method among them. After extensive experimental results and analysis, it has been concluded that the three block methods 4×4, 8×8 and 16×16, are the optimum methods with relatively good results of 67%, 71% and 68%average F-Score as compared to the other methods. These three methods are in the middle while the performance in terms of the retrieval of the first four i.e. Whole-Image-as-One-Block, 2-Blocks-Column-Wise, 2-Blocks-Row-Wise, 2×2 and the last two methods 32×32, and 64×64 are relatively low. The optimum method is 8×8 blocks which shows that that this method has good potential to provide local color information for the retrieval of images in the spatial domain for CBIR. The overall average F-Score is 65% which shows good and improved performance in terms of the retrieval of the images using the color moments of the grayscale images by dividing them into different block sizes.

Table 4.10 Average F-Score of the query images from the entire image categories for different sub-block methods.

Categories	**Whole Image as One-Block**	**2 Blocks Columns Wise**	**2-Blocks Row Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	**Average**
Dinosaurs	100	100	100	100	100	**100**	100	100	100	**100**
Roses	95	66	83	84	83	**74**	81	79	78	**80**
Elephants	40	60	70	66	70	**74**	74	74	74	**67**
Buses	50	71	56	67	55	80	67	50	63	62
Horses	47	60	50	64	62	76	72	63	58	61
Buildings	72	53	67	53	54	63	63	67	57	61
People	56	63	63	63	63	63	57	53	64	60
Beaches	63	63	53	53	67	57	57	57	56	58
Mountains	57	53	43	44	57	67	53	57	55	54
Foods	43	46	46	43	57	57	57	57	46	50
Average	62	63	63	64	**67**	**71**	**68**	66	65	65

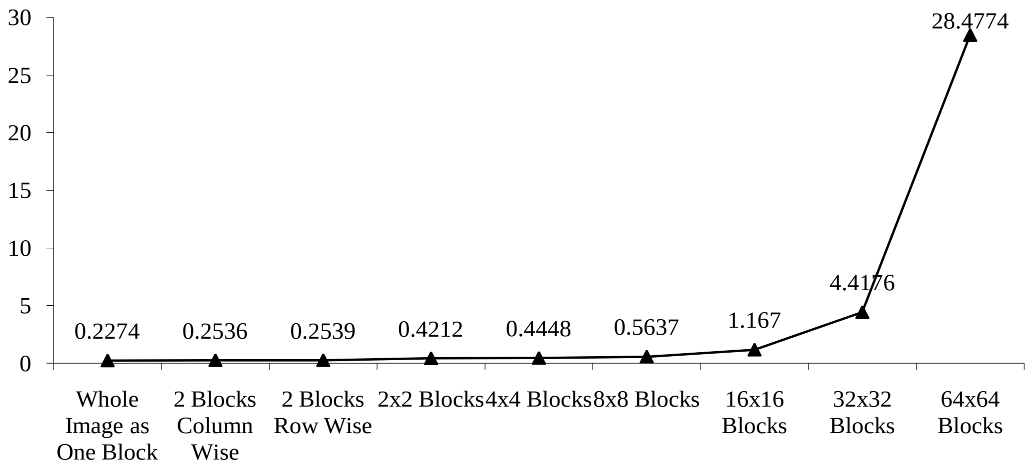
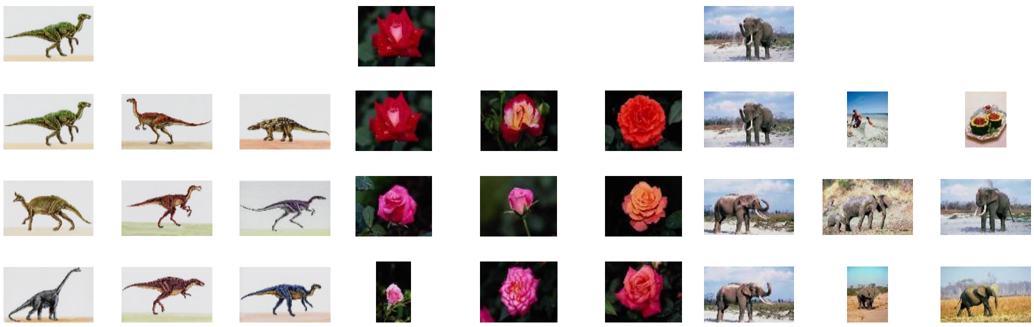


Figure 4.6 Feature database creation time in minutes of the different sub-block methods using the entire image categories.

Figure 4.6 shows the computational cost in terms of time taken by 9 different block methods to create a feature database for all of the categories of the images. It can be seen in Fig. 4.6 that the three block methods 4×4, 8×8, and 16×16, are again in the middle on the basis of the computational cost for the creation of a feature database. These three methods have more computational cost than the first four, i.e., the Whole-Image-as-One-Block, 2-Blocks-Column-Wise, 2-Blocks-Row-Wise, 2×2 but less than the last two methods, 32×32, and 64×64. It is concluded that the computational cost increased as the number of blocks increased. Since the retrieval performance of the 4×4, 8×8, and 16×16 block methods is high, hence they are considered the optimum methods while their computational cost is relatively high. Thus, the performance of our proposed approach is not only relatively efficient in computations of feature extraction but also gives good accuracy in terms of F-Score.

Figure 4.7 (a-c) shows the results of the user queries. Each figure consists of a query image and the retrieved images from the database. The results show that the proposed approach has good retrieval performance.



(a) (b) (c)

Figure 4.7 Query image results of (a) dinosaurs, (b) roses and (c) elephants using 8×8 sub-block method for extraction of color features.

4.2.3.2 Summary

In this proposed Approach-3 which has been based on the statistical color moments and these moments are extracted from the sub-blocks of different sizes of the images. It has been shown that the statistical color features have good retrieval performance using the color information of the local blocks in the images. The grayscale image is used for the feature extraction to reduce the computational cost and increase efficiency. The grayscale image is divided into blocks of different sizes to calculate the local statistical features of the mean and standard deviation of the pixels in each block. In this approach, 9 different block methods have been used. The color moments have been extracted in all of the methods and analyzed for their individual retrieval performance in terms of accuracy. The experimental results have been shown that the proposed approach is efficient in the feature extraction for different block methods and gives the improve performance in terms of accuracy, especially for the 8×8 and 16×16 block methods. It has been shown that the proposed approach is not only efficient in the feature extraction but also gives good accuracy in terms of retrieval.

4.2.4 Features Analysis for CBIR Based on the Statistical Texture Features using the Block Methods (Approach-4)

In Approach-4 the retrieval results of the CBIR by using sub-blocks of grayscale image to extract texture features, are analyzed based on sub-blocks of different sizes.

4.2.4.1 Results and Discussion

In the experiments, the two steps are performed for all of the proposed sub-block methods, Whole-Image-as-One-Block, 2-Blocks-Column-Wise, 2-Blocks-Row-Wise, 2×2, 4×4, 8×8, 16×16, 32×32, and 64×64 blocks, separately. Query images from the entire image categories of the Corel database are used and the performance of the proposed approach is measured by calculating the average precision, recall and F-Score for all of the proposed sub-block methods. The results are shown in Tables 4.11 to 4.13.

Table 4.11 shows the average precision in the percentage for the query images of the entire image categories against 9 different sub-block methods by calculating the statistical texture features. The results in terms of precision of the proposed approach are improved for the image categories, especially for dinosaurs, roses and elephants. These three categories provide local texture information in the blocks especially using the 4×4, 8×8 and 16×16 block methods because these block methods give better results of 67%, 70% and 68% precision as compared to the other block methods.

Table 4.12 shows the average recall in percentage for the query images from the image categories against 9 different block methods by using the statistical texture features. The proposed approach gives a good result in terms of recall for all of the

Table 4.11 Average precision of the query images from the entire image categories for different sub-block methods using the texture features.

Categories	Whole-Image-as-One-Block	2-Blocks-Columns-Wise	2-Blocks-Row-Wise	2×2 Blocks	4×4 Blocks	8×8 Blocks	16×16 Blocks	32×32 Blocks	64×64 Blocks	Average
Dinosaurs	100	100	100	100	100	100	100	100	100	**100**
Roses	87	85	89	100	100	100	100	100	100	**96**
Elephants	70	78	78	80	82	88	79	80	78	**79**
Buses	52	56	59	64	65	66	66	60	61	**61**
Horses	52	58	59	60	63	70	71	59	51	**60**
Buildings	46	47	56	54	57	60	58	50	42	**52**
People	45	40	60	56	44	65	56	44	43	**50**
Mountains	40	45	47	48	55	55	54	55	52	**50**
Beaches	43	41	55	56	56	47	44	48	40	**48**
Foods	33	44	42	43	44	52	56	50	62	**47**
Average	**57**	**59**	**65**	**66**	**67**	**70**	**68**	**65**	**63**	**64**

Table 4.12 Average recall of the query images from the entire image categories for different sub-block methods using the texture features.

Categories	Whole-Image-as-One-Block	2-Blocks-Columns-Wise	2-Blocks-Row-Wise	2×2 Blocks	4×4 Blocks	8×8 Blocks	16×16 Blocks	32×32 Blocks	64×64 Blocks	Average
Dinosaurs	100	100	100	100	100	100	100	100	100	**100**
Roses	80	57	60	57	95	99	100	100	96	**83**
People	67	57	83	86	72	86	69	67	58	**72**
Beaches	63	67	83	86	70	67	60	80	60	**71**
Buildings	62	65	67	71	65	80	70	80	72	**70**
Foods	38	100	60	76	60	88	60	60	70	**68**
Horses	55	57	83	67	71	71	75	55	62	**66**
Elephants	64	64	58	64	64	73	73	64	64	**65**
Buses	66	68	67	57	70	55	68	57	69	**64**
Mountains	40	67	67	71	62	66	62	80	61	**64**
Average	**64**	**70**	**73**	**74**	**73**	**79**	**74**	**74**	**71**	72

categories and is good for dinosaurs, roses and people. The results in terms of recall are good for the block methods as well as for the 4×4, 8×8 and 16×16 block methods.

Table 4.13 shows the F-Score of the entire image categories against 9 different block methods used for the extraction of texture features from grayscale images. The F-Score shows the overall performance of the proposed approach for all of the images using the block methods. It can be seen from Table 4.13 that the performance in terms of average F-Score is better for the categories dinosaurs, roses and elephants with 100%, 88% and 72%, respectively.

Table 4.13 Average F-Score of the query images from the entire image categories for different block methods using the texture features.

Categories	**Whole-Image-as-One-Block**	**2-Blocks-Columns-Wise**	**2-Blocks-Row-Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	**Average**
Dinosaurs	100	100	100	100	100	100	100	100	100	**100**
Roses	83	68	72	73	97	99	100	100	98	**88**
Elephants	67	70	67	71	72	80	76	71	70	**72**
Horses	53	57	69	63	67	70	73	57	56	**63**
Buses	58	61	63	60	67	60	67	58	65	**62**
Buildings	53	55	61	61	61	69	63	62	53	**60**
People	54	47	70	68	55	74	62	53	49	**59**
Beaches	51	51	66	68	62	55	51	60	48	**57**
Mountains	40	54	55	57	58	60	58	65	56	**56**
Foods	35	61	49	55	51	65	58	55	66	**55**
Average	**59**	**62**	**67**	**68**	**69**	**73**	**71**	**68**	**66**	**67**

It can be seen in Table 4.13 that the three block methods, 4×4, 8×8 and 16×16, give relatively good results with the average F-Score of 69%, 73% and 71% as compared to the other methods. It shows that these methods have a good potential to provide local statistical texture information for retrieval of images in CBIR. The overall average F-Score is 67% which shows good and improved performance in terms of the retrieval of images using the texture features of the grayscale images by dividing them into different block sizes.

Figure 4.8 shows the computational cost in terms of time taken by 9 different block methods for the creation of a feature database using query images of the categories. It can be seen in Fig. 4.8 that the three block methods, 4×4, 8×8 and 16×16, are in the middle on the basis of the computational cost for the creation of a feature database. These three methods have more computational cost than the first four, i.e., Whole-Image-as-One-Block, 2-Blocks-Column-Wise, 2-Blocks-Row-Wise and 2×2 while the computational cost is less than the last two methods, 32×32 and 64×64. It has been concluded that the computational cost increases as the number of blocks increases. Since the retrieval performance of the 4×4, 8×8 and 16×16 block methods is high, they are considered the optimum methods; moreover, their computational cost is relatively low.

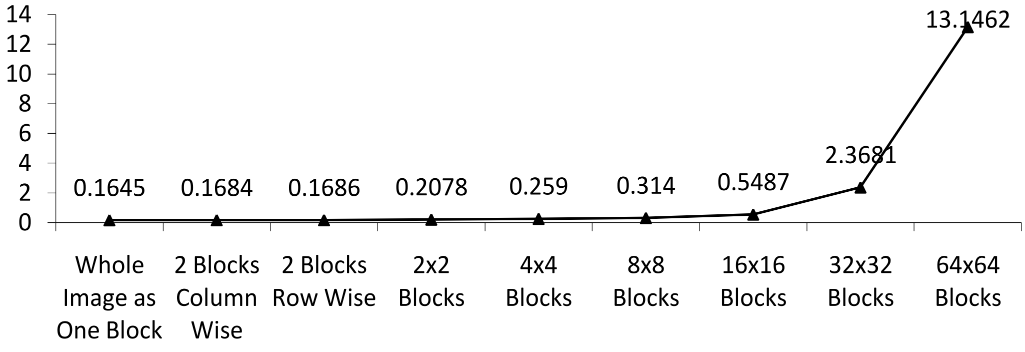
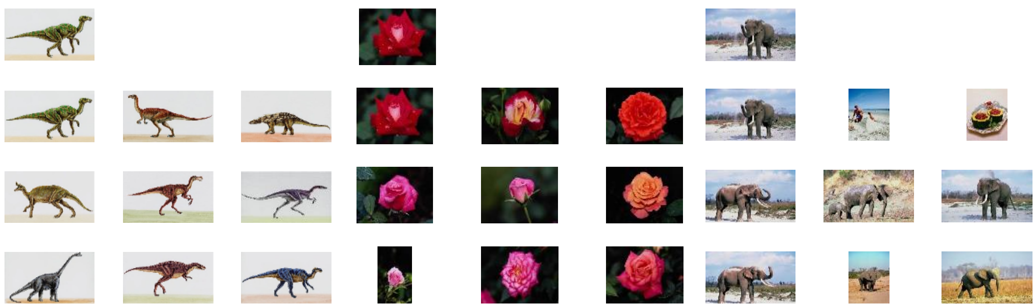


Figure 4.8 Feature database creation time in minutes of the different sub-block methods using the texture features in Approach-4.

Thus, the performance of the proposed Approach-4 is not only efficient in computations of texture feature extraction but it also gives good accuracy in terms of precision, recall and F-Score.

Figure 4.9 (a-c) shows the results of the user queries. The results show that the proposed algorithm has good retrieval accuracy.



(a) (b) (c)

Figure 4.9 Query image results of (a) dinosaurs, (b) roses and (c) elephants using texture features of 8×8 sub-block method in Approach-4.

4.2.4.2 Summary

In the proposed Apparoch-4 the statistical texture features are extracted from blocks of different sizes of images. It has been shown that the statistical texture features has good retrieval performance. The grayscale image is used for feature extraction to reduce the computational cost and increase the efficiency. The grayscale image is divided into sub-blocks of different sizes to calculate the local statistical features of mean, standard deviation, skewness, flatness, energy, entropy and smoothness of the pixels in each block. In this approach, 9 different block methods have been used. The statistical texture features have been extracted in all of the methods and their individual retrieval performance has been analyzed in terms of accuracy. In the experiment, the results show that the proposed approach is efficient in feature extraction for different sub-block methods and give good performance in terms of accuracy, especially for the 8×8 and 16×16 block methods. It has been shown that proposed approach is not only efficient in feature extraction but also gives good accuracy in terms of retrieval.

4.2.5 Combination of the Color and Texture Features for CBIR using the Blocks Methods (Approach-5)

In Approach-5 the color and texture features of images are combined for the retrieval of similar images using sub-block methods. The block methods have been used individually for the extraction of color and texture in Approach-3 and Approach-4 with proper analysis of the performance in terms of retrieval using 9 different block methods, separately. In Approach-3, the statistical color features are extracted from blocks of an image. After analysis of the results, it is shown that the 4×4, 8×8 and 16×16 block methods gives good and improved results with a 67%, 71% and 68% accurate retrieval, respectively. To further improve the retrieval performance for CBIR, the statistical texture features of mean, standard deviation, skewness, flatness, energy, entropy and smoothness, are extracted from the blocks of images. It has been shown in the Approach-4, that the statistical texture features have good retrieval performance using the texture information of the local blocks in the images. In the Approach-4, 9 different block methods have been used. The texture features have been extracted in all of the methods and an analysis of their individual retrieval performance in terms of accuracy has been performed. For improvement in retrieval, it has been shown in Approach-4 again, that the 4×4, 8×8 and 16×16 block methods gives better and improved results with a 69%, 73% and 71% accurate retrieval, respectively. The result of the 8×8 method is improved from 71% to 73%.

