MULTI FEATURE CBIR MODEL FOR TRADEMARK IMAGE DISTINCTIVENESS

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Pratima Soni

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Program:
Software Engineering

February 2019

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Multi Feature CBIR Model For Trademark Image Distinctiveness

By

Pratima Soni

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Kendall Nygard
Chair

Dr. Simone Ludwig

Dr. Sangita Sinha

Approved:

03/27/2019	Dr. Kendall Nygard
Date	Department Chair

ABSTRACT

A Content-Based Image Retrieval (CBIR) system can help in detecting trademark infringement. This paper describes an improvement and extension of previous work for the existing Universal Model for a CBIR system, which only uses texture along with color to produce a reliable and capable system.

To enhance the security and efficiency of the CBIR based Universal Model, the extension is a Multi Feature CBIR Model that can utilize many parameters including color, texture, and shape with the facility of choosing a suitable and flexible set of features. The supported selection techniques give priority to individual features and help increase the efficiency of the system. I am going to provide write-ups with the proven computational work over the existing feature methods of the Universal Model, which becomes the foundation for the Multi-Feature CBIR Model. In comparison to the Universal Model, the Multi-Feature CBIR Model can obtain higher recall and precision values.

ACKNOWLEDGMENTS

Sincere gratitude is hereby extended to Dr. Kendall Nygard who never ceased in helping until this report paper was structured and finalized. I appreciate the time, support, guidance, and patience for the development and completion of this research project. He has always motivated me and helped me at every step starting from writing the proposal to finalize the report. He always answered all my questions with detailed and precise guidance and feedback I needed.

I am also grateful and appreciate Dr. Sangita Sinha, Professor of Chemistry and Biochemistry for her consideration and taking out time from her busy schedule to be a part of my supervisory committee and showing interest in my research work.

I would also like to thank Dr. Simone Ludwig, Professor of Computer Science and Operations Research at North Dakota State University, for her time and to be a member of my supervisory committee. Also, for her continuous guidance and the compassion she had shown throughout my Master degree program that helped me immensely to achieve my goals.

A special thanks to the faculty of the Computer Science department for all their help and support that was necessary at all the time throughout my program.

TABLE OF CONTENTS

ABSTRACT iii

ACKNOWLEDGMENTS iv

LIST OF TABLES vii

LIST OF FIGURES viii

LIST OF ABBREVIATIONS x

1. INTRODUCTION 1

1.1. Overview 1

1.1.1. Use of Image 1

1.1.2. Image Retrieval Problem 2

1.1.3. Visual Content 3

1.2. CBIR System 3

1.2.1. Feature 5

1.2.2. Trademark 7

1.3. Features Description 8

1.3.1. The Color Feature 8

1.3.2. The Spatial Layout of the Color Feature 11

1.3.3. The Texture Feature 13

1.3.4. The Shape Feature 14

1.4. Research Objective and Motivation 15

1.5. The Layout of the Paper 16

2. PREVIOUS WORK AND LITERATURE REVIEW 18

2.1. The Formulation of The Idea 18

2.2. Conclusion 22

3. PROPOSED MULTI FEATURE CBIR MODEL 23

3.1. Proposed Method 24

3.1.1. The Technique Used for Color Feature 25

3.1.2. The Technique Used for Spatial Layout 29

3.1.3. The Technique Used for Texture 31

3.1.4. The Technique Used for Shape 36

3.2. Similarity Calculation 38

3.3. Combining the Features 39

3.4. Database 40

4. RESULT AND DISCUSSION 42

4.1. Performance 42

4.2. Recall-Precision Graph 43

4.3. Average Response Time 45

4.4. Graphical User Interface 47

4.5. System Specification 50

4.6. Hardware and Software Requirement 50

4.7. Result Snapshots 51

5. CONCLUSION AND FUTURE WORK 53

5.1. The Conclusion of the Work 53

5.2. Future Work 53

REFERENCES 55

LIST OF TABLES

Table Page

1: Recall and Precision Value of the System 20

2: Comparison of Techniques........ 22

3: Precision and Recall Values in % 44

4: System Requirements Tabulation........ 50

LIST OF FIGURES

Figure Page

1: Principle of CBIR System 4

2: Flow Chart of CBIR System 6

3: Examples of Trademark Images 7

4: RGB Color Model 9

5: RGB Coordinate System 9

6: HSI Color Model 10

7: Same Color Histogram for All Different Image but Different Color Layout 11

8: Region Distribution for Computing Histograms 12

9: Different Texture Patterns 13

10: Boundary Based and Region Based Shape Representation 15

11: Query Images Used In the Universal Model. 19

12: Block Diagram of the Multi-Feature CBIR Model 25

13: Sample Trademark Image 26

14: Histogram of Hue 28

15: Histogram of Saturation 28

16: Histogram of Value 29

17: Circular Region Distribution of the Proposed Method 30

18: Computing the Gray Level Co-occurrence Matrix 32

19: Tree-Structured Wavelet Transform 34

20: Levels of Decomposition 34

21: Decomposition of an Image into Sub-Bands 35

22: Edge Histogram Obtained for a Trademark Image 36

23: Precision Recall Graph of the Multi-Feature CBIR Model 44

24: Snapshot of GUI (1) 47

25: Snapshot of GUI (2) 49

26: Snapshot of Result (1) 51

27: Snapshot of Result (2) 51

28: Snapshot of Result (3) 52

29: Snapshot of Result (4) 52

LIST OF ABBREVIATIONS

CBIR	Content-Based Image Retrieval
CBVIR	Content Based Visual Information Retrieval
QBIC	Query by Image Content
TIR	Trademark Image Retrieval
GLCM	Gray Level Co-occurrence Matrix
GUI	Graphical User Interface
RGB	Red, Green, Blue
HSV	Hue, Saturation, Value
CMY	Cyan, Magenta, Yellow
CMYK	Cyan, Magenta, Yellow, Black

1. INTRODUCTION

In this chapter, the background, and details needed to understand the proposed Multi Feature CBIR Model are described. The problem of interest is presented along with a discussion of research objectives and motivation.

1.1. Overview

The work of this paper introduces a multi-feature Content-Based Image Retrieval (CBIR) model for trademark image distinctiveness, providing improvement and extension over an existing work called the Universal Model for CBIR done by Dr. B.S. Adiga and Dr. S. Nandagopalan and published in World Academy of science, engineering, and technology [1].

The Universal Model for CBIR uses two main features, color and texture to determine the similarity between two images, without including a shape feature. The Universal Model also treats the features with equal priority.

The proposed Multi Feature CBIR Model uses all three main features color, texture, and shape. In the extended model, we have used techniques like a color histogram, a Gray level co-occurrence matrix, and an edge histogram to achieve the feature priority level so that no one feature can dominate on the other while calculating the similarity between two images. This prioritization technique and facility of flexible feature selection are the key concepts proposed to prevent trademark infringement using the Multi-Feature CBIR Model.

1.1.1. Use of Image

We all know that images are widely used. Archaeologists have discovered that ancient humans used images in the form of symbols and paintings that were carved on walls of caves, rocks etc. In today’s world images are used as symbols or trademarks to represent a group, organization, company, etc. [15].

With advances in digital technology, many ways to store images have been developed. If an image stored in a digital form in a database, it is efficient, convenient, safe and easy to access and process. Images are being used in many applications including surveillance, government operations, crime prevention, academics, commerce, journalism, fashion, graphic design, property rights etc. [14].

X-Ray images and scanned images are stored in image databases in the field of medicine, help the doctors in their diagnosis. Image database of criminals, robbery and theft scenes help crime prevention officials to identify criminals and solve cases. A huge database of registered trademarks, logos or symbols stored in digitized form help in claiming property rights.

1.1.2. Image Retrieval Problem

When the image database is small it is easy to select or browse an image. But when the database size increases, access is more difficult. To search for a particular image out of thousands of images in a huge database becomes time-consuming which results in the image retrieval problem [14].

Now we discuss the image retrieval problem more specifically. In image retrieval, the user requests a query image according to a need and applies certain constraints to get appropriate images as output to retrieve from a large database of images [6]. Image retrieval is used in intellectual property rights to ensure that an already issued registered trademark image will not again be issued to another individual or organization [3]. This work needs automation and a system of standards and algorithms to carry out matches, using features like color, and texture [18].