Now, the Approach-5 in this section is an attempt to combine the color and texture features using the optimum 8×8 sub-block method with efficient retrieval and low computational cost, along with other sub-block methods of different sizes. The statistical color moments of the mean and standard deviation, and the texture features of mean, standard deviation, skewness, flatness, energy, entropy and smoothness, are calculated in non-overlapping sub-blocks of the grayscale image. For feature extraction, the image is divided into non-overlapping sub-blocks of different sizes like 2×2, 4×4, 8×8 etc. The statistical texture features are calculated by using the intensity level distribution in each block of the image. A feature vector is constructed by combining the local color and texture features to retrieve the similar images. After extensive experiments, the results are analyzed in terms of precision, recall and F-Score measurements for all of the proposed block methods.

4.2.5.1 Results and Discussion

In the experiments, two steps for image retrieval are performed for all of the proposed 9 block methods to extract the color and texture features, create a feature database and retrieve similar images. The performance of the proposed approach is measured by calculating the average precision, recall and F-Score for all of the proposed block methods. The results are shown in Tables 4.14 to 4.16.

Table 4.14 shows the average precision in percentage for all of the categories against the 9 different block methods by calculating and combining the color and texture features. The result in terms of the precision of the proposed approach is good for query images of all of the categories, especially for dinosaurs, roses and elephants. These three categories provide good local color and texture information in the blocks, especially using the 4×4, 8×8 and 16×16 block methods, because these block methods give better results, with 77%, 82% and 79% precision, than the other block methods.

Table 4.15 shows the average recall in percentage for all of the categories against the 9 different block methods by using the combination of the statistical color and texture features. The proposed approach, gives good results in terms of recall and the better are for dinosaurs, roses and horses. Results in terms of recall for the block methods are also good and the better are for the 4×4, 8×8 and 16×16 methods as compared to other methods.

Table 4.14 Average precision of the query images from the entire image categories for different sub-block methods using the combination of color and texture features.

Categories	**Whole-Image-as-One-Block**	**2-Blocks-Columns-Wise**	**2-Blocks-Row-Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	**Average**
Dinosaurs	100	100	100	100	100	100	100	100	100	100
Roses	92	89	100	100	100	100	100	100	100	98
Elephants	71	70	84	86	93	92	92	93	92	86
Horses	65	58	71	80	91	92	93	91	90	81
Beaches	59	51	61	59	77	82	89	87	87	72
Mountains	36	43	63	67	67	77	87	90	90	69
Buses	56	63	63	68	72	85	64	56	41	63
People	59	51	62	48	52	58	51	24	18	47
Foods	45	48	37	49	62	65	65	27	22	47
Buildings	46	40	49	56	54	70	44	29	26	46
Average	**63**	**61**	**69**	**71**	**77**	**82**	**79**	**70**	**67**	**71**

Table 4.15 Average recall of the query images from the entire image categories for different sub-block methods using the combination of color and texture features.

Categories	**Whole-Image-as-One-Block**	**2-Blocks-Columns-Wise**	**2-Blocks-Row-Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	Average
Dinosaurs	100	100	100	100	100	100	100	100	100	100
Roses	59	78	100	100	100	100	100	100	100	93
Horses	68	75	59	78	72	90	88	82	81	77
Elephants	63	85	64	70	80	82	69	80	79	75
Foods	70	85	65	79	76	83	77	65	69	74
Mountains	73	72	59	68	75	78	71	80	72	72
Buildings	77	77	73	67	69	70	77	62	72	72
People	68	61	72	66	73	72	78	76	70	71
Beaches	88	73	65	64	73	68	68	63	62	69
Buses	60	65	67	62	61	78	72	70	69	67
Average	**73**	**77**	**72**	**75**	**78**	**82**	**80**	**78**	**77**	77

Table 4.16 shows the F-Score of all of the categories of images against the 9 different block methods used for the extraction of the color and texture features from the grayscale image. The F-Score shows the overall performance of the proposed approach for all of the images using the block methods. It can be seen from Table

4.16 that the performance in terms of the average F-Score is better for the categories: dinosaurs, roses and elephants with 100%, 95% and 80%, respectively.

Table 4.16 Average F-Score of the query images from the entire image categories for different sub-block methods using the combination of color and texture features.

Categories	**Whole-Image-as-One-Block**	**2-Blocks-Columns-Wise**	**2-Blocks-Row-Wise**	**2×2 Blocks**	**4×4 Blocks**	**8×8 Blocks**	**16×16 Blocks**	**32×32 Blocks**	**64×64 Blocks**	**Average**
Dinosaurs	100	100	100	100	100	100	100	100	100	**100**
Roses	72	83	100	100	100	100	100	100	100	**95**
Elephants	67	77	73	77	86	87	79	86	85	**80**
Horses	67	66	65	79	80	91	91	86	85	**79**
Beaches	71	60	63	61	75	74	77	73	72	**70**
Mountains	48	54	61	67	71	77	78	85	80	**69**
Buses	58	64	65	65	66	81	68	62	51	**65**
Foods	55	61	47	60	68	73	70	38	34	**56**
Buildings	57	53	59	61	61	70	56	39	38	**55**
People	63	56	67	55	61	64	62	37	28	**55**
Average	**66**	**67**	**70**	**73**	**77**	**82**	**78**	**71**	**67**	**72**

It can be seen in Table 4.16 that the three block methods of 8×8, 16×16 and 4×4 methods give relatively good results with 82%, 78% and 77% average F-Score as compared to the other methods. It shows that these methods have good potential to provide local color and texture information for the retrieval of images in CBIR. The overall average F-Score is 72% which shows good and improved performance in terms of the retrieval of images using the combination of the color and texture features of the grayscale images by dividing them into the non-overlapping blocks of different sizes.

Figure 4.10 shows the computational cost in terms of time taken by the 9 different block methods to create a feature database for all of the categories of the database images. It can be seen that the time is also increased when the number of blocks is increasing. Since the 8×8 blocks method gives better performance in terms of retrieval, it is considered the optimum block method to provide texture features for image retrieval even though it takes more time in minutes than the other block methods like 2×2 and 4×4 while it takes less time than 16×16 and 32×32. Thus, the 8×8 blocks method not only gives good results in terms of retrieval but is also comparatively efficient in the extraction of features.

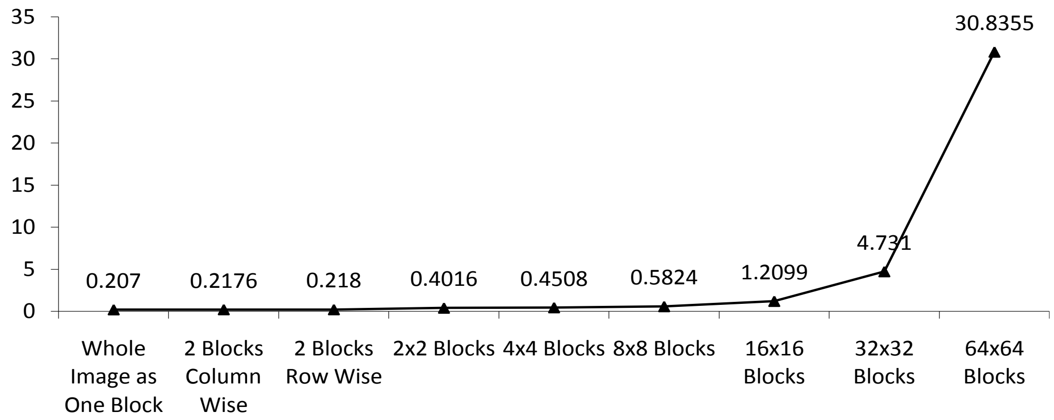


Figure 4.10 Feature database creation time in minutes of the different sub-block methods using the combination of color and texture features for the entire image categories.

Thus, the performance of the proposed algorithm is not only efficient in the computations of feature extraction but also gives good accuracy in terms of precision, recall and F-Score.

4.2.5.2 Summary

In this section an Approach-5 has been proposed which is based on the combination of the statistical color and texture features. These features are extracted from non-overlapping blocks of different sizes of the images. It has been shown in the proposed approach that the combination of the color and texture features has good retrieval performance using the combined color and texture information of the local blocks in the images. A grayscale image is used for the feature extraction to reduce the computational cost and increase the image retrieval efficiency. The grayscale image is divided into non-overlapping sub-blocks of different sizes. In all of the blocks, mean and standard deviation are calculated as color features by using the pixel values of the blocks. The statistical texture features, mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness are calculated by using the probability distribution of the intensity levels in the blocks. In this approach, 9 different block methods have been used. Both the color and texture features are extracted in all of the methods and analyzed for their individual retrieval performance in terms of accuracy. The results of the experiments show that the proposed approach is efficient in the features extraction for sub-block methods with different sizes and gives the optimal performance in terms of accuracy, especially for the 8×8 and 16×16 block methods.

4.3 Results of the Proposed Approaches in the Frequency Domain

In this section the results of all the proposed approaches in frequency domain in Chapter 3, are discussed:

4.3.1 Quantized Histogram Texture Features Based on Median, Median with Edge Extraction Method and Laplacian Filters (Approach-6)

In Approach-6 the retrieval results of the CBIR by extracting quantized histogram texture features, are analyzed based on different filters and different quantization schemes in the frequency (DCT) domain.

4.3.1.1 Results and Discussion

The proposed Approach-6 is performed in two steps as discussed in section 3.4. In the first step, the feature database is created by extracting the feature vectors. In the second step, the feature vector of the query image is created and compared with the database feature vector. In the experiments, the two steps of the proposed approach are performed for the three filter methods, separately by quantizing them into different histogram bins like 4, 8, 16 and 32 bins, separately. The results are analyzed on the basis of the three filter methods against the different quantization schemes. The performance is measured in terms of the F-Score using the precision and recall for the randomly selected query images of the Corel database as shown in Tables 4.17 to 4.19.

The results in Table 4.17 show the average F-Score of the 10 categories of images against the quantization schemes of 4, 8, 16 and 32 bins using the median filter method by extracting the texture features. The F-Score shows the overall performance of the proposed approach for the images of all of the categories using the texture features in the different histogram quantization bins of the DCT blocks. It can also be seen that the average F-Score is high for the categories of dinosaurs, people and horses with 100%, 94% and 73% F-Score, respectively. The 32 bins quantization scheme gives good results as compared to the other schemes. The overall average F-Score is 66%. This shows that the quantized histogram texture features with 32 bins gives good results for the median filtered images.

Table 4.17 Average F-Score of the query images from the entire image categories for different quantization schemes using the median filter.

Categories	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**Average**
Dinosaurs	100	100	100	**100**	**100**
People	77	100	100	**100**	**94**
Horses	44	80	84	**84**	73
Buses	51	72	78	81	70
Beaches	58	69	72	72	68
Buildings	62	67	67	67	66
Roses	53	56	55	56	55
Elephants	51	51	48	55	51
Mountains	53	45	39	45	46
Foods	45	21	34	43	36
Average	60	66	**68**	**70**	66

Table 4.18 shows the F-Score of the image categories for the quantization of histograms of the median with edge extraction filtered images in different quantization schemes that include 4, 8, 16 and 32 bins using the quantized histogram texture features. It can be seen from Table 4.18 that the performance in terms of average F-Score for the categories of dinosaurs with 100%, people with 99% and horses with 77%, is better than other categories, especially for 32 bins, and the overall average F-Score is 69%. The average F-score for the different quantization schemes is incremental from 64% to 73% which shows that dividing the enhanced image into more bins gives more texture information by replacing the edge information of the median filtered image as compared with a less number of bins.

Table 4.19 shows the average F-Score of the image categories of the Laplacian sharpened images against the different quantization schemes of 4, 8, 16 and 32 bins using the histogram texture features. It can be seen from Table 4.19 that the

Table 4.18 Average F-Score of the query images from the entire image categories for different quantization schemes using the median filter with edge extraction method.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	Average
Dinosaurs	100	100	100	**100**	**100**
People	96	100	100	**100**	**99**
Horses	45	88	84	**90**	**77**
Beaches	59	74	85	74	73
Buses	55	63	76	84	70
Buildings	69	70	66	61	66
Roses	57	57	60	64	60
Elephants	59	62	55	62	59
Mountains	57	48	47	49	50
Foods	48	23	48	43	40
Average	64	69	**72**	**73**	69

Table 4.19 Average F-Score of the query images from the entire image categories for different quantization schemes using the Laplacian filter.

Categories	4 Bins	8 Bins	16 Bins	32 Bins	Average
Dinosaurs	100	100	100	**100**	**100**
People	75	96	100	**100**	**93**
Horses	75	82	88	**85**	**83**
Beaches	62	90	92	83	82
Buses	70	74	75	78	74
Roses	66	64	74	79	71
Buildings	58	61	67	69	64
Elephants	63	44	48	60	54
Mountains	47	52	52	60	53
Foods	46	35	42	62	46
Average	66	70	**74**	**78**	72

performance in terms of the average F-Score for the categories of dinosaurs with 100%, people with 93% and horses with 83%, is better than other categories, especially for the 32 bins, and the overall average F-Score is 72%. The average F-Score for the different histogram bins is incremental from 66% to 78% for the 4 to 32 bin quantization which shows again, that dividing the Laplacian sharpened and enhanced image into more bins gives more texture information as compared with a less number of bins.

Table 4.20 shows the comparison of the results of the proposed approach on the basis of the three filter methods: median, median with edge extraction and Laplacian filters in terms of the average F-Score using all of the categories of the images of the Corel database for the texture histogram statistical features. The average F-Score is based on the median, median with edge extraction and Laplacian filters are 66%, 70% and 72%. The overall average F-Score is 69%. The F-Score using the Laplacian filter is 78% in the 32 bin histogram quantization and the average is 74%, which is better than other filter methods using the quantization bins other than 32 bins. This shows that the Laplacian sharpened image has more energy for the retrieval of similar images using the texture information of histograms in the DCT domain.

Table 4.20 Average F-Score of the filters method for different quantization schemes.

Filter Methods	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**Average**
Median filter	60	66	68	70	66
Median with Edge Extraction	64	69	72	73	70
Laplacian filter	66	70	74	**78**	**72**
Average	63	68	**71**	**74**	69

It can be seen in Table 4.20 that the optimal quantization scheme for all three filter methods is 32 bins. So by extracting texture features using distribution of values in 32 bins give robust retrieval to the images sharpened with the Laplacian filter.

These filter methods provide enhanced images with more significant information in the histograms. The texture information can be extracted in the form of the statistical texture features which provide the good and improved performance in terms of retrieval, especially for the Laplacian filter as compared to other filter methods. The proposed approach also provides improved performance for the median and median with edge extraction methods.

Table 4.21 show the computational cost of the approach in terms of the time taken by different numbers of quantization bins for the creation of the feature database using all of the categories of the images. It can be seen that the time is also increasing when the number of the histogram quantization bins is increasing and less time is taken by using the Laplacian filter. Since the 32 bins gives better performance in retrieval, it is considered to be the optimal quantization even though it takes more time in minutes than other bins like 4, 8 and 16.

Table 4.21 Feature database creation time in minutes of the filter methods for different quantization schemes.

Filter Methods	4 Bins	8 Bins	16 Bins	32 Bins	Average
Median with Edge Extraction	19.83	19.85	19.9	19.97	19.89
Median filter	16.22	16.24	16.29	16.31	16.27
Laplacian filter	16.00	16.17	16.22	**16.27**	**16.17**
Average	17.35	17.42	17.47	**17.52**	17.44

4.3.1.2 Summary

In this section, a CBIR Approach-6 has been proposed for the effective image retrieval in which the experimental comparison of the statistical texture features in terms of the accuracy of the image retrieval based on the median and Laplacian filters is performed in the DCT domain. Only the DC and the first three AC coefficients with more significant energy are selected in each DCT block to get the quantized histogram statistical texture features. These features are extracted from the median, median with edge extraction and Laplacian filtered images. The experimental comparison of the results of the three filter methods are analyzed for all of the image categories in terms of the accuracy of the image retrieval. It has been concluded that the enhanced and sharpened Laplacian filtered images using the quantized histogram texture features give good performance in terms of the F-Score in the DCT domain for the compressed images as compared to the retrieval of the images in the spatial domain.

4.3.2 Analysis of the Distance Metrics in CBIR using the Statistical Quantized Histogram Texture Features in the Frequency Domain (Approach-7)

In Approach-7 the retrieval results of the CBIR by extracting quantized histogram texture features, are analyzed based on the different distance metrics and different quantization schemes in the frequency (DCT) domain.

4.3.2.1 Results and Discussion

In the experiment, the two steps of the approach are performed for all of the quantization of the histograms into 4, 8, 16 and 32 bins, separately. For each quantization of the histograms, all the distance metric are used, separately, in the experiments to get the results in terms of precision and recall to calculate the F-Score. Consequently, each distance metric is tested in all of the quantization bins and the results are analyzed for all of the distance metrics against all of the quantization schemes. The results in precision, recall and F-score of all of the quantization schemes using only the Euclidean distance metric for the similarity measurement and all of the image categories, are shown in Tables 4.22 to 4.24.

Table 4.22 shows the average precision in percentage of the seven distance metrics in different histogram quantization bins using query images of all of the image categories for the matching of the query image with the target images of the database in searching for similar images. It can be seen that the dinosaurs, roses, people and horses give better results as compared to the other categories whereas using the Euclidean distance gives the optimal results for the quantization bins, especially for 16 and 32 bins, which have an average precision of 81% and 82%, as compared to the other distance metrics. The average precision is improved incrementally from the 4 bins quantization to 32 bins. The overall average precision is 70%, which shows good and improved retrieval.

Table 4.22 Average precision of the query images from the entire image categories for distance metrics using different quantization schemes.

Categories	Sum of Absolute Difference (SAD)				Sum of Squared of Absolute Differences (SSAD)				Euclidean distance				City block distance				Canberra distance				Maximum value distance				Minkowski distance (with p=3)				Average precision
	Bins				Bins				Bins				Bins				Bins				Bins				Bins				
	4	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	
People	73	75	85	100	80	80	85	88	100	100	100	92	83	87	90	92	74	87	91	96	85	96	97	100	92	94	97	100	**90**
Beaches	63	78	76	78	63	60	66	69	70	73	86	86	63	66	68	71	57	58	67	73	69	69	83	87	63	67	68	73	70
Buildings	42	46	50	69	50	55	56	57	63	65	66	68	56	57	62	65	50	53	60	56	50	56	58	65	42	48	55	62	56
Buses	36	46	48	58	59	74	79	93	78	79	86	89	68	78	81	85	69	77	85	89	53	58	83	86	45	48	50	74	70
Dinosaurs	84	96	96	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**99**
Elephants	38	57	60	65	46	65	69	71	74	76	76	80	67	70	75	80	63	68	70	75	42	51	63	71	37	39	41	55	62
Roses	66	76	79	85	78	83	92	100	86	90	96	99	78	84	98	100	68	75	89	98	74	76	83	93	47	50	52	78	**81**
Horses	58	60	65	69	78	94	95	98	85	89	91	92	68	94	96	97	63	71	83	86	60	84	90	92	43	46	47	65	77
Mountains	29	35	40	45	52	53	55	57	63	66	66	67	50	54	54	54	42	44	53	55	35	45	47	49	41	42	45	54	50
Foods	22	27	28	32	50	53	55	59	55	47	48	52	47	56	58	61	50	54	56	56	28	28	42	50	39	45	49	52	46
Average	51	60	63	70	66	72	75	79	77	78	81	**82**	68	75	79	**80**	64	69	75	78	60	66	75	79	55	58	60	68	70

Table 4.23 shows the average recall in percentage of all of the image categories in different histogram quantization bins using all of the distance metrics for the matching of the query image with the database images in searching for similar images. The results show that the performance in recall is also better for the dinosaurs, roses, people and horses as compared to the other categories. Moreover, using the Euclidean distance gives the best results for the quantization bins, especially for 16 and 32 bins, which are an average recall of 81% and 84% as compared to other distance metrics. The average recall is also improved incrementally from the 4 bins quantization to 32 bins.

Table 4.23 Average recall of the query images from the entire image categories for distance metrics using different quantization schemes.

Categories	Sum of Absolute Difference (SAD)				Sum of Squared of Absolute Differences (SSAD)				Euclidean distance				City block distance				Canberra distance				Maximum value distance				Minkowski distance (with p=3)				Average recall
	Bins				Bins				Bins				Bins				Bins				Bins				Bins				
	4	8	16	32	4	8	16	32	4	8	16	32	4	8	16	32	4	8	16	32	4	8	16	32	4	8	16	32	
People	77	82	82	93	73	75	76	78	86	92	95	98	72	74	78	80	71	74	75	78	92	95	96	100	83	85	89	95	**84**
Beaches	79	80	84	89	63	65	65	69	63	68	71	74	69	70	73	75	73	75	77	78	72	74	76	78	71	76	78	79	74
Buildings	51	56	57	82	71	73	75	79	64	65	67	70	68	71	72	75	69	73	71	63	56	60	63	68	51	63	68	73	67
Buses	62	65	58	68	72	74	76	80	74	75	76	78	73	75	80	83	77	77	78	82	66	64	64	67	48	60	65	69	71
Dinosaurs	93	94	96	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**99**
Elephants	50	53	54	65	69	72	73	75	68	69	70	71	66	69	72	74	71	74	75	78	67	69	70	70	61	76	78	77	69
Roses	80	81	82	86	81	84	86	89	82	85	89	94	81	83	92	95	78	82	87	91	79	81	82	87	66	71	73	76	**83**
Horses	72	76	78	83	80	83	84	85	81	84	84	92	79	81	81	83	77	83	85	86	76	69	81	83	73	76	78	79	**81**
Mountains	53	56	59	63	75	74	75	79	76	78	81	88	71	73	75	77	71	73	74	73	61	61	66	70	67	72	74	74	71
Foods	40	43	48	52	64	62	63	65	70	75	76	77	63	65	69	73	56	60	63	71	62	62	68	75	54	63	65	66	63
Average	66	69	70	78	75	76	77	80	76	79	**81**	**84**	75	76	80	**82**	75	78	79	80	73	74	77	80	68	75	77	79	76

Table 4.24 shows the average F-Score which describes the overall performance of the retrieval of the similar images category-wise in various histogram quantization bins using the Euclidean distance. The retrieval performance of the proposed method using the Euclidean distance for the categories of dinosaurs, roses, people and horses is better. It can be seen in Table 4.24 that the F-Score retrieval is increased from the 4 bins quantization towards the 32 bins for all of the distance metrics. The otimal performance in terms of the average F-Score is given by using the Euclidean distance for all of the quantization bins, especially for the 16 and 32 bin quantization having 81% and 83% average F-Scores which show that 32 bins provides more energy in the DCT blocks for retrieval.

Table 4.24 Average F-Score of the query images from the entire image categories for distance metrics using different quantization schemes.

Categories	Sum of Absolute Difference SAD				Sum of Squared of Absolute Differences SSAD				Euclidean distance				City Block Distance				Canberra Distance				Maximum value distance				Minkowski distance (with p=3)				Average F-Score
	Bins				Bins				Bins				Bins				Bins				Bins				Bins				
	4	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	**4**	**8**	**16**	**32**	
People	75	78	83	96	76	77	80	83	92	96	97	**95**	77	80	84	85	73	80	83	86	89	96	96	100	87	90	93	97	**87**
Beaches	70	79	80	83	63	62	65	69	66	71	78	**79**	66	68	71	73	64	66	72	76	71	72	80	82	67	71	73	76	72
Buildings	46	51	53	75	59	63	64	66	63	65	66	**69**	62	63	67	70	58	62	65	59	53	58	61	67	46	55	61	67	61
Buses	45	54	53	63	65	74	77	86	76	77	80	**83**	71	76	81	84	73	77	81	85	59	61	72	75	46	54	57	71	70
Dinosaurs	89	95	96	100	100	100	100	100	100	100	100	**100**	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**99**
Elephants	43	55	57	65	55	68	71	73	71	72	73	**76**	67	69	74	77	67	71	72	76	52	59	66	71	46	52	54	65	65
Roses	72	78	80	85	79	83	89	94	84	88	92	**97**	80	83	95	97	73	79	88	95	77	79	83	90	55	59	61	77	**82**
Horses	64	67	71	75	79	88	89	91	83	87	87	**92**	73	87	88	89	70	77	84	86	67	76	85	88	55	57	59	72	78
Mountains	37	43	48	53	62	62	64	66	69	72	73	**76**	59	62	63	63	53	55	62	63	45	52	55	58	51	53	56	63	58
Foods	29	33	35	40	56	57	59	62	62	58	59	**62**	54	60	64	66	53	57	59	62	38	38	52	60	45	53	56	58	53
Average	57	63	66	**73**	69	74	76	**79**	77	79	**81**	**83**	71	75	79	**80**	69	73	77	**79**	65	69	75	**79**	60	64	67	**75**	73

Table 4.25 shows the average precision of the proposed distance metrics which are used in the matching of the images to retrieve the similar images using different histogram quantization bins for the extraction of statistical texture features in the DCT blocks. It can be seen that the optimal retrieval average precision is 82% of the Euclidean distance using the 32 bin quantization. The City Block, Sum of Squared of Absolute Differences (SSAD) and Canberra distance also give good performance in terms of precision. The results show that using the top four distance metrics, as shown in Table 4.25, for matching of the images for retrieval of similar images, the optimal results are obtained, especially in the 32 bin quantization in the DCT domain.

Table 4.25 Overall average precision of the query images from the entire image categories for distance metrics using different quantization schemes.

Distance Metrics	4 Bins	8 Bins	16 Bins	32 Bins	Average
Euclidean distance	77	78	81	**82**	**80**
City Block Distance	68	75	79	**80**	76
Sum of Squared of Absolute Differences (SSAD)	66	72	75	79	73
Canberra Distance	64	69	75	78	72
Maximum value distance	60	66	75	79	70
Sum of Absolute Difference (SAD)	51	60	63	70	61
Minkowski distance (with p=3)	55	58	60	68	60
Average	63	68	73	**77**	70

Table 4.26 shows the average recall of the proposed distance metrics which are used in the matching of the images to retrieve the similar images using the different histogram quantization bins for the extraction of the statistical texture features in the DCT blocks. The best retrieval recall is 84% of the Euclidean distance using the 32

bins quantization. The City Block, Canberra Distance and Sum of Squared of Absolute Differences (SSAD) also give good performance in terms of recall. The overall average recall is 76%.

Table 4.26 Overall average recall of the query images from the entire image categories for distance metrics using different quantization schemes.

Distance Metrics	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**Average**
Euclidean distance	76	79	**81**	**84**	**80**
City Block Distance	75	76	80	**82**	78
Canberra Distance	75	78	79	80	78
Sum of Squared of Absolute Differences (SSAD)	75	76	77	80	77
Maximum value distance	73	74	77	80	76
Minkowski distance (with p=3)	68	75	77	79	75
Sum of Absolute Difference (SAD)	66	69	70	78	71
Average	73	75	77	**80**	76

Table 4.27 shows the average F-Score of the distance metrics using the quantized histogram texture features of the DCT blocks. The F-Score results show that by using histogram statistical texture features of different quantization bins, the Euclidean distance gives good retrieval performance, especially in 32 bins in the DCT domain. The City Block, Canberra Distance and Sum of Squared of Absolute Differences (SSAD) also give good performance in terms of retrieval. Table 4.27 shows that the 32 bin quantization provides good energy in the DC and in the first three AC coefficients in the DCT blocks in the frequency domain for the opimal retrieval of the JPEG images in terms of an average 73% F-Score.

Table 4.27 Overall average F-Score of the query images from the entire image categories for distance metrics using different quantization schemes.

Distance Metrics	**4 Bins**	**8 Bins**	**16 Bins**	**32 Bins**	**Average**
Euclidean distance	77	79	**81**	**83**	**80**
City Block Distance	71	75	79	**80**	76
Canberra Distance	69	73	77	79	75
Sum of Squared of Absolute Differences (SSAD)	69	74	76	79	75
Maximum value distance	65	69	75	79	72
Minkowski distance (with p=3)	60	64	67	75	67
Sum of Absolute Difference (SAD)	57	63	66	73	65
Average	67	71	74	**78**	73

Table 4.28 shows the average computation time taken by the proposed distance metrics for the matching of a query image with the database images to retrieve the

similar images. It can be seen in Fig. 4.11 that the Euclidean distance, City Block Distance and Sum of Absolute Difference (SAD) take less computational cost for the similarity measurement of the query image with the database images. The results show that the Euclidean distance is not only effective in retrieval but also efficient in the computations. Fig. 4.12 shows that the computational time taken by using the 32 bins quantization is slightly greater as compared to the other quantizations but for the retrieval performance it is better than other bins.

Table 4.28 Average computation time (in minutes) of the proposed distance metrics for matching of the query image with the database images.

Distance Metrics	4 Bins	8 Bins	16 Bins	32 Bins	Average
Euclidean distance	0.0183	0.0185	0.0186	**0.0188**	**0.0186**
City Block Distance	0.0187	0.0188	0.0190	0.0190	0.0189
Sum of Absolute Difference (SAD)	0.0187	0.0188	0.0190	0.0193	0.0190
Maximum value distance	0.0187	0.0188	0.0195	0.0192	0.0191
Sum of Squared of Absolute Differences (SSAD)	0.0189	0.0189	0.0192	0.0194	0.0191
Minkowski distance (with p=3)	0.0190	0.0192	0.0192	0.0193	0.0192
Canberra Distance	0.0192	0.0193	0.0194	0.0193	0.0193
Average	0.0188	0.0189	0.0191	**0.0192**	0.0190

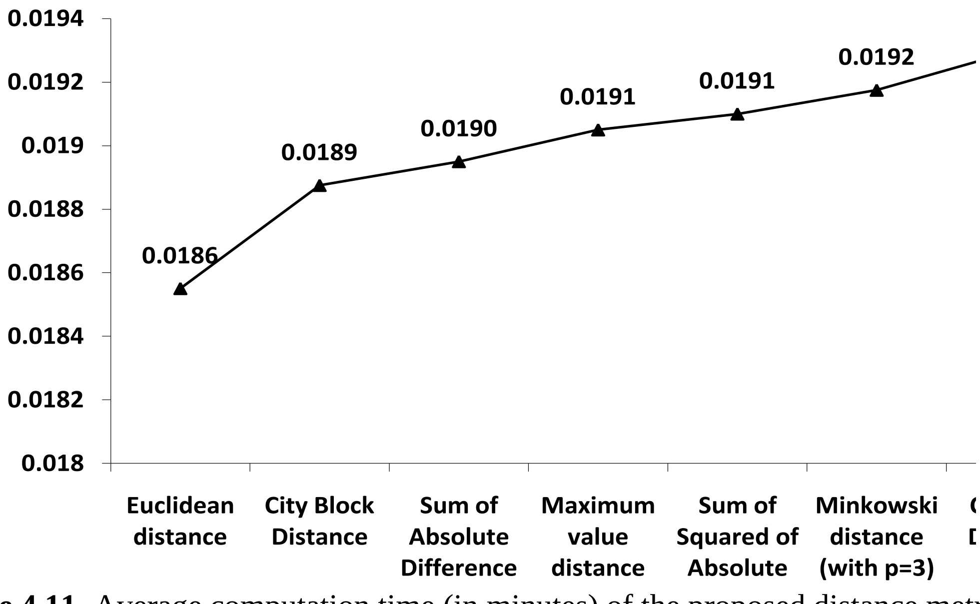


Figure 4.11 Average computation time (in minutes) of the proposed distance metrics for matching of the query image with the database images.

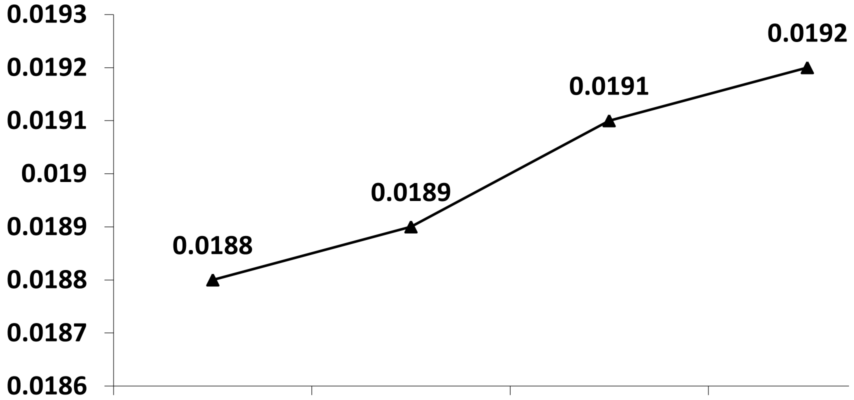
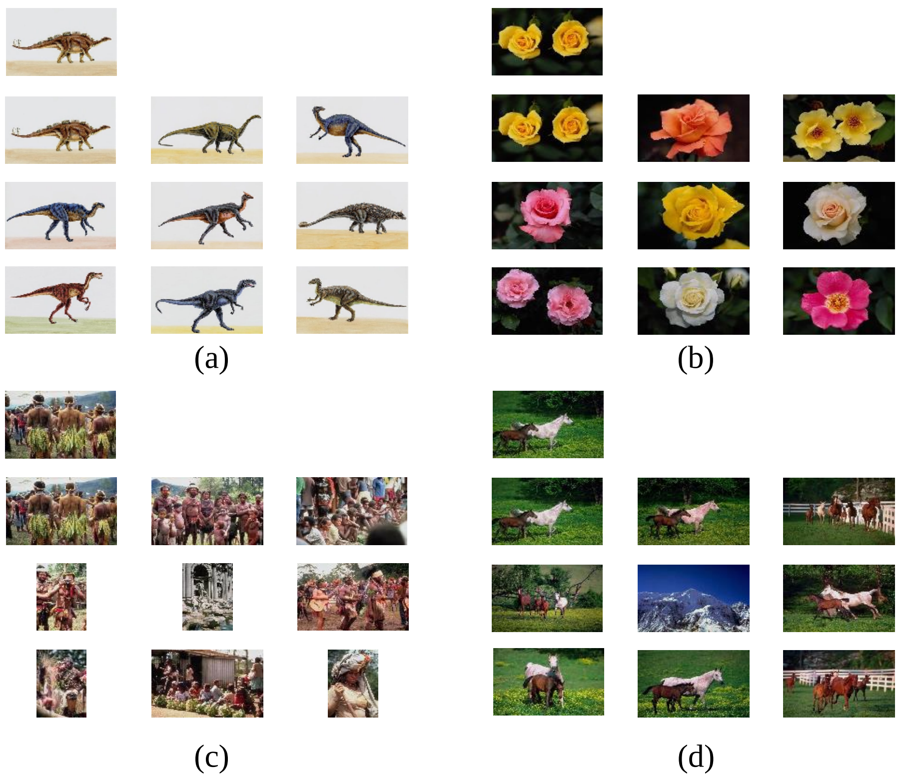


Figure 4.12 Average computation time (in minutes) of the proposed distance metrics for matching of the query image with the database images using different quantization schemes.