1.1.3. Visual Content

Visual content refers to content that is perceived by the eyes while looking at an image. Visual content cannot be quantitatively measured by any method but can be categorized into different levels that are given below: [7], [26].

- Low Level- It contains the basic features of the image that are visualized at the very first look, like texture pattern, spatial information and the basic colors [20].
- Middle Level- A somewhat higher level of information, like the shape of objects present in the image.
- High Level- This includes complex information that is derived from the image like the expressions and emotions or an idea or story behind the image [9].

1.2. CBIR System

A content-based image retrieval (CBIR), also known by names such as content-based visual information retrieval (CBVIR) or Query by image visual content (QBIC) is the application program of computer vision technique for the image retrieval problem [14]. The main goal of a CBIR system is to find, explore and search digital images in any kind of large database [14], [21], [24]. The term Content-based in CBIR refers to the search based on the actual content of the image, which means the information contained in the image rather than the information provided by the user explicitly [7], [18].

The key component/principle of CBIR (Figure.1) is the image describes the basic agenda of a CBIR system. In content-based image retrieval, the requested query is in the form of a digital image and its characteristics or features are used as the content describing the image [7]. These features are calculated for both a query image and retrieved images [14]. Various algorithms are used for extracting the features which are defined later in the work.

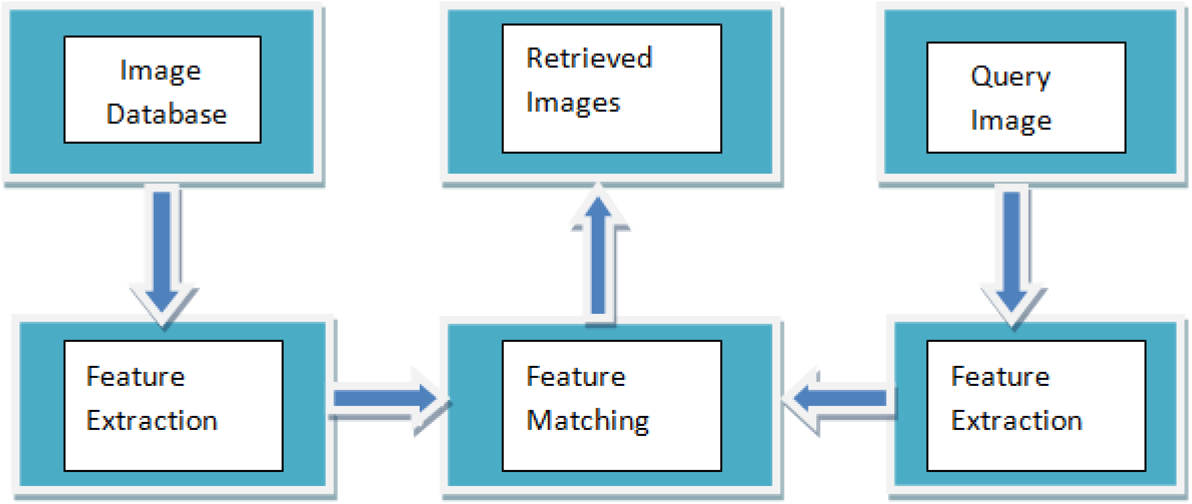


Figure 1: Principle of CBIR System

The feature extraction algorithms work the same for both the query image and database images [6], [30]. After feature extraction of all the images, the system finds all the images from the database that have equivalent feature vectors to the requested query image. After feature matching, the system presents the similar images as output to the requested query image in decreasing order of similarity. Figure.1 above shows the components of a CBIR system [5], [13], [21].

The above CBIR system has an image database on one side that contains different types of images. It could have a large number of images for comparison with the query image. On the other side, the CBIR system has a query image that will be compared with the images saved in the image database. Feature matching is the component that matches the features of the query image with the images saved in the image database. Feature matching can be done by using different types of techniques. After matching the all features of the query image with the images saved in the image database the resultant images will be retrieved.

Exact matching of an image is done to find an exact copy of the query image in a database or to find similar images from a large and diverse database. Similar images do not exactly match the query image but are somewhat similar by some measure [14]. Approximate matching refers to finding the image from the database that is most closely related to the requested query image. Content-Based Image Retrieval search is divided into broad and narrow domains depending upon the database used [14].

Broad domain- Photo collections, Internet images

Narrow domain- Medical images retrieval, Trademark retrieval, Satellite Image retrieval

The broad domain is primarily used for public use purposes. It consists of images having a high variation that needs to be highly interactive [3]. In a narrow domain, the variation in the image content is low and, the content of the image is used for professional purposes. The source of the images is specific and stored locally [7].

1.2.1. Feature

The primitive characteristic present in all images that distinguishes an image from others of the same type is called the distinguishing attribute of the image or image feature [14]. These image features are identified as the normal visual appearance of the image or can be derived by applying calculations and manipulations. Basic features like color usually appear in an image like color with an intensity value for every pixel present in the image while the complex features like texture, include things like histograms and, a frequency spectrum which must be computed [9], [13].

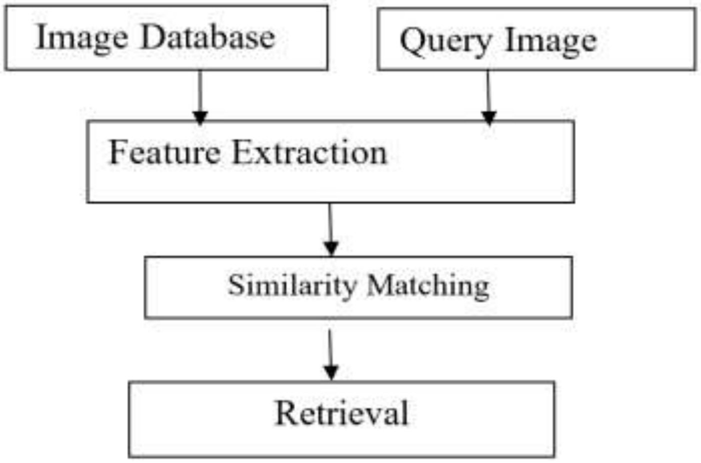


Figure 2: Flow Chart of CBIR System

To distinguish an image, the extraction of image features is essential. This extraction of image features can be done in either online or offline modes for the retrieval process from a database [14]. There is a need for preprocessing phase before feature extraction, to carry out the removal of noise, aligning the image to a particular axis etc. A representation phase includes determining image feature vectors corresponding to every image in the database [5], [13]. Area of CBIR application includes:

- Photograph archives
- Art collections
- Retail catalogs
- Crime prevention
- Medical diagnosis
- The military
- Geographical information and remote sensing
- Architectural and engineering design
- Intellectual property

1.2.2. Trademark

A trademark illustrated in Figure.3 is a unique distinct sign used by any individual, business organization or any judicial legal entity as intellectual property. It can be a logo, an image, word, design, phrase, text, clip-art or a combination of all these entities. A Trademark basically originates from or belongs to a unique source and is used to distinguish one business organization or legal entity and its product and services from another [2].

Since the trademark is a property right belongs to an individual owner that refers to its services and product, if any other entity mistreats or misuses the trademark, then the owner may initiate legal proceedings against the unauthorized entity [3]. Trademark images are classified as device-only mark and word-only mark and device-and-word mark. All depend on the image content of the trademark, such as annotations, symbols, text, tags, descriptions or a combination.

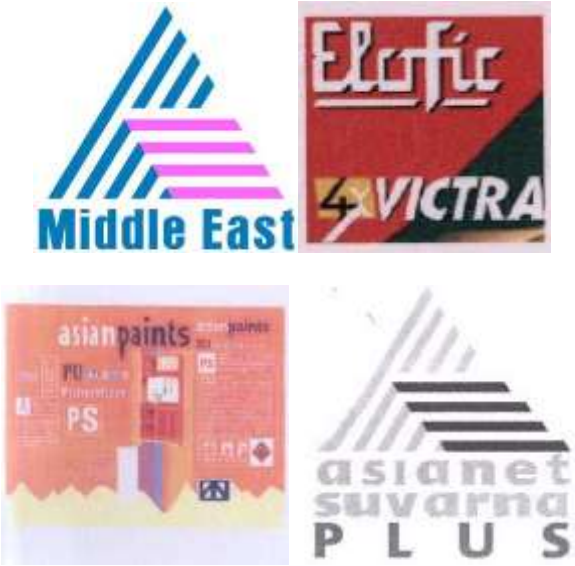


Figure 3: Examples of Trademark Images

The above images are examples of trademark images. These trademark images are related to an organization.