Figure 4.13(a-d) shows the results of the user queries using the histogram of 32 bins and the Euclidean distance for the similarity measurement. Each figure consists of a query image and the similar retrieved images from the database. The top single image is the query image and the below nine are the relevant images. The results show that proposed method has good retrieval accuracy.



(a) (b) (c) (d)

Figure 4.13 Query results of (a) dinosaurs, (b) roses, (c) people and (d) horses, using the histogram of 32 bins and the Euclidean distance for the similarity measurement.

4.3.2.2 Summary

In this section, a CBIR approach is proposed which is based on the performance analysis of various distance metrics using the quantized histogram statistical texture features in the frequency domain. Only the DC and the first three AC coefficients having more significant energy are selected in each DCT block to get the quantized histogram statistical texture features. The similarity measurement is performed by using seven distance metrics. The experimental results are analyzed on the basis of various distance metrics, separately, using different quantized histogram bins such that the Euclidean distance has better efficiency in computation and effective retrieval in the 32 bin quantization. We have concluded that the Euclidean distance, City Block Distance and Sum of Absolute Difference (SAD) metrics give good performance in terms of the F-Score using the quantized histogram texture features in the DCT domain for compressed images.

4.3.3 Experimental Analysis of the Combination of the Statistical Quantized Histogram Texture Features in the Frequency Domain (Approach-8)

In Approach-8 the retrieval results of the CBIR by extracting quantized histogram texture features, are analyzed based on the different combination of features and different quantization schemes in the frequency (DCT) domain.

4.3.3.1 Results and Discussion

The approach is performed in two steps such that in the first step, the proposed feature database is created. In the second step, the feature vector of the query image is also constructed and compared with the feature vectors of the feature database.

In the experiments, the query images from all of the image categories of the Corel database are used and the performance of the proposed approach is measured by calculating the precision, recall and F-Score.

To get the effective retrieval of similar images, the comparison of the combination of the statistical texture features is performed since a single texture feature does not give a complete description of the image to be presented. The two steps in section 3.4 are performed for the single and all of the combinations of the features using all of the image categories. In the experiments, the optimal combination of the features is selected on the basis of the retrieval of the images. We demonstrate the combination of the quantized histogram statistical texture features to get improved performance in terms of image retrieval in the DCT domain. The results of the different combinations of the texture features in the quantization of the histograms in the 32 bins in the DCT domain are shown in Tables 4.29 to 4.31.

Table 4.29 shows the average precision, recall and F-score of all of the image categories for the combination C32 having seven texture features: mean, standard deviation, skewness, kurtosis, energy, entropy and smoothness, in the quantized histogram of 32 bins in the DCT domain. It can be seen that the dinosaurs, roses, people, horses and buses give better results as compared to the other categories. All of the images of each category are used as query images. For this combination, the histograms are quantized into 32 bins. The overall average F-Score is 83%, which shows good and improved retrieval.

Table 4.29 Average precision, recall and F-Score of the query images from the image categories for the combination C32 using the quantization scheme of 32 bins.

Categories	**Precision**	**Recall**	**F-Score**
Dinosaurs	**100**	100	**100**
Roses	**96**	93	**94**
People	88	95	91
Horses	**93**	92	**93**
Buses	86	80	83
Elephants	79	80	79
Beaches	82	76	79
Buildings	77	79	78
Mountains	62	73	67
Foods	60	70	65
Average	**82**	**84**	**83**

Table 4.30 and Fig. 4.14 show the results of the single and the different combinations of texture features in the quantization scheme of 32 bins using randomly selected query images. It can be seen that the performance of the texture features in

Table 4.30 Average precision, recall and F-Score of the query images from the entire image categories for different combinations of the texture features using the quantization scheme of 32 bins.

Type	Code	Features	Precision	Recall	F-Score
Single Features	C01	Mean	61	69	65
	C02	Standard deviation	67	67	67
	C03	Skewness	62	70	66
	C04	Kurtosis	69	77	73
	C05	Energy	61	73	66
	C06	Entropy	71	70	71
	C07	Smoothness	33	47	39
		Average	61	68	64
Two Features	C08	Mean + Standard Deviation	71	69	70
	C09	Skewness + Kurtosis	70	73	71
	C10	Energy + Entropy	60	63	61
	C11	Kurtosis + Energy	63	70	66
	C12	Kurtosis + Entropy	70	75	72
		Average	67	70	68
Three Features	C13	Energy + Entropy + Smoothness	51	47	49
	C14	Mean + Standard deviation + Skewness	74	72	73
	C15	Kurtosis + Energy + Entropy	69	75	72
	C16	Mean + Standard deviation + kurtosis	79	77	78
	C17	Mean Standard deviation + Energy	71	73	72
	C18	Mean +Standard deviation + Entropy	74	73	73
	C19	Mean+ Standard deviation + Smoothness	71	72	71
		Average	70	70	70
Four Features	C20	Mean + Standard deviation + Skewness + Kurtosis	**80**	74	**77**
	C21	Mean + Standard deviation + Skewness + Energy	74	72	73
	C22	Mean + Standard deviation + Skewness + Entropy	76	74	75
	C23	Mean + Standard deviation + Skewness + Smoothness	74	75	74
	C24	Skewness + Kurtosis + Energy + Entropy	73	74	73
	C25	mean + Standard deviation + Kurtosis +Energy	79	75	**77**
	C26	Mean + Standard deviation + Kurtosis + Entropy	**80**	73	**76**
	C27	Mean + Standard deviation + Kurtosis + Smoothness	79	73	**76**
	C28	Mean + Standard deviation + Energy + Entropy	74	74	74
	C29	Mean + Standard deviation + Energy + Smoothness	72	70	71
	C30	Mean + Standard deviation + Entropy + Smoothness	74	73	73
	C31	Kurtosis + Energy + Entropy + Smoothness	73	80	76
		Average	76	74	75
Seven Features	C32	Mean + Standard deviation + Skewness + Kurtosis + Energy + Entropy + Smoothness	82	84	**83**

terms of F-Score increases as the combination of the features also increases. Using single statistical features give a low performance of a 64% F-Score. Two texture features combination gives a comparatively good average F-Score of 68%. Three features random combinations give an average F-Score of 70%. Four features random combinations provide improved performance in terms of 75% average F-Score. Finally, the combination of all the seven features, gives the optimal results of 83% average F-Score. It has been noted that the combination of the mean and standard deviation with other features give the best results, especially with skewness, kurtosis and entropy. While the combination of the other features like smoothness, energy, entropy, skewness and kurtosis with each other in different combinations like two and three features combinations give relatively low performance in terms of the F-Score. However, the combination of all of the features gives a high performance.

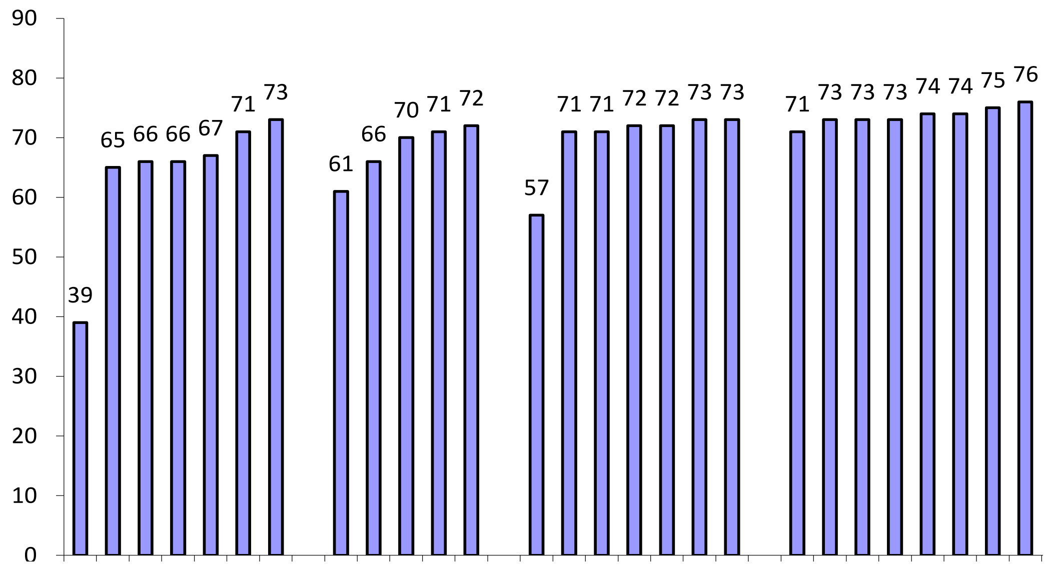


Figure 4.14 Average F-Score of the query images from the entire image categories for different combinations of the texture features using 32 bins quantization scheme.

Table 4.31 shows the overall results in the average F-Score of the different combinations. It can be seen that the F-Score increases from the single feature towards all seven features combination. The four features and all the features combinations give high performance in terms of the F-Score such as 75% and 83%. Therefore, the combination of the texture features of the quantized histograms in the DCT domain using the 32 bin quantization gives good performance in terms of the F-Score in the retrieval of similar images instead of using a single feature or a combination of texture features less than three features.

Table 4.31 Average F-Score of the different combinations of the texture features.

Feature combination	**Average F-Score**
Single Feature	64
Two features combination	68
Three features combination	70
Four features combination	**75**
All features combination	**83**

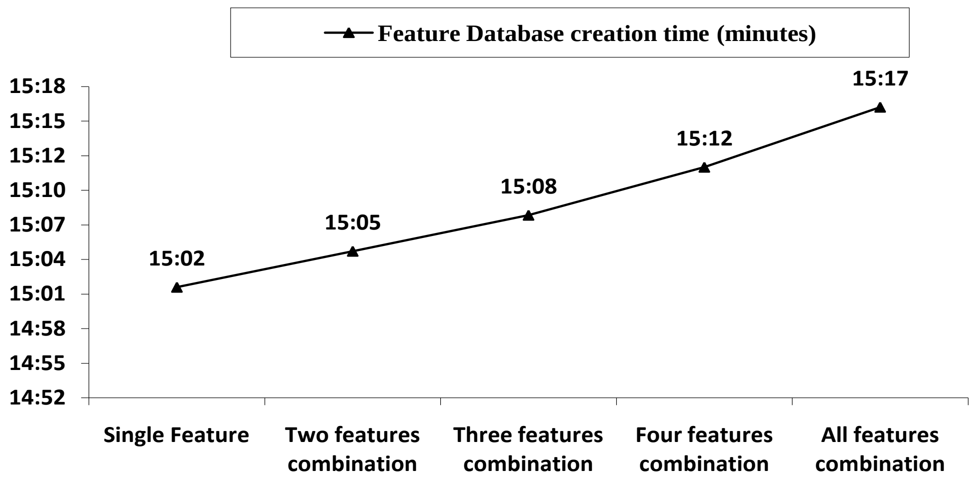


Figure 4.15 Average time taken by the different combination of the features for the creation of the feature database.

Figure 4.15 shows the average computational time taken by the different combination of texture features for creation of the feature databases. It can be seen that the combination of all seven features takes more time, with less difference of time in seconds only, than the other combinations but the retrieval performance is higher than the other combinations.

4.3.3.2 Summary

A CBIR Approach-8 has been proposed in which the statistical texture features are extracted from the quantized histograms in the DCT domain. Only the DC and the first three AC coefficients having more significant energy are selected in each DCT block to get the quantized histogram statistical texture features. In this approach, the analysis of the results of the different combinations of the statistical quantized histogram texture features is performed for the optimal features combination in terms of effective retrieval and efficiency. The experimental results show that the combination of more features gives better results as compared to a single feature or a few texture features combination. The quantization in 32 bins for the four and for the all seven texture features combinations gives the optimal results in terms of the F-Score as compared to the low and high bins quantization.

4.3.4 Analysis of the Experimental Performance of the Combination of the Statistical Texture Features in the Spatial and Frequency Domains for CBIR (Approach-9)

In Approach-9 the retrieval results of the CBIR by extracting quantized histogram texture features, are analyzed based on the combination of texture features in the spatial and frequency domains.

4.3.4.1 Results and Discussion

The approach is performed in two steps such that in the first step, the feature database is created. In the second step, the feature vector of the query image is also constructed and compared with the feature vectors of the feature database.

In the experiments, the query images which are randomly selected from all of the image categories of the Corel database are used and the performance of the proposed approach is measured by calculating the average precision, recall and F-Score. The results are shown in Table 4.32.

Table 4.32 shows the average precision, recall and F-Score in percentage for all of the categories by extracting the statistical texture features of the images using the 8×8 block method in the spatial domain and the 8×8 DCT block transformation in the frequency domain. The result in terms of F-Scores of the proposed approach is improved and better for all of the categories, especially for dinosaurs, roses, horses and buses. These categories provide significant texture information in the 8×8 block partition in the spatial and frequency domains using the Euclidean distance for the similarity. The overall average F-Score is 84% which shows improved performance in terms of retrieval of similar images by combining the feature vectors of the texture features in the spatial and frequency domains of the grayscale images by dividing them into 8×8 non-overlapping blocks.

The average computation time taken by the proposed approach for the creation of the feature database is 15 minutes which is reasonable time for creation of a database of images which are created once and are to be used for the matching with the query image.

Table 4.32 Average precision, recall and F-Score of the query images using the combination of the texture features in the spatial and frequency domains.

Categories	Precision	Recall	F-Score
Dinosaurs	**100**	100	**100**
Roses	**100**	95	**97**
Horses	**100**	90	**95**
Buses	93	88	91
Elephants	85	90	87
Beaches	82	79	80
Buildings	76	79	77
Foods	72	75	73
People	73	70	71
Mountains	56	83	67
Average	84	85	**84**

Thus, the performance of the proposed Approach-9 is not only efficient in the computations of the feature extraction but also effective in terms of the retrieval of similar images.

4.3.4.2 Summary

In this section an approach for CBIR has been proposed which is based on the combination of the feature vectors of the statistical texture features in the spatial and frequency domains using the 8×8 non-overlapping block methods and the DCT block transformation. The probability distribution of the intensity levels of all of the blocks have been used to calculate the texture features to create feature vectors in both domains. These feature vectors are combined for retrieval of similar images and these features are extracted from non-overlapping blocks of different sizes of the images. For the similarity measurement, the Euclidean distance is used to measure the similarity of the query image with images in the database. The experimental results show that the proposed approach is efficient in the feature extraction as well as effective in retrieval.

4.3.5 Analysis of the Experimental Combination of the Color and Texture Features in the Spatial and Frequency Domains (Approach-10)

In Approach-10 the retrieval results of the CBIR by extracting quantized histogram texture features, are analyzed based on the combination of color features in the spatial and the texture features in the frequency domains.

4.3.5.1 Results and Discussion

The approach is performed in two steps such that in the first step, the proposed feature database is created. In the second step, the feature vector of the query image is also constructed and compared with the feature vectors of the feature database.

In the experiments, the query images are randomly selected from all of the image categories of the Corel database and the performance of the proposed approach is measured by calculating the precision, recall and F-Score. The results are shown in Table 4.33.

Table 4.33 Average precision, recall and F-Score of the query images using the combination of the color and texture features in the spatial and frequency domains.

Categories	Precision	Recall	F-Score
Dinosaurs	**100**	100	**100**
Roses	**100**	98	**99**
Horses	**97**	95	**96**
Buses	95	90	92
Elephants	94	86	90
Beaches	89	82	85
Buildings	77	78	77
Foods	71	80	75
People	70	76	73
Mountains	72	70	71
Average	**87**	86	**86**

Table 4.33 shows the precision, recall and F-Score in percentage for all of the categories by extracting the statistical color and texture features of the images using the 8×8 block method in the spatial domain and the 8×8 DCT block transformation in

the frequency domain. The results in terms of the F-Scores of the proposed approach are improved and better for all of the image categories, especially for dinosaurs, roses, horses and buses. These categories provide significant color information in the 8×8 block partition in the spatial domain and texture in the frequency domain using the Euclidean distance for the similarity. The overall average F-Score is 86% which shows improved performance in terms of the retrieval of similar images. This is accomplished by combining the color feature vector with the texture feature vector in the spatial and frequency domains of the grayscale images by dividing them into 8×8 non-overlapping blocks.

The average computational time taken by the proposed approach for the creation of the feature database is 14.27 minutes which is a reasonable time for creation of the feature database. Thus the performance of the proposed Approach-10 is not only efficient in the computations of the feature extraction but also effective in terms of the retrieval of similar images.