1.3. Features Description

1.3.1. The Color Feature

Color is the most easily recognizable and convenient characteristic of any image or object which is used by humans in daily life to recognize places, objects, people etc. Color is the property of any entity or object that is based on reflection of light and perceived by our eyes. This light is then processed by the brain to identify objects, places, people etc. [11], [13].

1.3.1.1. Color Space

A color space defines a coordinate system within the system. Every color value in the system is then acted as a point within that subspace. Color spaces like CMY (Cyan, Magenta, Yellow), Lab (CIE L*a*b*, CIELAB), CMYK (Cyan, Magenta, Yellow and Black), are rarely used while RGB (red, green, blue), and its transform HSV (Hue, Saturation and value), are used frequently [16], [30]. The Hue is the most obvious characteristic of color, basically, it defines the color like blue, red, pink etc. We can find an infinite different range of hues between the numbers of different colors. For example, between red and yellow, we can find the range for orange hues. Saturation depicts the purity of a color. High saturation shows the richness of color and low saturation colors look dull and grayish. Value shows lightness or darkness of a color. Light colors are sometimes called tints and dark colors shades. The high saturation colors have medium value because light and dark colors are achieved by mixing with white or black color.

Different color models are discussed below:-

The RGB color model-RGB color model is the most commonly used model. Below (Figure.4) shows the Cartesian coordinate system to represent this model. Red, Green, and Blue, which are the main colors, are on the three corners of the cube; Cyan, Magenta, and Yellow are on the other three corners. The origin of the cube has Black color and White is diagonally opposite to the origin.

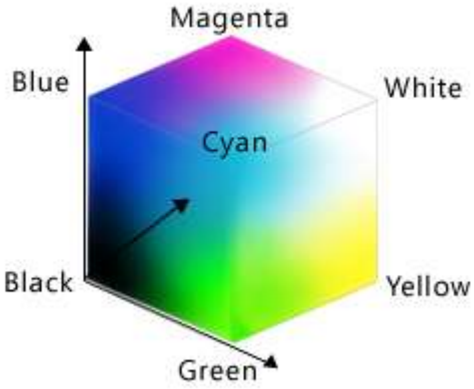


Figure 4: RGB Color Model

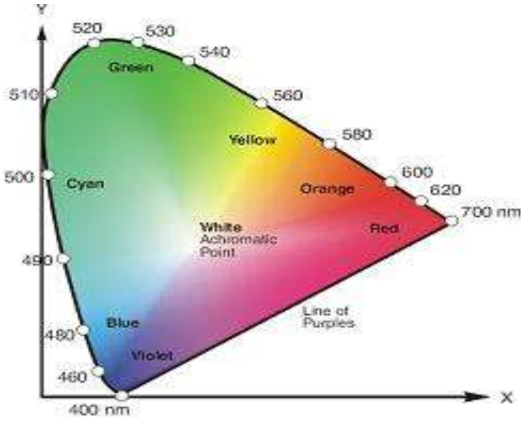


Figure 5: RGB Coordinate System

The line joining White and Black points is the grayscale line. The lines extend along origin to their points on the axes are the three primary colors RGB [20].

HSI color model- CMYK and RGB color models are not suitable for human understanding, for example, the color of an object in daily life is not described by its RGB content hence HSI color model is better to use i.e. Hue, Saturation and Intensity color model. The Hue depicts the color like Blue, Pink and Orange etc. in its pure form. Saturation describes the actual concentration of that color and the brightness is described by Intensity.

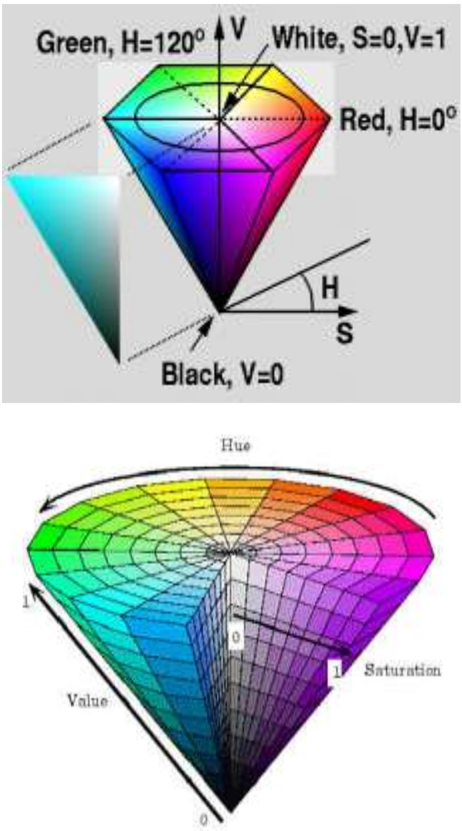


Figure 6: HSI Color Model

RGB ↔HSI Transformation

Conversion between HSI and RGB color models is computationally easier and easily perceivable than for any other model. The formulae used for conversion are mentioned below [31].

First of all the RGB values will have to be normalized [31].

$$r = \frac{R}{R+G+B}\ , g = \frac{G}{R+G+B}\ , b = \frac{B}{R+G+B}$$

HSIvalues are computed by using r,g and b values as follows

$$h = \cos^{-1}\left\{\frac{0.5\cdot[(r-g)+(r-b)]}{[(r-g)^2+(r-b)(g-b)]^{1/2}}\right\} \qquad h \in [0,\pi]\; for\; b \le g$$

$$h = 2\pi - cos^{-1}\left\{\frac{0.5\cdot\; [(r-g)+(r-b)]}{[(r-g)^2+(r-b)+(r-b)]^{1/2}}\right\} \quad h \in [\pi, 2\pi]\;\; for\; b > g$$

$$s = 1 - 3\cdot min(\,r, g, b)\,; \qquad s \in [0,1]$$

$$i = (R+G+B)/(3.255); \qquad i \in [0,1]$$

$$H = h \times \frac{180}{\pi}; \qquad S = s \times 100 \;\; and \;\; I = i \times 255$$

1.3.2. The Spatial Layout of the Color Feature

There is a disadvantage of using the color histogram, as it sometimes produces false results. Consider the images in Figure.7. [29]. The numbers of pixels having a black color in all images are the same as the number of white pixels, hence the color histogram will be the same for both the images but visually the two images are not similar [5]. The spatial layout technique is used to handle this type of situation. This technique considers the spatial distribution of color [10].

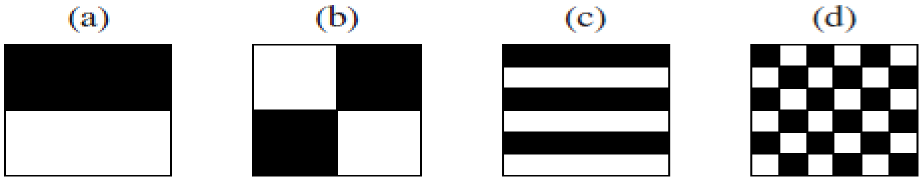


Figure 7: Same Color Histogram for All Different Image but Different Color Layout

Previous work done in this field of spatial layout divides the whole image into 3*3 rectangular grids. We carved five regions to enhance the center region and the local histogram was computed for all the five regions. All five histograms were then compared while matching the similarity between two images [11], [29].