4.3.5.2 Summary

In this section, an approach for CBIR is proposed which is based on the combination of the feature vectors of the statistical color and texture features in the spatial and frequency domains using the 8×8 non-overlapping block method and the DCT block transformation. The probability distribution of the intensity levels of all of the blocks have been used to calculate the color and texture features to create the feature vectors in both domains. These feature vectors are combined for the retrieval of similar images and these features are extracted from the non-overlapping blocks of the images. For the similarity measurement, the Euclidean distance is used to measure the similarity of the query image with the images in the database. The experimental results show that the proposed approach is efficient in the feature extraction as well as effective in retrieval.

4.4 Chapter Summary

In this chapter, the results of the various proposed approaches for the extraction of color and textures have been discussed and analyzed in the spatial and frequency domains to get an efficient CBIR.

In spatial domain, for extraction of the color features, the color histogram, color histogram refinement method and sub-block methods are used; while for the extraction of texture features, the block methods are used in various approaches. In Approach-1, the color histogram technique for a grayscale Laplacian filtered sharpened image gives retrieval performance with 66% average F-Score using 32 bins quantization scheme and 0.515 minutes as computational cost of creation of feature database. In Approach-2, to increase the efficiency of retrieval and retain the spatial information, the color histogram refinement method is used based on filters with improved retrieval performance of 75% average F-Score using Laplacian filter with 32 bins quantization but a high computation cost of 50 minutes of feature database creation. In Approach-3, to reduce the computational cost and improve the efficiency, the local statistical color features are extracted from non-overlapping sub-blocks of different sizes. However after experimental results the 8×8 block method gives 71% F-Score with reduced features database creation computational cost of 0.5155 minutes. In Approach-4, to reduce the computational cost furthermore and improve the efficiency, the local statistical texture features are extracted from non-overlapping sub-blocks of different sizes. Again the 8×8 block method gives 73% F-Score with reduced features database creation computational cost of 0.314 minutes. In order to improve the efficiency and reduce the computational cost, it has been shown that the combination of color and texture features in Approach-5 using 8×8 sub-block method, gives a robust retrieval of 82% average F-Score with the near to optimum features database creation computational cost of 0.5824 minutes. After the analysis of the results of the Approache-1 to Approach-5, it has been shown that the performance of the retrieval is incremental from the histogram to the block methods. The combination of the color and texture features in the block methods gives near to optimum retrieval performance in the spatial domain.

In frequency domain, quantized histogram texture features are extracted from the transformed non-overlapping 8×8 DCT blocks by using only the DC and the first three AC coefficients having more significant energy in Approach-6 to Appraoch-10.

In Approach-6, it has been shown after experimental results that using the quantized histogram texture feature based on Laplacian filter in the DCT domain for the compressed images, an improved retrieval performance is achieved with a 78% average F-Score using 32 bins quantization scheme and with features database creation computational cost of 16.27 minutes. In Approach-7, the retrieval performance is analyzed based on various distance metrics using the quantized histogram texture features and it has been shown that the Euclidean distance has better efficiency in computation and an effective retrieval with an 83% average F-Score in the 32 bins quantization. In Approach-8 it has been shown that combination of statistical quantized texture features in the DCT domain gives improved performance with combination of more features. In Approach-9, it has been show that the combination of the feature vectors of the statistical texture features in both domains of the spatial and frequency using the 8×8 non-overlapping blocks gives performance with 84% average F-Score using 32 bins quantization scheme and with features database creation computational cost of 15 minutes. In the last Approach-10, it has been show that the combination of the feature vectors of the statistical color and texture features in both domains of the spatial and frequency, achieves retrieval performance with 86% average F-Score using 32 bins quantization scheme and with features database creation computational cost of 14.27minutes.

After the analysis of the results of the different approaches, it has been shown that the performance of retrieval is robust to the color and texture features in the spatial and frequency domains, especially with the combination of color and texture features in both domains.

CHAPTER 5

PERFORMANCE ANALYSIS OF THE PROPOSED METHODS

5.1 Performance Analysis of the Proposed Approaches

In this chapter, comparisons of the proposed approaches for the effective CBIR are performed among themselves and then with the other approaches in literature. The comparisons of the approaches which are discussed and analyzed in Chapters 3 and 4 are discussed:

5.1.1 Approaches in the Spatial Domain

The results of the Approch-1 to Approch-5 are given in Table 5.1 and Fig. 5.1in terms of F-Score by selecting randomly the query images from all of the image categories of the Corel image database in the spatial domain according to the recommendations for implementation of CBIR algorithm given in section 4.1 by (Park *et al*., 2008). In order to improve the retrieval performance of CBIR, different techniques are fused in these approaches. In all of the approaches, the distribution of the pixel values in the grayscale images are used to compute the feature vectors using different techniques to retrieve the similar images to the example query image. It can be seen that the retrieval performance is incremental and improved from Approach-1 to Approach-5 as shown in Table 5.1. In Approach-1, the quantized color histogram features are extracted from the grayscale Laplacian filtered sharpened image. The analysis is performed on the basis of the different numbers of quantization bins and the 32 bins quantization give good results with an average F-Score 66%. In Approach-2, the color histogram refinement method is used to extract the features by computing the areas of the regions of the objects of the median and Laplacian filtered images. An analysis of the results is performed on the basis of the filters and the different numbers of quantization bins, and it has been shown that using the Laplacian filter with the 32 bins quantization gives improved retrieval with an average F-Score 75%. In Approach-3 and Approach-4, the color and texture features are extracted from the grayscale images by dividing them into different numbers of sub-blocks. It has been shown that the 8×8 block method provides important color and texture information for the retrieval and gives improved results with an average F-Score 71% and 73% in terms of retrieval. In Approach-5, the color and texture features are combined using the 8×8 block method and retrieval is an average F-Score 82%. This shows that the combination of the statistical color and texture features in the spatial domain using the block method give better results as compared to the other approaches.

Table 5.1 Comparison of the proposed approaches in terms of the F-Score for the query images from the entire image categories in the spatial domain.

Categories	**Approach-1**	**Approach-2**	**Approach-3**	**Approach-4**	**Approach-5**
Dinosaurs	86	100	100	100	**100**
Roses	71	95	74	99	**100**
Horses	67	94	76	70	**91**
Elephants	68	79	74	80	87
Buses	65	77	80	60	81
Mountains	64	56	67	60	77
Beaches	58	59	57	55	74
Foods	68	48	57	65	73
Buildings	55	79	63	69	70
People	56	66	63	74	64
Average	66	75	71	73	**82**

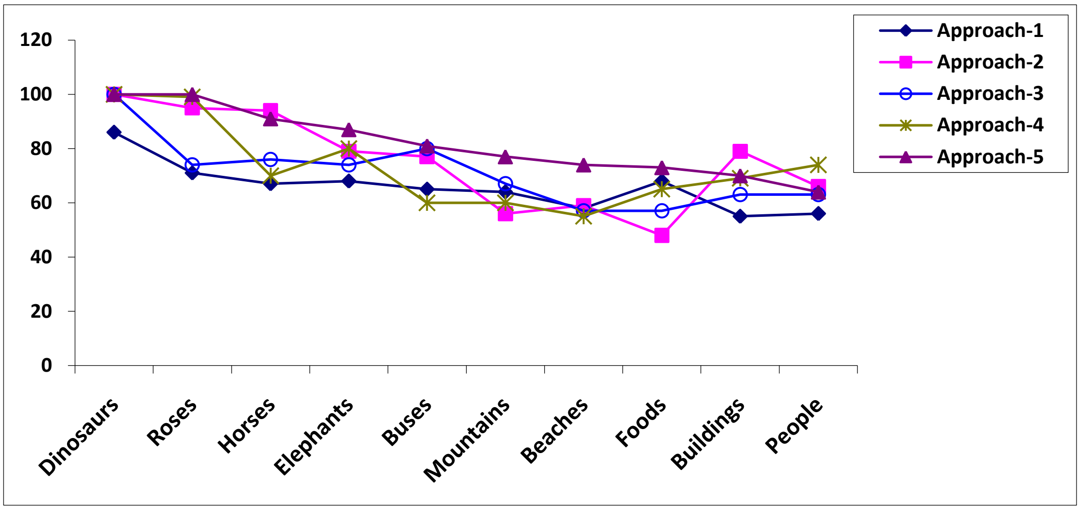


Figure 5.1 Comparison of the approaches in terms of the F-Score for the query images from the image categories in the spatial domain.

5.1.2 Approaches in the Frequency Domain

In order to improve the retrieval performance of CBIR, the different techniques are used in the frequency domain, in Approache-6 to Approach-10. Using these approaches, the grayscale image is transformed into 8×8 DCT blocks. The DC and first three AC coefficients are used to compute the feature vectors using different techniques to retrieve the similar images to the example query image. The results are shown in Table 5.2 and Fig. 5.2. It can be seen that the retrieval performance is further increased and improved from Approach-6 to Approch-10. The Approach-8 is in not

Table 5.2 Comparison of the approaches in terms of the F-Score for the query images from the image categories in the frequency domain.

Categories	**Approach-6**	**Approach-7**	**Approach-9**	**Approach-10**
Dinosaurs	100	100	100	**100**
Roses	79	94	97	**99**
Horses	85	93	95	**96**
Buses	78	83	91	92
Elephants	60	79	87	90
Beaches	83	79	80	85
Buildings	69	78	77	77
Foods	62	65	73	75
People	100	91	71	73
Mountains	60	67	67	71
Average	78	83	84	**86**

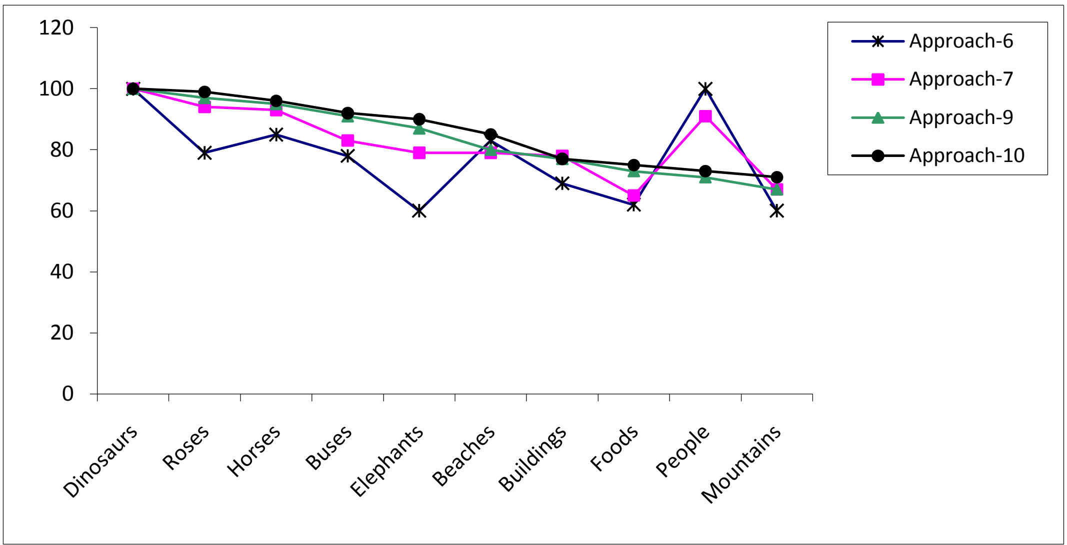


Figure 5.2 Comparison of the approaches in terms of the F-Score for the query images from the image categories in the frequency domain.

shown in Table 5.2 because the results of Approache-7 and Approache-8 are same. The optimum results of combination of features in the Approach-8 are used also by the Approach-7.

In Approach-6, the quantized histogram texture features are extracted from the 8×8 DCT blocks of the grayscale image based on the median and Laplacian filters, using the DC and AC coefficients of the blocks. The analysis is performed on the basis of the different numbers of quantization bins, and the 32 bin quantization gives good results with an average of 78% retrieval using the Laplacian filter. But, this result is less than the results of Approach-5. Hence, to further improve retrieval, in Approach-7, the quantized histogram texture features are extracted from the 8×8 DCT blocks of the grayscale image; however, the analysis of the results are performed on the basis of the various distance metrics used for the similarity measurement and on the different numbers of the quantization bins. It has been shown that the retrieval performance is better with an average 83% F-Score using the Euclidean distance and the 32 bins histogram quantization. In Approach-8, the analysis of the results of the different combinations of the statistical quantized histogram texture features is performed for the optimal feature combination in terms of effective retrieval and efficiency. The experimental results show that the combination of more features gives better results as compared to a single feature or a few texture features combination. The quantization in 32 bins for the four and for the all seven texture feature combinations gives the improved results in terms of the F-Score as compared to the low and high bins quantization. In Approach-9, the statistical texture features in the spatial domain using the 8×8 block method are combined with the quantized histogram texture features in the frequency domain using the 8×8 DCT blocks of the grayscale image, and the retrieval is an average F-Score 84%. In the last Approach-10, the color and texture features are combined using the 8×8 blocks in the spatial and the 8×8 DCT blocks in the frequency domains; the retrieval average for the F-Score is 86%. This shows that the combination of the statistical color and texture features in the spatial and frequency domains, using the 8×8 blocks of the image, gives better results as compared to the other approaches in the spatial as well as in the frequency domains.

In Fig. 5.3 and Fig. 5.4, the results of all of the approaches in the spatial and frequency domains are shown and it is clear that Approach-10 with the combination of color and texture features has overall the robust retrieval as compared with the other approaches. Thus, we conclude that the 8×8 block conversion of the grayscale image provides important color and texture information for the retrieval of the similar images in the spatial and frequency domains using the 32 bins histogram quantization.

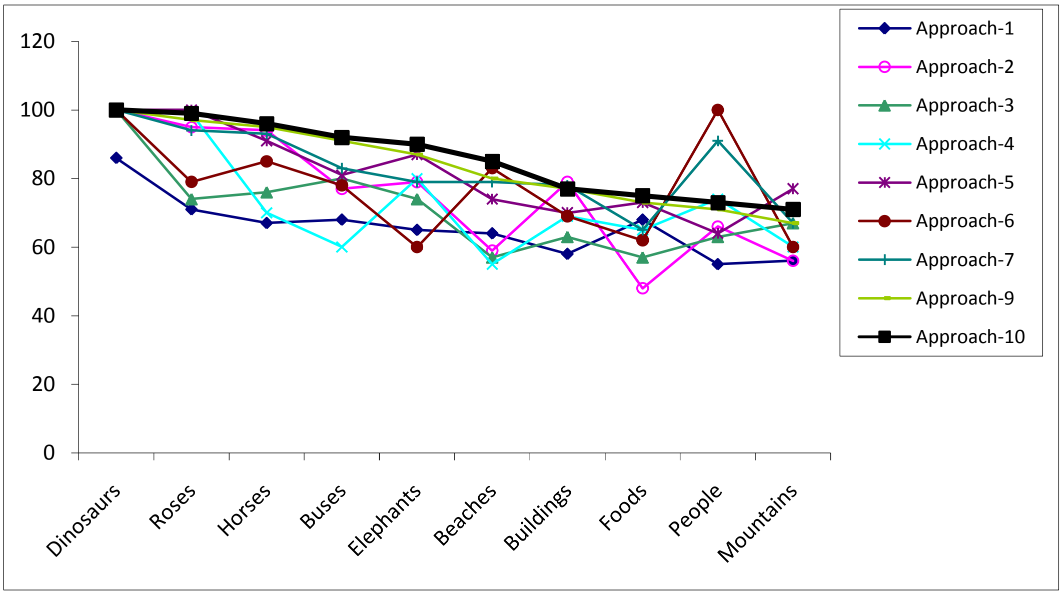


Figure 5.3 Comparison of all the proposed approaches in terms of the F-Score in the spatial and frequency domains for the query images from the entire image categories.