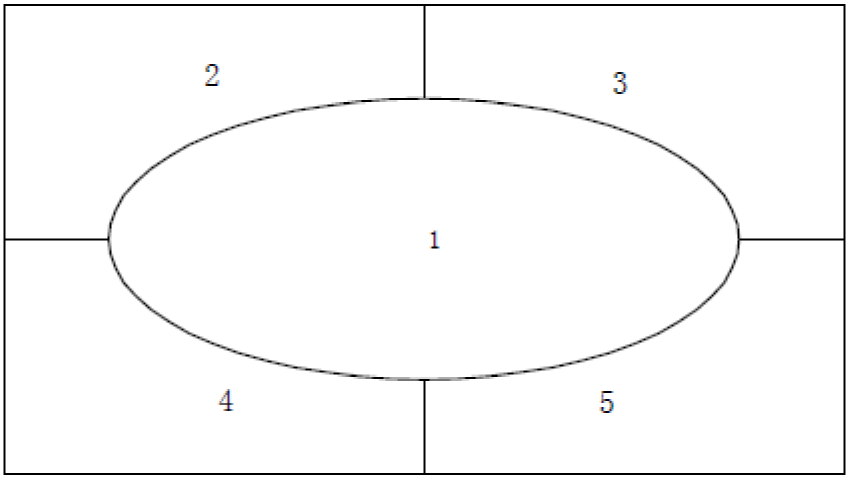


Figure 8: Region Distribution for Computing Histograms

The previous Model of Color Representation does not take into account the spatial layout of colors. To overcome this, we are partitioning the image into a layout region of fixed size and calculating features of color distribution for each of them.

The simplest way to store spatial information in the index is to divide the image into sub-images and then to extract the color features for each sub-image. Dividing the image into sub-images encodes the spatial information too strictly, i.e., during the matching phase, it is extremely hard to account even for small rotations and translations of an image. Relying on a complete segmentation implies in most domains of application a large amount of user interaction at the database population phase which is too time-consuming for large image or video databases. Most systems [5, 16, 1] work with more than one index to exploit the advantages of different approaches: no spatial information, image subdivisions, automated or user-assisted segmentation. The disadvantage of such an approach is that using several indices can make the combined index relatively large. Furthermore, a combined index may contain redundant information that makes the design of an efficient and yet intuitively clear retrieval function hard. A minimal amount of spatial information encoded in the index is necessary to improve the discrimination power of plain color indexing techniques. We propose an approach that lies between subdividing the images and relying on fully segmented images. We work with five partially overlapping, fuzzy regions from which we extract the average color and the covariance matrix of the color distribution.

The above figure.8 shows the division of the image into five sub-regions having center region enhanced by giving a higher priority.

1.3.3. The Texture Feature

Texture describes the homogenous pattern which is an innate property of an image. It means objects are structurally arranged in the image. Two different textures can be differentiated and they can be visualized but it is hard to define it. Texture specifies the coarseness, uniformity, smoothness etc. of objects in an image. Texture depends on the intensity of objects on different areas of the image [9], [26].

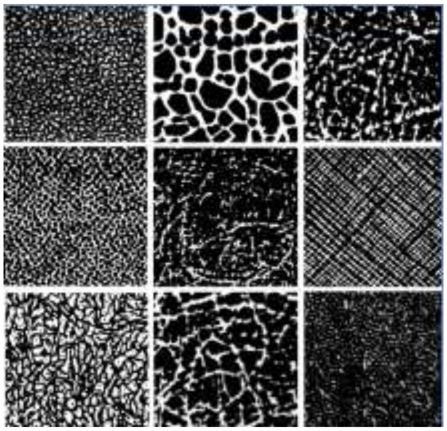


Figure 9: Different Texture Patterns

Texture analysis is an interesting, research-oriented and difficult topic in image processing since much work has been done on textures but still more remains. Common methods for describing it are Transform based methods and Statistical based methods [10], [12], [16], [22].

1.3.3.1. Statistical Method

The statistical method uses the gray level value for a point to compute the local features in the image. From these local features, certain statistics have been derived. To examine the statistical distribution of intensity a number of parameters have been computed in an image. An effective statistical method Gray Level Co-occurrence matrix [5, 6] is used to model a large class of textures [9], [27].

1.3.3.2. Transform Based Method

In this method, the frequency content is computed for analyzing the texture feature. Fourier transform is commonly used to compute the global frequency content which works at a single scale, however at a different degree of details there may be some texture patterns, a multiresolution analysis is used to analyze these patterns. [9].The tree-structured wavelet transform is used for this multiresolution analysis [7], [22], [23].

1.3.4. The Shape Feature

The space occupied by an object in its locality/surroundings is its shape which is described by contour or outline [25]. In an image, any object is distinguished by its outline from its background or surroundings. The shape of the objects is very much perceived by the human eye in the environment. In CBIR, Shape is an essential feature or the region of interest in images [8], [21], and [24].

In image processing, the shape is the segmented result of the outlines or contours of elements. Techniques for the identification of shape features are broadly classified as [4], [12].

- Region-based
- Boundary-based

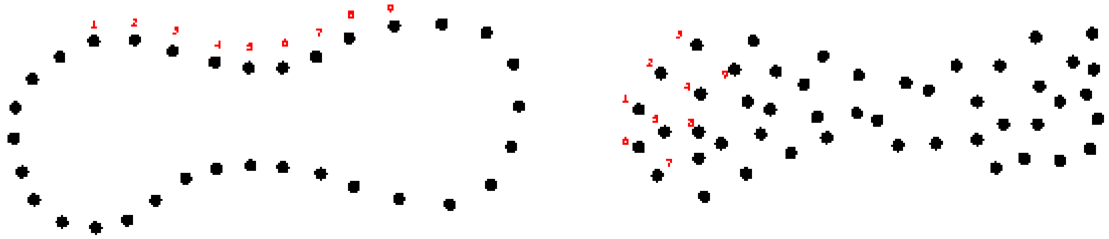


Figure 10: Boundary Based and Region Based Shape Representation

Region-Based- Internal characteristics like representing a region by pixels are used to describe the shape region.

Boundary Based- External characteristics describe the shape region like the boundary on any object representing by its pixels in an image [8].

Boundary-based regions used for simple methods like finding the eccentricity, perimeter, compactness, area, elongation, etc. of the regions [10]. It is also used for some complex methods like calculation of moments or Fourier descriptors [4], [11], [16].

1.4. Research Objective and Motivation

The main objective of this paperwork is to overcome the limitations of Universal CBIR Model for preventing the trademark infringement by suggesting the use of the Multi-Feature CBIR Model for Trademark Image Distinctiveness [2] that is an extension over the existing model with the help of some computational work, formulae and comparison values for the existing and extended features. Also, the objective includes the different Shape feature techniques (like Edge Histogram & Sobel Operator) which were not considered during previous work of the Universal Model.

As we already know that any trademark image is similar either physically or informatively to an already registered trademark image in any way that can be sufficient enough to arouse confusion to cause infringement. The similarity can be observed or measured in different ways according to the constraints and features selection [3].

Universal Model for CBIR system uses less number of features for the retrieval of images. They use texture along with the shape feature rather than use both of them. Also features techniques used are limited to their scope. Suggested Multi Feature model enhances the feature techniques and focuses on proper selection to prevent trademark infringement [4], [13], [25].

Universal Model is being implemented with the norm of giving equal priority to each feature. It treats all of them with equal priority. No one feature dominates over the other while calculation of similarity between two images [9]. Existing CBIR model worked on all the perimeters but it is analyzed that in the real world certain features can be useful in the processing of images rather than apply all of them.

The best suitable way is a flexible feature selection system which can give priority to feature and increase the efficiency of the system i.e. use only those features which can best describe the image along with the other features. Hence in a flexible system, the best feature is being assigned with higher weight successively to the next one. This is what exactly Multi Feature CBIR Model focuses on [13], [24]. Higher recall and precision values can be expected for Multi-Feature CBIR by using the proposed method [21].

1.5. The Layout of the Paper

Chapter 1 defines the problem domain to the reader. This chapter covers all the prior knowledge and the background details required to correctly understand the crux of this project work.

Chapter 2 contains the details of the literature survey examined before starting this project work as well as revised in between the project work to achieve the desired objective/ goals.

Chapter 3 titled “Proposed Multi Feature CBIR Model” has two parts. The first part expatiates on the various image features. These features are used in the proposed method while the second part illustrates the whole details of the database used.

Chapter 4shows the performance and results in values by testing the correctness of the Multi-Feature CBIR Model. Hardware and software specifications also mentioned in this chapter [2].

Chapter 5 conclusions derived and future work has been discussed. This chapter also mentioned references that have been used.

2. PREVIOUS WORK AND LITERATURE REVIEW

Before the beginning of work, it is better to review what has been done previously [11]. This revision helps the developer to better understand the work and acknowledge him/her the shortcomings of the developed mechanism. This chapter describes the details about the already work done in the field of CBIR and trademark retrieval systems by mainly focusing on the Universal Model for CBIR [3]. The techniques and methods used by other researchers in the past have been briefly discussed. It also describes the details about some of the already exist commercially CBIR systems [2], [10], [21].

2.1. The Formulation of The Idea

The field of image processing came to prominence in the late 1970s. In this regard in 1979 a conference held in Florida. Database Techniques for Pictorial Applications was the concerned topic in that conference [12], [27]. Before this, image database management was intangible from the vicinity of researchers but since the beginning of the 1980's the potential applications regarding image database management were coming to the technical world and thus attracted a lot of researchers. Now, this field is spreading in all possible domains including military, scientific, medical, imaging industry, gaming applications of imaging etc. [11], [16].

Text-based search for images faces a lot of problems and drawbacks which are increasing with the advancement of the technology thus an urgent need of a system is required which is free from drawbacks and efficiently manage the increasing visual digital information [27].

A lot of extensive work and hard research has been carried out around the world in the field of trademark image retrieval systems. The three most effective and outstanding trademark retrieval systems that are present are TRADEMARK, STAR, and UNIVERSAL MODEL using CBIR approach.

Kato’s TRADEMARK [1] by the use of Graphical features (GF vector) system analyzes the image content automatically. Lam et al. [2] developed a STAR system that focuses on the shape of an object present in the image and spatial layout of the image and used mainstream image retrieval techniques to calculate the similarity between images [10].

In UNIVERSAL MODEL for CBIR, S. Nandagopalan et al. [1] used the Euclidean distance method to compare similarity which focused on color histogram, image edge density features, and texture of query and database images.

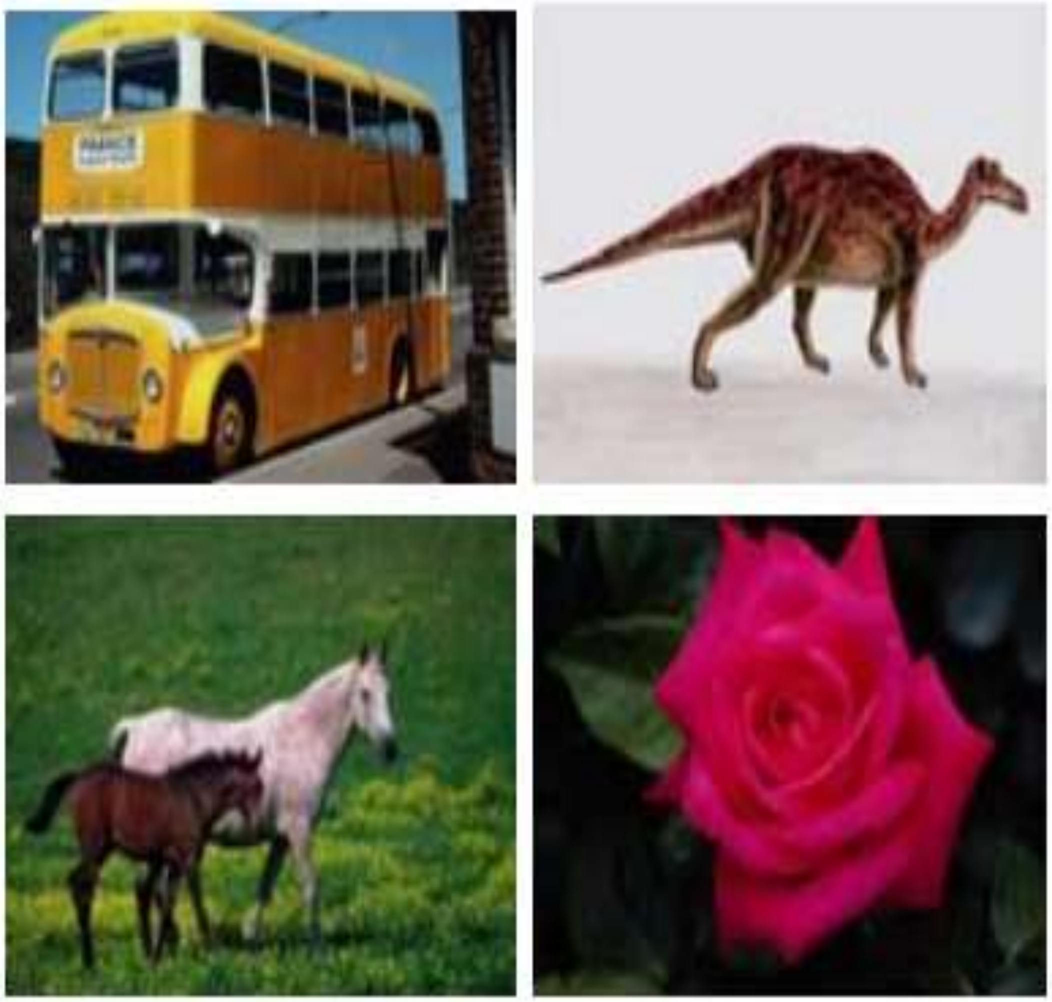


Figure 11: Query Images Used In the Universal Model.

Table 1: Recall and Precision Value of the System

Query Image	Color	Texture	EHD	All
1	21.8 28.0	50.0 15.0	23.6 34.1	35.2 60.0
2	100.0 98.0	75.0 62.0	87.0 68.0	100.0 78.0
3	74.6 59.0	20.0 10.0	65.0 37.0	42.8 90.0
4	91.0 24.0	75.0 33.0	85.6 34.9	92.0 28.0

Table 1 shows the precision and recall values calculated for four images (Figure 11) used in the universal model. In the cells of the table, the upper entry is the precision value and lower entry is the recall value of the images. These precision and recall values are based on the color, texture and edge histogram of density. The column 'All' shows the percentage of precision and recall values calculated for color, texture and edge histogram of density.

In RGB and HSV color space color histograms have been used for the representation of the color feature.256 bins have been used for quantization of this histogram.[15][30] The most popular statistical methodologies used for finding the texture of an image are: [9]

Tamura texture

Co-occurrence matrix

Wavelet transform

Kondekar et al. [7] presented various methods for the mathematical description of shape attribute:

Boundary-based [10]:

Curvature Models

Polygonal Models, boundary partitioning

Splines, higher order constructs

Fourier Descriptors:

Region-based [10]:

Blum's skeletons

Super quadrics

Implicit Polynomials

Fourier Descriptors

Fourier descriptors and moment invariants are most successful representation according to the paper.

Ranjeet Kumar et al. [11] had done the survey on various CBIR systems that made to get a better depth of the work before preceding the paperwork.

The first known commercially acclaimed CBIR system was QBIC (Query by image content). The way query to be generated in QBIC is:

Like a simple query, an only single feature is processed at a time or.

Like a multi-query, multiple features are processed for the retrieval purpose.

Table2 mentioned the techniques of famous commercial CBIR systems VisualSEEK [13], QBICandNeTra [12]

Table 2: Comparison of Techniques

Features → System ↓	Color	Shape	Texture
QBIC	Histogram HSV	Tamura Image, Euclidean Distance	Geometric Characteristics, moment Invariants
VisualSEEK	Histogram HSV		
NeTra	Histogram HSV, Color codebook Histogram HSV	Gabor Filter, Tamura Image, 3D Histogram	Fourier Descriptor

The above table is showing the different types of techniques used by famous commercial CBIR systems such as QBIC, VisualSEEK, and NeTra. These techniques have been used for three major features like color, texture, and shape of an image. QBIC used Histogram (HSV) for color, Tamura Image and Euclidean distance for Shape and Geometric characteristics and Moment Invariants for Texture. VisualSEEK used a Histogram (HSV) for color. NeTra used Histogram (HSV), color Codebook Histogram (HSV) for color, Gabor filters, Tamura image, a 3D histogram for shape, and Fourier descriptor for texture.

2.2. Conclusion

In the late 1970s, image processing was a major agenda for researchers. Many research works have been done and implemented in the image retrieval field. This chapter described the approaches used in the past by the numbers of researchers [12].