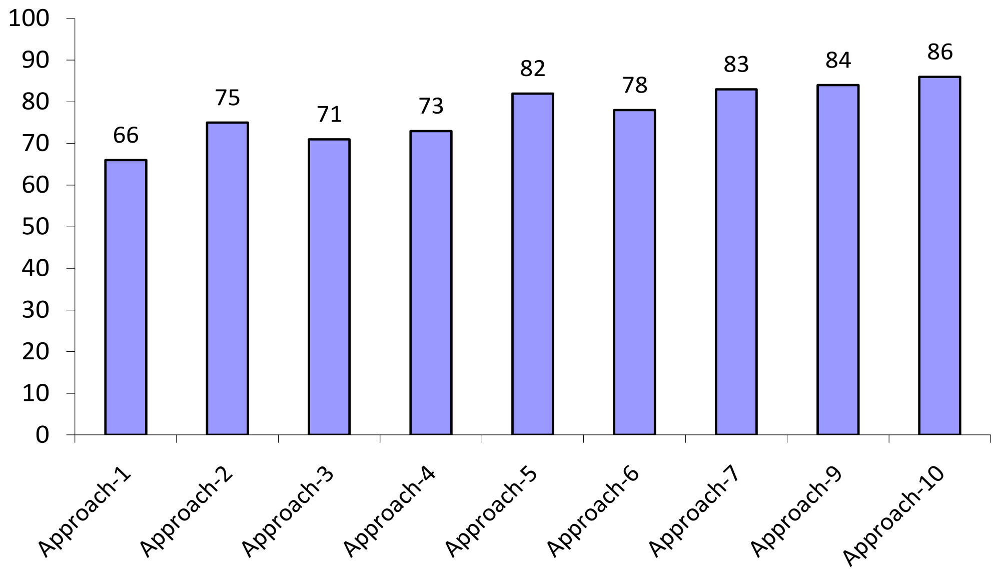
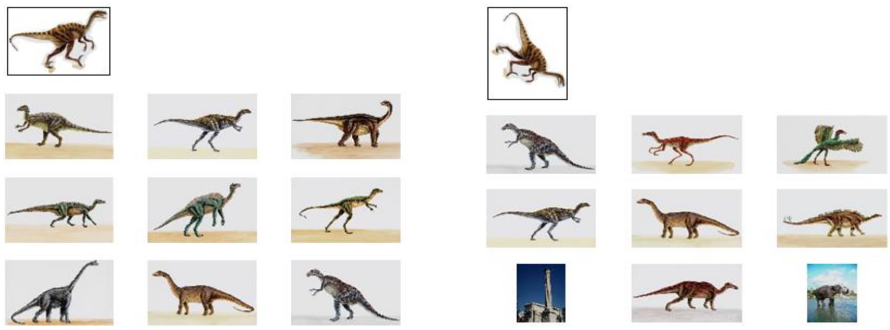


Figure 5.4 Comparison of the approaches in terms of the F-Score in the spatial and frequency domains for the query images from the entire image categories.

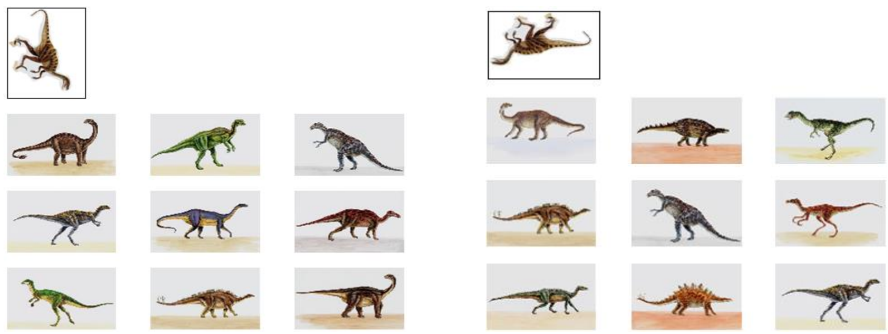
The proposed approaches can also accept any image as query image for retrieval of similar images other than the feature database images as shown in Fig. 5.5. These approaches are independent of dimension (size). For example the query images in Fig. 5.5(a) and Fig. 5.5(b) are taken from the outside source other than Corel Dataset and Fig. 5.5(a) has dimension of 734 × 817 pixels while Fig. 5.5(b) has dimension of 425 × 309 pixels. Hence an image with any dimension can be accepted.

(a) (b)

Figure 5.5 Query image results of (a) with dimension 734 × 817 (b) with dimension 425 × 309 using Approach-10.



(a) (b)



(c) (d)

Figure 5.6 Results of query image with rotation of (a) angle 0°, (b) angle 75° , (c) angle 90° and (d) angle 180° using Approch-10.

The proposed approaches are invariant to the rotation as shown in Fig. 5.6. By changing the rotation of the image at different angles like 0°, 75°, 90° and 180° the results of the retrieval are not so affected.

5.2 Performance Analysis of the Proposed Approaches with the Other Approaches in Related Works

Evaluation criteria for CBIR depend upon the benchmarking which is still an issue for the researchers. However, there are recommendations issued by the technical committee from the International Association for Pattern Recognition (IAPR) regarding benchmarking of image retrieval (An *et al.*, 2011). Guidelines based on the recommendations for implementation of CBIR algorithm are given by (Park *et al*., 2008), which have been discussed in section 4.1.

The results of the proposed near to optimum Approach-10, are also compared with the other approaches of (Hiremath and Pujari, 2007; Mohamed *et al.*, 2009; Murala *et al.*, 2009; Kavitha *et al.*, 2011; Soman *et al.*, 2011; Thawari and Janwe, 2011; Alnihoud, 2012; Singha and Hemachandran, 2012) based on the precision as shown in Table 5.3. In the method of (Hiremath and Pujari, 2007), color, texture and shape features are fused together and extracted in a non-overlapping partitioned image. Texture features are extracted by using the Gabor filter, the statistical color moments are used to calculate the color features. The shape of the objects is extracted by using the Gradient vector flow fields and the shape features are depicted by using invariant moments. The method is tested by using the Corel image database and the overall average precision is 55%. In the method of (Mohamed *et al.*, 2009), an image is divided into non-overlapping 8×8 blocks and then these blocks are transformed into the DCT domain. The DC and the first three AC coefficients of each block are picked up in a zigzag order. These coefficients are used to construct quantized histograms of 32 bins. These histograms are used to construct a feature vector for retrieval. They tested their method with the animal dataset and got an average precision of 70%.

In the method of (Murala *et al.*, 2009), the color and texture features are combined to retrieve similar images. For the color features, the mean and standard deviation are computed in a histogram of 64 bins in each channel of the RGB color image, to get a total of 192 features. For the texture features, the mean and standard deviation are computed in sub bands of the Gabor Wavelet Transform image with the three scales and four orientations to get a feature vector of 48 features. The performance of this method is measured in terms of an average precision of 65%.

Table 5.3 Comparison of the proposed Approach-10 with other approaches in terms of precision for the query images from the entire Corel's image categories.

Categories	**Hiremath and Pujari, 2007**	**Mohamed *et al.*, 2009**	**Murala *et al.*, 2009**	**Thawari and Janwe, 2011**	**Kavitha *et al.*, 2011**	**Soman *et al.*, 2011**	**Alnihoud, 2012**	**Singha and Hemachandran, 2012)**	**Proposed Approach-10**
Dinosaurs	95	99	100	90	61	80	100	97	**100**
Elephants	48	70	62	N/A	39	60	50	86	**94**
Horses	74	40	91	N/A	35	80	85	87	**97**
Roses	61	N/A	80	N/A	87	88	97	76	**100**
People	48	N/A	76	50	44	55	87	65	**70**
Buses	61	N/A	52	50	75	50	84	92	**95**
Beaches	34	N/A	50	40	50	50	68	62	**89**
Buildings	36	N/A	47	35	45	25	70	71	**77**
Mountains	42	N/A	28	N/A	34	40	32	49	**72**
Foods	50	N/A	63	N/A	31	40	63	77	**71**
Average	**55**	**70**	**65**	**53**	**50**	**57**	**74**	**76**	**87**

In the approach of (Thawari and Janwe, 2011) the HSV color space is used with three color channels, H, S and V. The histogram of each channel is quantized into 96 blocks, and each block has a dimension of 32×32 pixels. The statistical texture moments of mean, standard deviation, skew, kurtosis, energy; entropy and smoothness are calculated in each bin of the histogram. The total 96×7×3=2016 features are computed. Thus, this process of feature extraction involves a large number of computations which increase computational cost. The method has used 500 images of the Corel database for testing. The average precision of the method in (Thawari and Janwe, 2011) is 53%. In the approach of (Kavitha *et al.*, 2011), the color and texture features are also combined. The HSV color space image is divided into sub-blocks. The color features are calculated by quantizing the histograms of each block. The texture features are calculated by using the grey level co-occurrence matrix. The most similar highest priority (MSHP) principle is used to calculate the difference between the query and the target image blocks. The Euclidean distance is used to calculate the similarity measurement. In the approach of (Soman *et al.*, 2011), the color and texture features are also combined. For the color features, the color moments of mean, standard deviation and skewness are calculated in 8×8 blocks of the three components of the RGB color image to retrieve the similar images. The texture features are calculated in 8×8 DCT blocks by computing the DC and AC coefficients in 9 directions to get the feature vectors of the query and the retrieved images using the color features. In the method of (Alnihoud, 2012), the color and shape features are extracted based on the SOM (self-organizing map). A Fuzzy Color Histogram (FCH) and subtractive fuzzy clustering algorithms are used to get the color features and the object Model Algorithm is used to get the edge of the objects. Then, the shape features like area, centroid, major axis length, minor axis length, eccentricity and orientation are computed to get the performance in terms of the average precision of 74%. The CBIR approach proposed by (Singha and Hemachandran, 2012) is based on the combination of the texture and color features. For the color features, the RGB image is converted into the HSV (Hue, Saturation and Value) color space. Each color component is quantized into 8 bins to get normalized histogram color features. For the texture features, the RGB color image is transformed using the Haar Wavelet Transformation to get vertical, horizontal and diagonal coefficients. These coefficients are converted into the HSV plane and each component is quantized into 8 bins to get normalized histogram texture features which are combined with the color features. The Histogram Intersection Distance method is used for the similarity measurements.

In order to get an effective and efficient CBIR, in this research work, we have fused the various approaches in the spatial and frequency domains. After the comprehensive analysis of the approaches, we have concluded that the fusion of the color and texture features in the spatial and frequency domains in Approach-10, gives the optimal and most robust performance of retrieval. Hence, the proposed Approach-10 is compared with the other approaches in Table 5.3 in terms of precision. This proposed approach starts with the conversion of the RGB color images into grayscale images to reduce the computational cost. To extract the color features, the grayscale image is divided into 8×8 non-overlapping blocks and the texture features are extracted by converting the grayscale image into non-overlapping 8×8 DCT blocks. The DC and first three AC coefficients with significant energy of each block are picked up in a zigzag order to construct the histograms. The histograms are quantized into 32 bins to calculate the statistical texture features in the histogram bins. The color and texture features are combined to retrieve the similar images. The proposed method is tested by using the same Corel image database as used by the other existing methods. The results of this approach are effective not only in retrieval but also in efficiency. The overall average precision of this proposed method is 87%, which is higher than the other methods using the 32 bins histograms. Thus, in this proposed approach, to reduce the computational cost, the image is converted into a single plane grayscale image. The mean and standard deviation are calculated as the color moments and the seven texture features in non–overlapping blocks. For the similarity measurement, the Euclidean distance is used. The proposed method has good performance in terms of precision using the color quantized texture features and the Euclidean distance for the similarity measurement in the spatial and DCT domains for the compressed images as shown in Table 5.3.

However the difference between our proposed Approach-6 and the approach of (Mohamed et al., 2009 is that we have used filters to get enhanced image with convoluted values of images and then histograms of the transformed values of these filtered values are constructed and quantized by using different quantization schemes. After that we calculate statistical texture features of histograms. They have constructed only histograms of coefficients of blocks and quantized only in 32 bins and histograms are used as features. We have studied the effect of texture features in DCT domain and the results are analyzed on the basis of filters as well different quantization schemes.

In all of the above approaches of the related works to calculate the precision, an image from any category of the Corel database is selected randomly and the relevant images are displayed to the user according to the query. Similarly in our approaches we also calculate the precision by selecting randomly 30 images from all the image categories and the relevant images are displayed to the users as thumbnails.

5.3 Chapter Summary

In this chapter, comparisons of all the proposed approaches in the spatial and frequency domains for the effective CBIR are performed first among themselves and then the proposed approach with near to optimum performance with the other approaches in the literature.

In the spatial domain different techniques used for color and texture features in various approaches, however the Approach-5 gives optimum performance for the combination of color and texture features using the 8×8 block method with an average F-Score of 82%.

In the frequency domain different techniques used for texture features in various approaches, however the Approach-10 gives optimum performance for the combination of color and texture features in the spatial and frequency domains with an average F-Score of 86%.

Finally our Approach-10 with near to optimum result of 87% average precision is compared with the other existing approaches in the literatures. It has been shown that our proposed approach has better performance than the other approaches in terms of retrieval.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Summary of Contributions

Effective CBIR is based on efficient feature extraction for indexing and on effective query image matching with the indexed images for retrieval. However the main issue in CBIR is that how to extract the features efficiently because the efficient features describe well the image and they are used efficiently in matching of images to get robust image retrieval. This issue is the main inspiration for this thesis to develop an efficient hybrid CBIR with high performance in the spatial and frequency domains using color and texture features. We propose various approaches in which different techniques are fused to extract the statistical color and texture features efficiently in both domains. In the spatial domain, the statistical color histogram features are computed using the pixel distribution of the Laplacian filtered sharpened images based on the different quantization schemes. However color histogram does not provide the spatial information. The solution is by using the histogram refinement method in which the statistical features of the regions in histogram bins of the filtered image are extracted. This approach gives efficient retrieval but it has high computational cost, which is reduced by dividing the image into the sub-blocks of different sizes, to extract the local color and texture features of images and also to get local information in place of using complex segmentation. To improve further the performance, the color and texture features are combined using sub-block methods due to the less computational cost and good local information.

In the frequency domain, the statistical quantized histogram texture features are extracted from 8×8 DCT (Discrete Cosine Transformation) blocks and effectiveness of CBIR is studied based on: median and Laplacian filters, distance metrics, different combination of features, combination of texture features in both domains and combination of color and texture features in both domains are presented in order to get an efficient hybrid CBIR. Experimental results using benchmark Corel database have been shown that the proposed approaches achieve an average accuracy of 82% in spatial domain and 86% in frequency domain and the improved performance of proposed approaches outperform the approaches in the related works in the literature.

In this thesis, we present various approaches in the spatial and frequency domains for image retrieval in CBIR. The approaches in the spatial domain include: approaches for color features using histogram and the color histogram refinement method (CHRM) based on median and Laplacian filters, and the approaches for the color and texture features based on sub-blocks of an image. The approaches in the frequency domain include: approaches for the quantized histogram texture features using the DCT blocks based on median and Laplacian filters, various distance metrics and various combinations of features. Other approaches based on the combination of the texture features in the spatial and frequency domains and the combination of the color in the spatial domain with the texture features in the frequency domain have also been presented.

The contribution of the thesis starts with a proposed Approach-1 in the spatial domain in which we have used the Laplacian sharpened grayscale image for the feature extraction because the energy is compensated in the sharpen method, which is lost by the Laplacian filter in the preprocessing of the grayscale image to get a sharpened and enhanced image without noise. In the sharpening processing using the Laplacian filter, not only is the noise reduced but the information is also preserved that plays useful role in retrieval of the similar images from a database. The sharpened image is quantized using different quantization schemes like 4, 8, 16 and 32 bins. The statistical color features are extracted from the bins and represented in feature vectors. These vectors are used in the similarity measurement for the retrieval of similar images. From the results, it has been concluded that the quantization scheme with histograms of 32 bins gives good performance in terms of the F-Score in the retrieval of similar images. For the evaluation of the retrieval performance, another Approach-2 is proposed which is based on the performance of the different filter methods using the color histogram refinement method for the feature extraction in which the pixel values are convoluted with filter values. Median, median with edge extraction and

Laplacian filter methods are applied on the grayscale images for the noise removal before applying the histogram method. During the median filtration, edge information is lost which is restored by the Canny edge detection method while in the Laplacian filter some information is also lost which is restored by subtracting the Laplacian image from the grayscale image to get a more enhanced and sharpened image. The statistical features of mean and standard deviation of the quantized histograms are calculated using the mathematically computed values of the connected regions. These statistical features are used for the retrieval of the relevant images. These features do not depend upon the orientation of the image. In this approach, the spatial information in the images is preserved. The performance is analyzed on the basis of the three filter methods using the spatial information of the histograms by quantizing them using different quantization schemes. The results show that the approach is efficient in retrieval but has high computational cost.

To reduce the computational cost and improve the efficiency of retrieval an Approach-3 is proposed which is based on the statistical color moments and these moments are extracted from the non-overlapping sub-blocks of different sizes using the distribution of the pixel values of the images. It has been shown in this approach that the statistical color features has good retrieval performance by using the statistical local information of the local blocks in the images. In this approach, 9 different sub-block methods have been used. The color moments have been extracted in all of the methods and their individual retrieval performance has been analyzed in terms of the F-Score. In another proposed Approach-4, the statistical texture features are extracted using the distribution of the pixels in the sub-blocks of the grayscale image. These features represent the brightness, contrast, skewness, flatness, uniformity, randomness and smoothness in the sub-blocks of the images. In the experiment, the results show that the proposed approaches using the sub-block methods are efficient in color and texture feature extraction for the different block methods and give the best performance in terms of accuracy comparatively, especially for the 8×8 and 16×16 block methods. In the spatial domain Approach-5 is proposed in which the color and texture features are combined using the sub-block methods, and the improved good results are provided by using the 8×8 sub-blocks.

In order to improve the retrieval performance of CBIR, an Approach-6 has been proposed in the frequency domain for the effective image retrieval in which the experimental analysis of the statistical texture features based on the median and Laplacian filters is performed in the DCT domain. Only the DC and the first three AC coefficients with more important energy are selected in each DCT block to get the quantized histogram statistical texture features. These features are extracted from the median, median with edge extraction and Laplacian filtered images. The experimental analysis is performed on the basis of the results of the three filter methods using the different quantization schemes, and it has been shown that the enhanced and sharpened Laplacian filtered images using the quantized histogram texture features give good performance in terms of the F-Score in the DCT domain for the compressed images.

An analysis of the statistical quantized histogram texture features is presented in another Approach-7 in the DCT domain based on the various distance metrics using the different quantization schemes. It has been shown that different values are calculated by using different distance metrics for the similarity measurements. The Euclidean distance has the optimal retrieval performance using the 32 bins quantization scheme as compared to the other proposed distance metrics. The combination of the statistical quantized histogram texture features is presented in another Approach-8 in the DCT domain, and it has been shown that the combination of more than one texture features give the optimal and improved performance in terms of retrieval.

Finally, an Approach-9, to combine the textures features in the spatial and frequency domains has been presented. While in another Approch-10, the color features in the spatial domain are combined with the quantized histogram texture features in the frequency domain. It has been shown that the color and texture features' integration in the spatial and frequency domains provide the best results as compared to the other approaches in the spatial and frequency domains. The performances of the proposed approaches are compared to the related works, and we give an extensive discussion on their performances. The detailed results of the retrieved images for the different query images on the different categories of the benchmark Corel database demonstrate the effectiveness of these approaches and the combination schemes.

In summary, the proposed work in this thesis has mainly focused on the extraction of the statistical color and texture information of the images for the effective retrieval of similar images. The histograms and sub-block methods provide local color and texture information of the grayscale image. The last approach includes both the color information of the pixel values and the texture information of image.

6.2 Limitations of the Research Work

By the nature of knowledge, every research work in any filed has some limitations to make sure the future research directions in that filed. In the same way, the work in this thesis also has some limitations as follows:

- The proposed approaches do not provide good retrieval to the images with complex background.
- Images with multiple objects in foreground also affect the retrieval performance with some extent of the proposed work.
- Feature database is created once and loaded at run time for searching but new image cannot be added at run time. For new addition the whole feature database will be recreated.
- Object-based search cannot be performed and the retrieval is only on the global information of the overall image.
- Another problem in this work is that the proposed features do not describe the images, semantically. This lack of semantics is called the semantic gap. The semantic gap is the lack of information between the user query and the retrieval algorithm. The semantic gap between the retrieval using low level features and the high level user’s request leads CBIR algorithms to the wrong results (Smeulders *et al.*, 2000).
- Only tested in Corel dataset.

6.3 Future Work

The limitations of the research work give new directions for future research work. Hence the limitations of this work also provide foundation for future research.

- In this thesis, the work has shown that the global histogram and sub-block color and texture features provide good performance in terms of retrieval results. However, different methods and image features could be combined to provide better image retrieval results, particularly on large image databases.

- This combination of different techniques and features in the spatial and frequency domains provide our future work directions as to how to get appropriate features for image matching and retrieval.

- Region-based features can be further improved to resolve the background complexity problem using histogram refinement method.

- The local features of the image play important roles in object recognition and provide effective local object information in images for retrieval. Hence, in future to further improve the retrieval performance local features can be included.

- The hybrid CBIR system should be dynamic to add any new image in the feature database at run time without recreation of the whole feature database.

- The global features should be combined with local information to perform object-based retrieval to further enhance the retrieval.

- It is very important to bridge the semantic gap between the low level features and the high level semantics of the image. However, using currently relevant feedback techniques and linking the low level visual features with annotations the semantic gap is still a challenging task for CBIR.

LIST OF PUBLICATIONS

JOURNALS

1. **Fazal Malik** and Baharum Baharudin, "Mean and standard deviation features of color histogram using Laplacian filter for content-based image retrieval," *Journal of Theoretical and Applied Information Technology* (JATIT), vol. 34, no. 1, pp. 1-7, 2011.
2. **Fazal Malik** and Baharum Baharudin, "Median and Laplacian Filters based Feature Analysis for Content Based Image Retrieval Using Color Histogram Refinement Method," *Journal of Applied Sciences*, vol. 12, no. 5, pp. 416-427, 2012.
3. **Fazal Malik** and Baharum Baharudin, "Features Analysis for Content-Based Image Retrieval Based on Color Moments," *Research Journal of Applied Sciences, Engineering and Technology*, vol. 4, no. 9, pp. 1215-1224., 2012.
4. **Fazal Malik** and Baharum Bin Baharudin, "Feature Analysis of Quantized Histogram Color Features for Content-Based Image Retrieval Based on Laplacian Filter," *International Proceedings of Computer Science and Information Technology* vol. 34, pp. 44-49, 2012.
5. **Fazal Malik** and Baharum Baharudin, "The Statistical Quantized Histogram Texture Features Analysis for Image Retrieval Based on Median and Laplacian Filters in the DCT Domain," *The International Arab Journal of Information Technology*, vol. 10, no. 6, 2012.
6. **Fazal Malik** and Baharum Baharudin, “Analysis of Distance Metrics in Content-Based Image Retrieval Using Statistical Quantized Histogram Texture Features in the DCT Domain, *Journal of King Saud University - Computer and Information Sciences* (2012), doi: http://dx.doi.org/10.1016/j.jksuci.2012.11.004
7. **Fazal Malik** and Baharum Baharudin, "Effective image retrieval based on an experimental combination of texture features and comparison of different histogram quantization in the DCT domain," *The International Arab Journal of Information Technology*, vol. 11, no. 3, 2013.

8. **Fazal Malik** and Baharum Baharudin, “Analysis of the different combinations of the texture features in the spatial and frequency domains for content-based image retrieval”, *Journal of King Saud University - Computer and Information Sciences* (Accepted).

9. **Fazal Malik** and Baharum Baharudin, "Analysis of Statistical Texture Features Based on Histogram Quantization Bins for Image Retrieval in DCT Domain," *Journal of Computers, Academy Publisher,* (Submitted)

CONFERENCES

10. **Fazal Malik** and Baharum Baharudin, "Efficient image retrieval based on texture features," *IEEE National Postgraduate Conference (NPC 2011),* pp. 1-6, 2011.

11. **Fazal Malik**, Baharum Baharudin, and Kifayat Ullah, "Efficient Image Retrieval Based on Quantized Histogram Texture Features in DCT Domain," in *IEEE Frontiers of Information Technology (FIT),* 2011, pp. 89-94.

12. **Fazal Malik** and Baharum Baharudin, "Quantized histogram color features analysis for image retrieval based on median and Laplacian filters in DCT domain," *IEEE International Conference on Innovation Management and Technology Research (ICIMTR),* pp. 624-629, 21-22 May 2012.

13. **Fazal Malik** and Baharum Baharudin, "Effective content-based image retrieval: Combination of quantized histogram texture features in the DCT domain," *IEEE International Conference on Computer & Information Science (ICCIS),* vol. 1, pp. 425-430, 12-14 June 2012.

REFERENCES

Alnihoud, J. (2012). "Content-Based Image Retrieval System Based on Self Organizing Map, Fuzzy Color Histogram and Subtractive Fuzzy Clustering." The International Arab Journal of Information Technology 9(5): 452-458.

An, Y., W. Rasheed, S. Park and J. Park (2011). "Feature Extraction Through Generalization of Histogram Refinement Technique for Local Region-Based Object Attributes." International Journal of Imaging Systems and Technology - Multimedia Information Retrieval 21(3): 298–306.

Anjum, M. A. and M. Y. Javed (2007). "Multiresolution and Varying Expressions Analysis of Face Images for Recognition." Information Technology Journal 6(1): 57-65.

Arevalillo-Herráeza, M., J. Domingob and F. J. Ferria (2008). "Combining similarity measures in content-based image retrieval." Pattern Recognition Letters 29(16): 2174-2181.

Baaziz, N., O. Abahmane and R. Missaoui (2010). "Texture feature extraction in the spatial-frequency domain for content-based image retrieval." Department of Computer Science and Engineering, Université du Québec en Outaouais: 1-19.

Bach, J., C. Fuller, A. Gupta, A. Hampapur, B. Gorowitz, R. Humphrey, R. Jain and C. Shu. (1996). "Virage image search engine: an open framework for image management, pages." In Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases IV, San Jose, CA: 76–87.

Bae, H.-J. and S.-H. Jung (1997). Image retrieval using texture based on DCT. Proceedings of 1997 International Conference on Information, Communications and Signal Processing (ICICS) 2: 1065-1068.

Bai, C., K. Kpalma and J. Ronsin (2012). "Color textured image retrieval by combining texture and color features." European Signal Processing Conference (EUSIPCO-2012), Bucharest, Romania: 170-174.

Balamurugan, V., P. Anandhakumar, S. Nivaneethakrishnan and N. Venkatesan (2010). "An Effective Color and Texture Feature Based Image Retrieval Technique For Medical Images Using Relevance Feedback." International Conference on Information Science and Applications (ICISA '10): 4-9.

Bannour, H., L. Hlaoua and B. Ayeb (2009). "Survey of the Adequate Descriptor for Content-Based Image Retrieval on the Web : Global versus Local Features." Conférence en Recherche d'Information et Applications (CORIA '09): 445-456.

Belongie, S., J. Malik and J. Puzicha (2002). "Shape matching and object recognition using shape contexts." IEEE Trans. Pattern Anal. Mach. Intell. 24(4): 509–522.

Berretti, S., A. D. Bimbo and P. Pala (2000). "Retrieval by shape similarity with perceptual distance and effective indexing." IEEE Trans. Multimed 2(4): 225–239.

Biederman, I. (1987). " Recognition-by-components: A theory of human image understanding " Psychological Review 94(2): 115--147.

Bimbo, A. D. and P. Pala (1997). "Visual image retrieval by elastic matching of user sketches." IEEE Trans. Pattern Anal. Mach. Intell 19(2): 121–132.

Cha, S.-H. (2007). "Comprehensive Survey on Distance/Similarity Measures between Probability Density Functions." International journal of mathematical models and methods in applied sciences 4(1): 300-307.

Chakravarti, R. and X. Meng (2009). A Study of Color Histogram Based Image Retrieval. Sixth International Conference on Information Technology: New Generations (ITNG '09). 1323-1328.

Chun, Y. D. (2005). "Content-based image retrieval using multiresolution color and texture features." Ph.D. thesis, Dept. Elect. Eng., Kyungpook National Univ., Daegu, Korea, [Online]. Available: http://vcl.knu.ac.kr/phd_thesis/ydchun.pdf.

Ciaccia, P., M. Patella and P. Zezula (1997). "M-Tree: An efficient access method for similarity search in metric spaces." In Proceedings of the International Conference on Very Large Databases (VLDB).

Datta, R., D. Joshi, J. li and J. Z. Wang (2008). "Image Retrieval: Ideas, Influences, and Trends of the New Age." ACM Computing Surveys, Article-5 40(2): 5:1-5:60.

Demigny, O. (2002). "On Optimal Liear Filtering for Edge Detection." IEEE TRANSACTIONS ON IMAGEP POCESSING.

Deng, Y., B. S. Manjunath and C. Kenney (2001). " An efficient color representation for image retrieval." IEEE Transactions on Image Processing 10(1): 140–147.

Deselaers, T., D. Keysers and H. Ney (2005). Discriminative Training for Object Recognition Using Image Patches. Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2, IEEE Computer Society: 157-162.

Deselaers, T., D. Keysers and H. Ney (2007). "Features for image retrieval: An experimental comparison." Information Retrieval 11(2): 77-107.

Deselaers, T., D. Keysers and H. Ney (2008). "Features for image retrieval:an experimental comparison." Information Retrieval 11(2): 77-107.

Do, M. N. and M. Vetterli (2002). "Wavelet-Based Texture Retrieval Using Generalized Gaussian Density and Kullback-Leibler Distance." IEEE Trans. Image Processing 11(2): 146--158.

Do, M. N. and M. Vetterli (2002). "Wavelet-Based texture retrieval using generalized Gaussian density and Kullback-Leibler distance." IEEE Trans. Image Process. 11(2): 146–158.

Dubey, R. S., R. Choubey and J. Bhattacharjee (2010). "Multi Feature Content Based image Retrieval." International Journal on Computer Science and Engineering (IJCSE) 2(6): 2145-2149.

Eakins, J. and M. Graham (1999). "Content-based Image Retrieval:A report:39, to the JISC Technology Applications Programme, Joint Information Systems Committee, University of Manchester, UK."

Fagin, R. (1997). "Combining fuzzy information from multiple systems." In Proceedings of the ACM-SIGACTSIGMOD-SIGART Symposium on Principles of Database Systems (PODS).

Fan, Y. and R. Wang (2002). "An image retrieval method using DCT features." Journal of Computer Science and Technology 17(6): 865-873.

Feig, E. (1990). "A fast scaled DCT algorithm." In Proc. SPIE Image Processing Algorithms and Techniques, 1244: 2–13.

Feng, G. and J. Jiang (2003). "JPEG compressed image retrieval via statistical features." Pattern Recognition 36(4): 977–985.

Flickner, M., H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker (1995). "Query by image and video content: The QBIC system." IEEE Comput 28(9): 23–32.

Goldberger, J., S. Gordon and H. Greenspan (2003). An efficient image similarity measure based on approximations of KL-divergence between two gaussian mixtures. Ninth IEEE International Conference on Computer Vision (ICCV ’03), 487–493.

Gong, Y., C. H. Chuan and G. Xiaoyi (1996). "Image indexing and retrieval using color histograms " Multimed. Tools Appl. 2: 133–156.

Gonzalez, R. C., R. E. Woods and S. L. Eddins (2004). Digital Image Processing using MATLAB, 2nd Edition, Pearson Prentice Hall, 2nd Edition: 100-103.

Gudivada, V. N. and V. V. Raghavan (1995). "Content based image retrieval systems." Computer 28(9): 18-22.

Hafner, J., H. S. Sawhney, W. Equitz, M. Flickner and W. Niblack (1995). "Efficient Color Histogram Indexing for Quadratic Form Distance Functions." IEEE Trans. Pattern Anal. Mach. Intell. 17(7): 729-736.

Han, J. and K. K. Ma (2002). "Fuzzy colour histogram and its use in color image retrieval." IEEE Trans. Image Process. 11(8): 944–952.

Haralick, R. (1979). "Statistical and structural approaches to texture." Proc. IEEE 67(5): 786–804.

Haralick, R. M. (1979). "Statistical and structural approaches to texture." Proceedings of the IEEE 67(5): 786 - 804.

Haralick, R. M., K. Shanmugam and I. Dinstein (1973). "Textural features for image classification." IEE Tnansaction on Systems, Man, Cybernetics SMC-3(6): 610-621.

Harris, C. and M. Stephens (1988). "A combined corner and edge detector}." In Proc. of Fourth Alvey Vision Conference: 147--151.

Heng, Y. and W. Qing (2008). A novel local feature descriptor for image matching. Multimedia and Expo, 2008 IEEE International Conference on: 1405-1408.

Hideyuki, T., M. Shunji and Y. Takashi (1978). "Textural features corresponding to visual perception." IEEE Transactions on Systems, Man & Cybernetics 8(6): 460-473.

Hiremath, P. S. and J. Pujari (2007). Content Based Image Retrieval Using Color, Texture and Shape Features. Proceedings of the 15th International Conference on Advanced Computing and Communications, IEEE Computer Society: 780-784.

Huang, J., S. R. Kumar, M. Mitra, W.-J. Zhu, R. Zabih and (1997). "Image indexing using color correlograms." IEEE Conference on Computer Vision and Pattern Recognition (CVPR ’97): 762–768.

Jacob, I. J. and K. G. Srinivasagan (2013). "Image Retrieval Using Local Object Appearance and Spatial Color Layout." International Journal of Computer Science and Electronics Engineering (IJCSEE) 1(3): 425-429.

Jafari-Khouzani, K. and H. Soltanian-Zadeh (2005). "Radon transform orientation estimation for rotation invariant texture analysis." IEEE Transactions on Pattern Analysis and Machine Intelligence 27(6): 1004-1008.

Jain, A. and F. Farrokhnia (1990). "Unsupervised texture segmentation using Gabor filters. ." Proceedings of the International Conference on Systems, Man and Cybernetics.

Jain, A. and G. Healey (1998). "A multiscale representation including opponent colour features for texture recognition." IEEE Transactions on Image Processing 7(1): 124-128.

Jeong, J. (1997). The JPEG standard. Digital consumer electronics handbook, McGraw-Hill, Inc.: 91-99.

Jia, L. and J. Z. Wang (2003). "Automatic Linguistic Indexing of Pictures by a statistical modeling approach." IEEE Transactions on Pattern Analysis and Machine Intelligence 25(9): 1075-1088.

Jin, C. (2009). A Statistical Image Retrieval Method Using Color Invariant. Proceedings of the 2009 Sixth International Conference on Computer Graphics, Imaging and Visualization, IEEE Computer Society: 355-360.

John, C. (1986). "A Computational Approach to Edge Detection." Pattern Analysis and Machine Intelligence, IEEE Transactions PAMI-8(6): 679-698.

Kato, T., T. Kurita and H. Shimogaki (1991). "Intelligent Visual Interaction with Image Database Systems: Toward the Multimedia Personal Interface." Journal of information processing 14(2): 134-143.

Kavitha, C., B. P. Rao and A. Govardhan (2011). "An efficient content based image retrieval using color and texture of image subblocks." International Journal of Engineering Science and Technology (IJEST) 3(2): 1060-1068.