3. PROPOSED MULTI FEATURE CBIR MODEL

Until now the discussion was about a better understanding of the existing Universal Model using CBIR methodology. In this chapter, we will now explain our write-ups for our proposed Multi Feature CBIR Model for Trademark Image Distinctiveness that is an extension over the existing Universal Model. We will show all the improvements over the existing Universal CBIR Model by discussing our proposed methods and the use of different feature techniques to prevent trademark infringement and better image retrieval. We will explain my write-ups by providing the details of my computational work and the formulae used.

As the Model name says ‘Multi Feature CBIR’ we will be proposing unique combinations of different techniques for four features (Color, Texture, Shape and Spatial layout) and their proper usage. For this, we have done some computational work, formulae calculations, comparisons and finally prepared a Proposed Method that shows the flow of data and explains how ‘Query Image' will be compared against ‘Database Image'. We also suggested using all the weight values calculated for these features together by combining them with the help of a proposed calculated final formula and compare against the database. We have used this software using ‘Multi Feature CBIR Model’ over a Database of 175 images, observed our result, and compared the result with the existing Universal Model. Comparison results with outputs will be mentioned in the Result & Conclusion section latter.

Also as it is mentioned that this proposed work of the Multi-Feature CBIR Model is an extension and improvement of previous research work of the Universal CBIR Model, some standard and proven matrixes, formulas and calculations have been used directly by providing their source and references in References Section of this paper.

This chapter will elaborate firstly the Proposed Method, different feature's techniques used for finding similar images with the proposed methodology and then the trademark database used with its source and specification at the end.

3.1. Proposed Method

For the purpose of software creation using MATLAB, the proposed method is depicted with its block diagram shown below. The proposed method consists of the Query image, database images, techniques that are used for color, texture and shape feature, feature matching and retrieval of images. Query image is the image which is for comparison with the database images to find out the similar images. Database images are the collection of similar images used for the comparison with the query image. Techniques shown in the below diagram such as Color Histogram is used for the color feature, GLCM and Wavelet Transform (Discrete Wavelet Transform and tree-structured wavelet transform) are used for texture feature, Edge Histogram is used for shape feature and spatial layout is calculated by circular portioning. These techniques are also used for database images. After using all techniques for color, texture and shape feature, the system is using the Euclidean Distance equation for similarity matching (Feature Matching) between query image and database images. As per the feature matching, the system is retrieving similar images from the database to show the result.

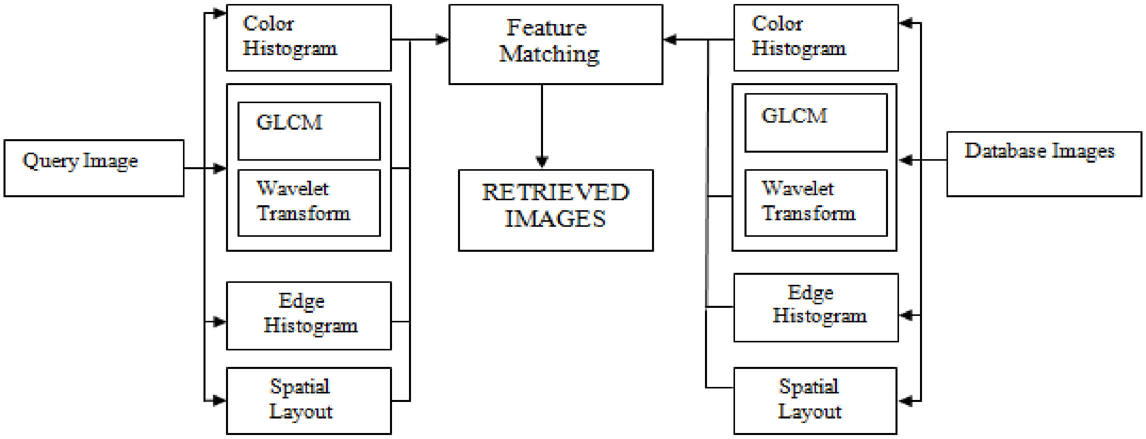


Figure 12: Block Diagram of the Multi-Feature CBIR Model

3.1.1. The Technique Used for Color Feature

3.1.1.1. Color Histogram

The color histogram is the representation of color and its frequency on a graph where the X-axis represents the color and Y-axis is for the frequency [10], [11]. The number of bins in the bar graph is the same as the number of colors present in an image and the frequency of any color value represents by a bin in the image. Color histogram method is invariant to rotation of image [5], [15], [20]. The steps for finding color histogram to find the similarity matching between two images is:

a) First Query and database images are read and their HSV color map is calculated using RGB to HSV formulae (Mentioned in the Introduction Chapter 1).

b) Color histogram for HSV color space is computed for database and query images.

c) The distance between two histograms calculated by the Euclidean distance metric approach. [5], [29], [30].

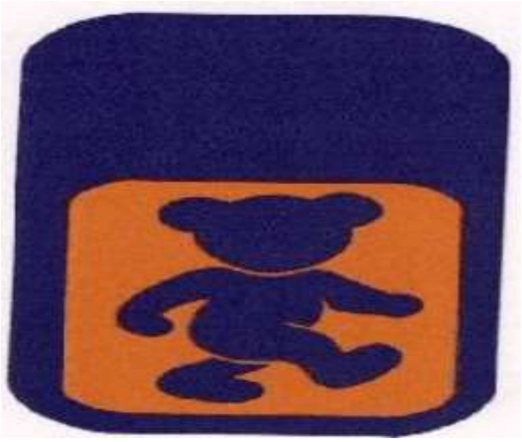


Figure 13: Sample Trademark Image

Below Figure.14, Figure.15 and Figure.16 show the histogram of Hue, the histogram of saturation and the histogram of Value respectively of the sample trademark image fig13.

For this, the MATLAB language has been used in the software and below is the code to draw the hue, saturation and value histogram.

```
“shapeHistogram.m”

function [histogram]=shapeHistogram(x)
image = rgb2gray(x);
% smoothing
image = BiSmooth(image,8);
[rows,cols]=size(image);
%sobel operator
Sy = [1 2 1; 0 0 0; -1 -2 -1];
Sx = [1 0 -1; 2 0 -2; 1 0 -1];
%convolution
Gy = conv2(double(image),double(Sy),'same');
Gx = conv2(double(image),double(Sx),'same');

for x=1:rows
    for y=1:cols
        X=Gx(x,y);
        Y=Gy(x,y);
        G(x,y)=sqrt( double(X*X + Y*Y) );
        if(G(x,y)>=150)
            G(x,y)=1;
        else
            G(x,y)=0;
                    end
        direction(x,y)=(atan2(double(Y),double(X))+pi)*180/pi;
    end
end
G = bwmorph(G,'skel',Inf); % finding the skeleton of the gradient image
```

```
histogram = zeros(1,72); % quantized histogram
cnt=0;

for x=1:rows
    for y=1:cols
        if(G(x,y)==1) %if(direction(x,y)>=0)
            index=roundn(direction(x,y),1); %rounding off to nearest 10's
place to keep the angle integer
            index=index/5; % bin size=5
            if index == 0
                index=1;
            end
            histogram(index)=histogram(index)+1;
            cnt=cnt+1;
        end
    end
end
histogram = histogram./cnt ; %normalized histogram/probabilities are mapped
not frequencies
```

"hsvhist1.m"

```
function a = hsvhist1(q)

j=rgb2hsv(q);
hue = j(:,1);
sat = j(:,2);
val = j(:,3);

[huecounts,x] = imhist(hue);

[satcounts,y] = imhist(sat);

[valcounts,z] = imhist(val);

a = [huecounts satcounts valcounts];
```

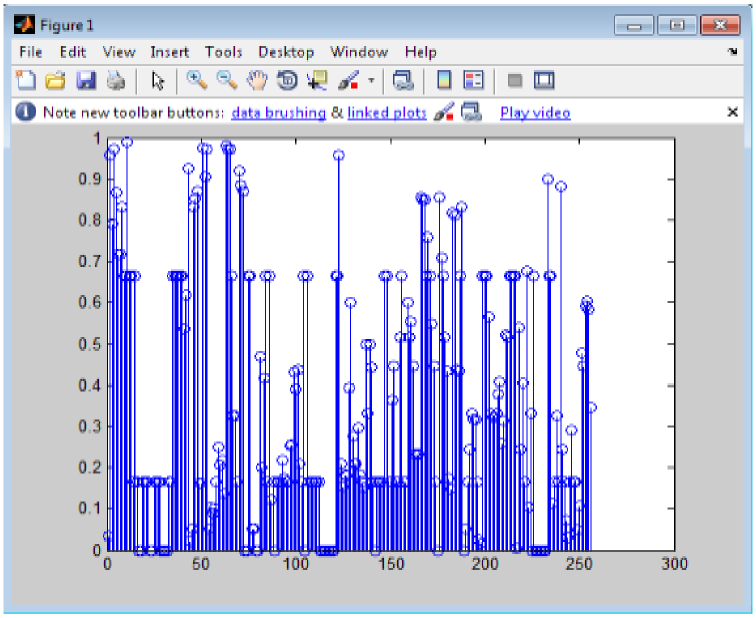


Figure 14: Histogram of Hue

The above figure shows the histogram of hue for sample trademark image fig13 by assigning different weights. This figure is drawn using a plot feature of MATLAB.

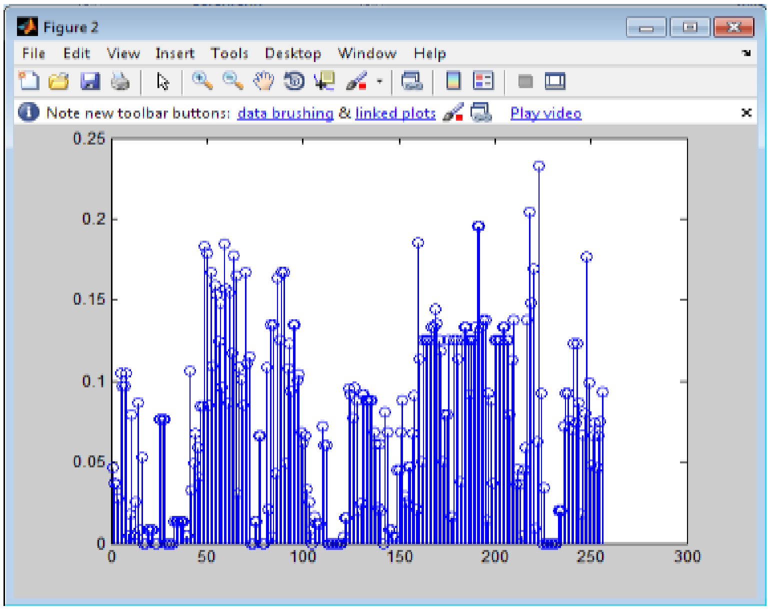


Figure 15: Histogram of Saturation

The above figure shows the histogram of saturation for sample trademark image fig13 by assigning different weights. This figure is drawn using a plot feature of MATLAB.

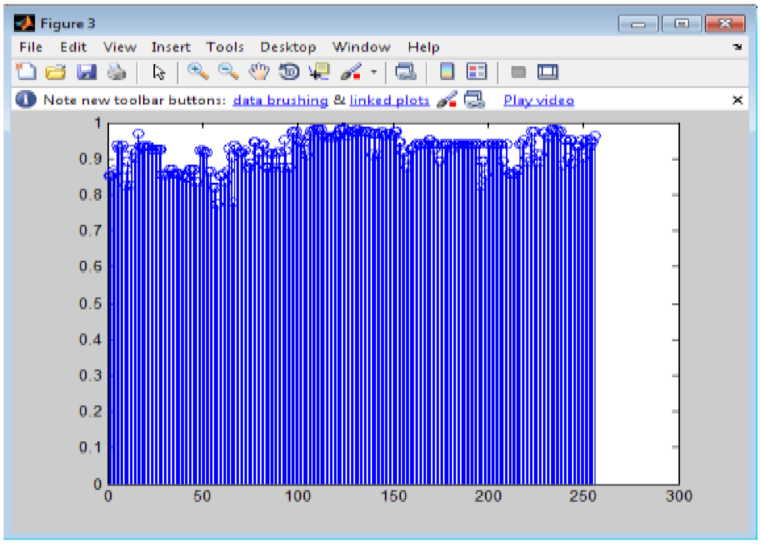


Figure 16: Histogram of Value

The above figure shows the histogram of value for sample trademark image fig13 by assigning different weights. This figure is drawn using a plot feature of MATLAB.

The color histogram is computed for all regions. The minimum distance between their corresponding regions of two images more they (images) are similar. By assigning different weights the priority can be changed for every region [5], [27], [29].

3.1.2. The Technique Used for Spatial Layout

3.1.2.1. Circular Partitioning

This method uses circles for partitioning the image into sub-regions. It is better to use circular partitioning compared to grid partitioning because circles do not get affected by rotation, flipping or any alteration. If an image is partitioned into R circles then total R+1 sub regions are formed of which R-2 regions are rings, one is a circle and one left over a region as shown in fig17.

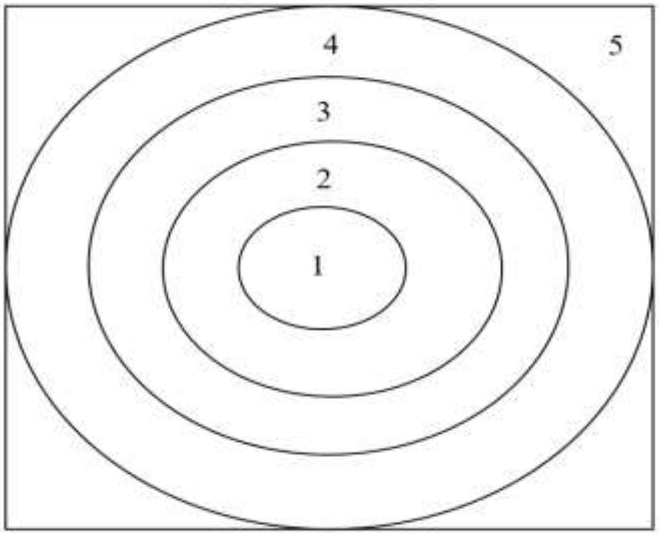


Figure 17: Circular Region Distribution of the Proposed Method

Using this circular partitioning method we have divided the input image into 5 different sub regions region1, region2, region3, region4 and region5. We have implemented following code using circular partitioning for spatial layout:

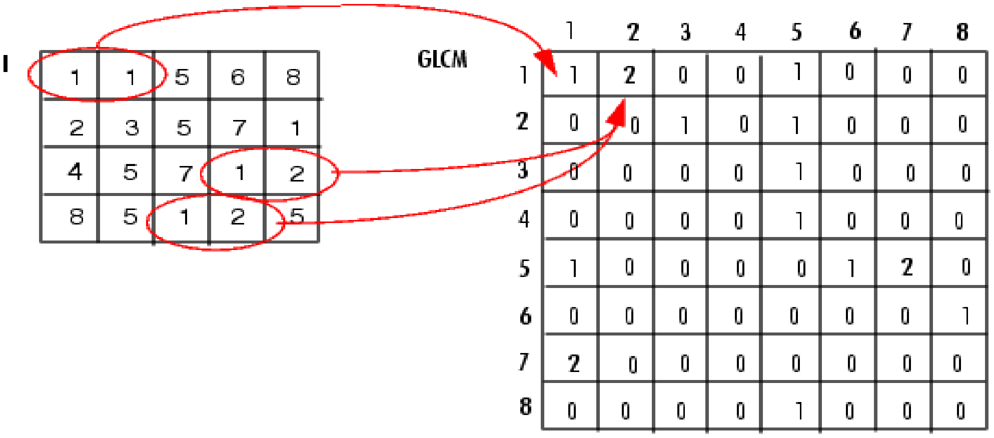
```
region1 = [];
region2 = [];
region3 = [];
region4 = [];
region5 = [];

r1 = 0;r2 = 0; r3 =0;r4 = 0;r5 = 0;
for i = 1 : len
    for j = 1 : wid
        curr_pixel = [ i j ];
        dist = euclideanDistance(centre,curr_pixel);
        dist = round(dist);
        if( dist <= radius1)
              r1  = r1 + 1;
           region1(r1,:) = [image_h(i,j) image_s(i,j) image_v(i,j)];
        elseif( dist > radius1 && dist <= (2 *radius1))
              r2 = r2 + 1;
            region2(r2,:) = [image_h(i,j) image_s(i,j) image_v(i,j)];
        elseif( dist > 2 * radius1 && dist <= 3 * radius1)
              r3 = r3 + 1;
              region3(r3,:) = [image_h(i,j) image_s(i,j) image_v(i,j)];
        elseif( dist > 3 * radius1 && dist <= 4 * radius1)
               r4 = r4 + 1;
            region4(r4,:) = [image_h(i,j) image_s(i,j) image_v(i,j)];
        else
           r5 = r5 + 1;
           region5(r5,:) = [image_h(i,j) image_s(i,j) image_v(i,j)];
        end
    end
end
nbins = 256;

hist_r1_h = imhist(region1(:,1),nbins);
hist_r1_s = imhist(region1(:,2),nbins);
hist_r1_v = imhist(region1(:,3),nbins);

hist_values_r1 = [hist_r1_h hist_r1_s hist_r1_v];

hist_r2_h = imhist(region2(:,1),nbins);
hist_r2_s = imhist(region2(:,2),nbins);
hist_r2_v = imhist(region2(:,3),nbins);


hist_values_r2 = [hist_r2_h hist_r2_s hist_r2_v];

hist_r3_h = imhist(region3(:,1),nbins);
hist_r3_s = imhist(region3(:,2),nbins);
hist_r3_v = imhist(region3(:,3),nbins);

hist_values_r3 = [hist_r3_h hist_r3_s hist_r3_v];

hist_r4_h = imhist(region4(:,1),nbins);
hist_r4_s = imhist(region4(:,2),nbins);
hist_r4_v = imhist(region4(:,3),nbins);

hist_values_r4 = [hist_r4_h hist_r4_s hist_r4_v];

hist_r5_h = imhist(region5(:,1),nbins);
hist_r5_s = imhist(region5(:,2),nbins);
hist_r5_v = imhist(region5(:,3),nbins);

hist_values_r5 = [hist_r5_h hist_r5_s hist_r5_v];

hist_values = [hist_values_r1; hist_values_r2; hist_values_r3; hist_values_r4;
hist_values_r5];
```