Kekre, H. B. and D. Mishra (2011). "Content Based Image Retrieval Using Full Haar Sectorization." International Journal of Image Processing (IJIP) 5(1): 1-12.

Kekre, H. B. and K. Patil (2009). "Standard Deviation of Mean and Variance of Rows and Columns of Images for CBIR." International Journal of Computer and Information Engineering 3(1): 1-4.

Kekre, H. B. and K. Sonawan (2012). "STATISTICAL PARAMETERS BASED FEATURE EXTRACTION USING BINS WITH POLYNOMIAL TRANSFORM OF HISTOGRAM." International Journal of Advances in Engineering & Technology, July 2012. 4(1): 510-524.

Kodituwakku, S. R. and S. Selvarajah (2010). "Comparison of Color Features for Image Retrieval." Indian Journal of Computer Science and Engineering 1(3): 207-211.

Kosch, H. and M. Maier (2009). "Content-Based Image Retrieval Systems - Reviewing and Benchmarking." The 9th Workshop on Multimedia Metadata (WMM), Toulouse.

Laine, A. and J. Fan (1993). "Texture classication by wavelet packet signatures." IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11): 1186-1191.

Latecki, L. J. and R. Lakamper (2000). "Shape similarity measure based on correspondence of visual parts." IEEE Trans. Pattern Anal. Mach. Intell. 22(10): 1185–1190.

Lay, J. A. and L. Guan (1999). Image retrieval based on energy histograms of the low frequency DCT coefficients. Proceedings, IEEE International Conference on Acoustics, Speech, and Signal Processing 6: 3009-3012.

Li, J., R. M. Gray and R. A. Olshen (2000). "Multiresolution image classification by hierarchical modeling with two dimensional hidden Markov models." IEEE Trans. Inf. Theory 46(5): 1826–1841.

Li, J., A. Najmi and R. M. Gray (2000). "Image Classification by a Two-Dimensional Hidden Markov Model." IEEE Transaction Signal Processing 48(2): 517–533.

Li, X. (2003). "Image retrieval based on perceptive weighted color blocks." Pattern Recogn. Lett. 24(12): 1935–1941.

Liapis, S. and G. Tziritas (2004). "Color and texture image retrieval using chromaticity histograms and wavelet frames." IEEE Transactions on Multimedia 6(5): 676-686.

Liapis, S. and G. Tziritas (2004). "Color and texture image retrieval using chromaticity histograms and wavelet frames." IEEE Transactions on Multimedia 6(5): 676–686.

Liu, Y., D. Zhang, G. Lu and W.-Y. Ma (2007). "A survey of content-based image retrieval with high-level semantics." Pattern Recogn. 40(1): 262-282.

Loncaric, S. (1998). "A Survey of Shape Analysis Techniques." Pattern Recognition 31(8): 983--1001.

Lu, G. and A. Sajjanhar (1999). "Region-based shape representation and similarity measure suitable for content-based image retrieval." Multimedia Systems 7(2): 165-174.

Lu, Z.-M., S.-Z. Li and H. Burkhardt (2006). "A content-based image retrieval scheme in JPEG compressed domain." International Journal of Innovative Computing, Information and Control 2(4): 831—839.

Ma, W.-Y. and B. Manjunath (1998). "Texture thesaurus for browsing large aerial photographs. ." J. Amer. Soc. Inf. Sci. 49(7): 633–648.

Ma, W.-Y. and B. S. M. Netra (1999). "A toolbox for navigating large image databases." Multimedia Systems 7(3): 184–198.

Malik, J. and P. Perona (1990). "Preattentive texture discrimination with early vision mechanisms." J. Optical Soc. Amer. A 7(5): 923–932.

Mandal, M. K., F. Idris and S. Panchanathan (1999). "A Critical Evaluation of Image and Video Indexing Techniques in the Compressed Domain." Image and Vision Computing 17(7): 513-529.

Manjunath, B. and W.-Y. Ma (1996). "Texture features for browsing and retrieval of image data. ." IEEE Trans. Pattern Anal. Mach. Intell. 18(8): 837–842.

Manjunath, B. S., J.-R. Ohm, V. V. Vasudevan and A. Yamada (2001). "Color and texture descriptors." IEEE Trans. Circ. Syst. Video Technol. 11(6): 703–715.

Mehrotra, R. and J. E. Gary (1995). "Similar-Shape retrieval in shape data management." IEEE Comput. 28(9): 57–62.

Mikolajczyk, K. and C. Schmid (2004). "Scale and affine invariant interest point detectors." Intl. J. Comput. Vision 60(1): 63–86.

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir and L. V. Gool (2005). "A comparison of affine region detectors." International Journal of Computer Vision 65(1-2): 43–72.

Mohamed, A., F. Khellfi, W. Ying, J. Jianmin and S. Ipson (2009). An efficient Image Retrieval through DCT Histogram Quantization. International Conference on CyberWorlds (CW '09): 237-240.

Muller, H., N. Michoux, D. Bandon and A. Geissbuhler (2004). " A review of content-based image retrieval systems in medical applications—clinical benefits and future directions,." International Journal of Medical Informatics 73: 1-23.

Murala, S., A. B. Gonde and R. P. Maheshwari (2009). "Color and Texture Features for Image Indexing and Retrieval." IEEE International Advance Computing Conference (IACC 2009): 1411-1416.

Mustaffa, M. R., F. Ahmad and R. Wirza (2008). "Content-based image retrieval based on color-spatial features." Malays. J. Comput. Sci 21(1): 1-12.

Netra, W. Y. M. (1997). "A Toolbox for Navigating Large Image Databases." PhD thesis, Dept. of Electrical and Computer Engineering, University of California at Santa Barbara.

Nezamabadi-pour, H. and S. Saryazdi (2005). Object based Image Indexing and Retrieval in DCT Domain using Clustering Techniques. Proceedings of World Academy of Science Engineering and Technology. (3).

Niblack, W., R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, C. Faloutsos and G. Taubin. (1993). "The qbic project: Quering images by content using color, texture, and shape." In Poceedings of the SPIE Conference on Storage and Retrieval for Image and Video atabases: 73–187.

Ortega, M., Y. Rui, K. Chakrabarti, S. Mehrotra and T. S. Huang (1997). " Supporting similarity queries in MARS." In Proceedings of the 5th ACM International Multimedia Conference, Seattle, Washington: 403–413.

Park, J., Y. An, G. Kang, W. Rasheed, S. Park and G. Kwon (2008). "Defining a new feature set for content-based image analysis using histogram refinement." International Journal of Imaging Systems and Technology - Multimedia Information Retrieval 18(2-3): 86-93.

Park, J., Y. An and P. Kim (2007). Quantization of colors using median of pixels for color correlogram. Proceedings of the 2nd international conference on Technologies for e-learning and digital entertainment. Hong Kong, China, Springer-Verlag: 723-729.

Park, M., J. S. Jin and L. S. Wilson (2002). Fast Content-Based Image Retrieval Using Quasi-Gabor Filter and Reduction of Image Feature Dimension. Proceedings of the Fifth IEEE Southwest Symposium on Image Analysis and Interpretation, IEEE Computer Society: 178–182.

Park, S.-S., K.-K. Seo and D.-S. Jang (2005). "Expert system based on artificial neural networks for content-based image retrieval." Expert Syst. Appl. 29(3): 589-597.

Park, S.-S., Y.-G. Shin and D.-S. Jang (2010). "A novel efficient technique for extracting valid feature information." Expert Systems with Applications: An International Journa. 37(3): 2654-2660.

Partio, M. (2002). Content-based Image Retrieval using Shape and Texture Attributes, Tampere University of Technology, Tampere, Finland. Master of Science Thesis, p-70.

Paschos, G., I. Radev and N. Prabakar (2003). " Image content-based retrieval using chromaticity moments " IEEE Trans. Knowl. Data Eng. 15(5): 1069–1072.

Pass, G. and R. Zabih (1996). Histogram refinement for content-based image retrieval. 3rd IEEE Workshop on Applications of Computer Vision (WACV '96) 96-102.

Petrakis, E. G. M., A. Diplaros and E. Milios (2002). "Matching and retrieval of distorted and occluded shapes using dynamic programming." IEEE Trans. Pattern Anal. Mach. Intell. 24(4): 509–522.

Portilla, J. and E. Simoncelli (2000). "A parametric texture model based on joint statistics of complex wavelet coefficients. ." Int. J. Comput. Vision 40(1): 49–71.

Pun, C.-M. and M.-C. Lee (2003). "Log-polar wavelet energy signatures for rotation and scale invariant texture classication." IEEE Transactions on Pattern Analysis and Machine Intelligence 25(3): 590-603.

Qiu, G. (2003). "Color Image Indexing Using BTC." IEEE TRANSACTIONS ON IMAGE PROCESSING 12(1): 93- 01

Rani, P. M. N. and A. Saravanan (2011). "Effective Semantics Intensive Image Retrieval using Fast Indexing Technique." Research Journal of Information Technology 3(2): 114-122.

Roumi, M. (2009). "Implementing Texture Feature Extraction Algorithms on FPGA, M.Sc Thesis; Computer Engineering, Mekelweg 4, 2628 CD, Delft University of Technology, The Netherlands.".

Rubner, Y., C. Tomasi and L. J. Guibas (2000). "The earth mover’s distance as a metric for image retrieval." Int. J. Comput. Vision 40: 99–121.

Rui, Y., T. Huang and S. Andmehrotra (1997). "Content-based image retrieval with relevance feedback inMars." In Proceedings of the IEEE International Conference on Image Processing (ICIP)

Schaffalitzky, F. and A. Zisserman (2001). "Viewpoint invariant texture matching andwide baseline stereo." In Proceedings of the IEEE International Conference on Computer Vision (ICCV).

Schulz, J. (2007). "Algorithms – Similarity." http://www.code10.info/index.php?option=com_content&view=article&id=49:article_canberra-distance&catid=38:cat_coding_algorithms_data-similarity&Itemid=57.

Selvarajah, S. and S. R. Kodituwakku (2011). "Analysis and Comparison of Texture Features for Content Based Image Retrieval." International Journal of Latest Trends in Computing 2(1): 108-113.

Semler, L. and L. Dettori. (2006). " Curvelet-based texture classication of tissues in computed tomography." In Proceeding of IEEE International Conference on Image Processing, ICIP: 2165-2168.

Sergyan, S. (2008). Color histogram features based image classification in content-based image retrieval systems. 6th International Symposium on Applied Machine Intelligence and Informatics(SAMI 2008): 221-224.

Shahbahrami, A., D. Borodin and B. Juurlink (2008). "Comparison Between Color and Texture Features for Image Retrieval." 19th Annual Workshop on Circuits, Systems and Signal Processing.

Shan, Z. and Y.-X. Liu (2009). Research on JPEG Image Retrieval. 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom '09) 16: 1-4.

Sharma, N., P. Rawat and J. Singh (2011). "Efficient CBIR Using Color Histogram Processing." Signal & Image Processing : An International Journal(SIPIJ) 2(1): 94-112.

Shen, S. (2013). "http://www.csee.umbc.edu/~pmundur/courses/CMSC691M-04/sharon-DWT.ppt."

Singha, M. and K. Hemachandran (2012). "Content Based Image Retrieval using Color and Texture." Signal & Image Processing : An International Journal (SIPIJ) 3(1): 39-57.

Smeulders, A. W. M., M. Worring, S. Santini, A. Gupta and R. Jain (2000). "Content-based image retrieval at the end of the early years." IEEE Transactions on Pattern Analysis and Machine Intelligence 22(12): 1349-1380.

Smith, J. R. and S.-F. Chang (1997). "Querying by color regions using the VisualSEEk content-based visual query system." In M. T. Maybury, editor, Intelligent Multimedia Information Retrieval. AAAI Press.

Soman, S., M. Ghorpade, V. Sonone and S. Chavan (2011). "Content Based Image Retrieval Using Advanced Color and Texture Features." International Journal of Computer Applications® (IJCA): 10-14.

Squire, D. M., W. Müller, H. Müller and J. Raki (1999). "Content-Based Query of Image Databases, Inspirations From Text Retrieval: Inverted Files, Frequency-Based Weights and Relevance Feedback." PATTERN RECOGNITION LETTERS: 143--149.

Stricker, M. and A. Dimai (1996). "Color indexing with weak spatial constraints." in: SPIE Conference on Storage and Retrieval for Image and Video Databases IV, SPIE Publishing, Washington,: 29–40.

Stricker, M. and M. Orengo (1995). "Similarity of color images." in Proc. SPIE Storage and Retrieval for Image and Video Databases.

Stricker, M. A. and M. Orengo (1995). "Similarity of color images." SPIE, Storage and Retrieval for Image and Video Databases: 381-392.

Stricker, M. A. and M. Orengo (1995). "Similarity of color images." SPIE 2420: 381-392.

Sudhakar, R., K. R. Krishnan and S. Muthukrishnan (2011). A hybrid approach to content based image retrieval using visual features and textual queries. Advanced Computing (ICoAC), 2011 Third International Conference on: 241-247.

Suresh, P., R. Sundaram and A. Arumugam (2008). Feature Extraction in Compressed Domain for Content Based Image Retrieval. IEEE International Conference on Advanced Computer Theory and Engineering, ICACTE , 2008.: 190-194.

Swain, M. J. and D. H. Ballard (1991). " Color indexing." International Journal of Computer Vision 7(2): 11-32.

Thawari, P. B. and N. J. Janwe (2011). "CBIR BASED ON COLOR AND TEXTURE." International Journal of Information Technology and Knowledge Management 4(1): 129-132.

Thomas, D., K. Daniel and N. Hermann (2008). "Features for image retrieval: An experimental comparison." Inf. Retr. 11(2): 77–107.

Tsai, T., Y.-P. Huang and T.-W. Chiang (2006). Dominant Feature Extraction in Block-DCT Domain. IEEE International Conference on Systems, Man and Cybernetics(SMC '06). 5: 3623-3628.

Tuceryan, M. and A. K. Jain (1993). "Texture analysis." The Handbook of Pattern Recognition and Computer Vision (2nd Edition), by C. H. Chen, L. F. Pau,

P. S. P. Wang (eds.), Singapore: World Scientific: 235-276.

Unser, M. (1995). "Texture classification and segmentation using wavelet frames." IEEE Trans. Image Process. 4(11): 1549–1560.

Vadivel, A., A. K. Majumdar and S. Sural (2003). "Performance comparison of distance metrics in content-based image retrieval applications." International Conference on Information Technology (CIT): 159-164.

Vadivel, A., A. K. Majumdar and S. Sural (2004). "Characteristics of weighted feature vector in content-based image retrieval applications." Proceedings of IEEE International conference on Intelligent Sensing and Information Processing: 127–132.

Vailaya, A., A. Jain and H. J. Zhang (1998). "On Image Classification: City Images vs. Landscapes." Pattern Recognition 31: 1921--1935.

Veltkamp, R. C. and M. Tanase (2002). "Content based image retrieval systems: A survey." Tech. Rep. No. UU-CS-2000-34. Department of Computer Science, Utrecht University, Utrecht, the Netherlands.

Vilnrotter, F. M., R. Nevatia and K. E. Price (1986). "Structural analysis of natural textures." IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI- 8(1): 76 - 89.

Wang, J. Z., J. Li and G. Wiederhold (2001). "SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries." EEE Trans. on Pattern Analysis and Machine Intelligence 23(9): 947-963.

Wang, L., Y. Zhang and J. Feng (2005). "On the Euclidean Distance of Images." IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8): 1334-1339.

Webe, M., M. Welling and P. Perona (2000). "Unsupervised learning of models for recognition." In Proceedings of the European Conference on Computer Vision (ECCV).

Wilson, R. and E. Hancock (1997). "Structural matching by discrete relaxation." IEEE Trans. Pattern Anal. Mach. Intell 19(6): 634–648.

Wolfson, H. and I. Rigoutsos (1997). "Geometric hashing: An overview." IEEE Trans. Comput. Sci. Eng. 4(4): 10–21.

Won, C. S., D. K. Park and S.-J. Park (2002). "Efficient Use of MPEG-7 Edge Histogram Descriptor." ETRI Journal, Volume 24, Number 1, February 2002 Electronics and Telecommunications Research Institute 24(1): 23-30.

Wu, Y., Q. Tian and T. S. Huang (2000). "Discriminant-EM algorithm with application to image retrieval." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Zhang, D. and G. Lu (2001). "A Comparative Study on Shape Retrieval Using Fourier Descriptors with Different Shape Signatures." International Conference on Intelligent Multimedia and Distance Education (ICIMADE01): 1-9.

Zhang, D. and G. Lu (2003). Evaluation of similarity measurement for image retrieval. IEEE International Conference Neural Networks & Signal Processing 2: 928-931.

Zhang, D. and G. Lu (2004). "Review of shape representation and description techniques." Pattern Recognition 37(1): 1-19.

Zhao, H., P. Kim and J. Park (2009). Feature Analysis Based on Edge Extraction and Median Filtering for CBIR. 11th International Conference on Computer Modelling and Simulation (UKSIM '09): 245-249.

Zhong, D. and I. Defee (2005). "DCT histogram optimization for image database retrieval." Pattern Recogn. Letters. 26(14): 2272-2281.