3.1.3. The Technique Used for Texture

3.1.3.1. Gray Level Co-occurrence Matrix

A **co-occurrence matrix** is a matrix in which the distribution of co-occurring values in an image at a given offset has been defined. Let Q (i, j) an image, P (i, j) be offset, a matrix A is computed which is N*N in size.

Suppose P matrix of size as of A. When A is divided by the total number of pairs that fulfill offset value p the matrix P obtained. P is called the GLCM [4, 5] characterized by position operator p. P (i, j) is the joint probability. Angle θ and distanced characterized the relationship operator R.



1	1	5	6	8
2	3	5	7	1
4	5	7	1	2
8	5	1	2	5

	1	2	3	4	5	6	7	8
1	1	2	0	0	1	0	0	0
2	0	0	1	0	1	0	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	0	1	0	0	0
5	1	0	0	0	0	1	2	0
6	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0

Figure 18: Computing the Gray Level Co-occurrence Matrix

P matrix is obtained by using formula-

$$P(i,j) = \sum\nolimits_{P=1}^{M} \sum\nolimits_{q=1}^{N} \begin{cases} 1 \quad if\ A(i,j) = i, \\ A\big((i + \Delta x), (j + \Delta y)\big) = \mathrm{j} \\ 0, \ \ otherwise \end{cases}$$

Here Δy and Δx specify the displacement vector. Intensity value at pixel location (i, j) specifies by A (i, j). These values are calculated for both database and query images. Two similar images have close values. Since offset (Δ*x*, Δ*y*) values are direction dependent hence this method is independent of image rotation. For an ideal CBIR system, it is an undesirable condition.[21]

So for removal of direction drawback, several matrices for different directional offset vector are computed corresponding to the rotation of image through 0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°, and 360°. Several texture descriptors can be calculated by using the value obtained.[26]

Shanmugam et al. [17] suggested 14 such texture descriptors obtained through the values of GLCM each impart one property description of the texture. These 14 descriptors included contrast (moment, variance), correlation, angular second moment, etc. [17].

3.1.3.2. Tree-Structured Wavelet Transform

An image is simply a 2D matrix of coefficients or pixel values. These coefficients are the intensity values at particular locations in the image. One cannot differentiate between the pixel locations looking merely at the coefficients that are more essential [23]. Typically in an image, sharp color variations represent fine details while smooth color variations describe more important features. One can easily find the important coefficients by conversion of the spatial distribution of the image into frequency domain i.e. low frequencies represented smooth variations of intensity and the high frequencies represented fine details which mean the edge or shape resulted by sharp intensity changes [8]. Thus the low-frequency components contribute in the base of the image while sharp details constitute by high-frequency components in the image. Hence the low-frequency components are concerned topic in the image [4], [25].

The tree-structured wavelet transform [23] breaks down sub-signals which contain essential information of images in the low-frequency channels. In this method, the signal is broke down into sets of frequency channels. These channels are low-frequency channels with narrow bandwidths. By this wavelet method signals containing smooth components i.e., low frequency are suitable to be transformed because lower frequency regions are more concentrated within the formation. Here Daubechies used as the wavelet function.

The Daubechies wavelets are a family of orthogonal wavelets defining a discrete wavelet transform and characterized by a maximal number of vanishing moments for some given support. There is a scaling function (called the father wavelet) which generates an orthogonal multiresolution analysis. The wavelet transform is also easy to put into practice using the fast wavelet transform. Daubechies wavelets extend the Haar wavelets by using more complicated average and difference operators. These operators are computed by filtering the signal and then sub-sampling by the resulting low/high pass filtered signal.

Each wavelet is uniquely determined by its low pass filter. The larger the size p=2*k of the filter, the higher is the number k of vanishing moment [32].

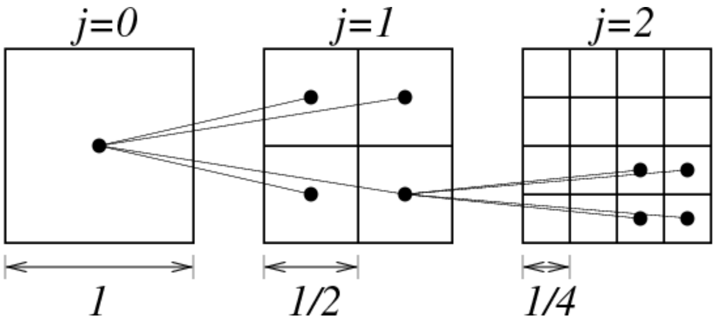
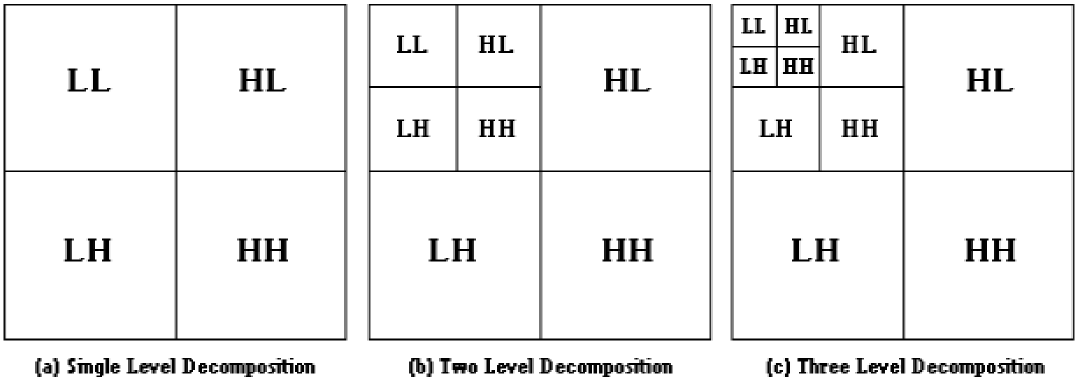


Figure 19: Tree-Structured Wavelet Transform



(a) Single Level Decomposition (b) Two Level Decomposition (c) Three Level Decomposition

Figure 20: Levels of Decomposition

3.1.3.3. Discrete Wavelet Transform

By applying DWT on an image its 2-dimensional wavelet decomposition is performed using a particular wavelet function (Daubechies wavelet is used in this paperwork) [23].

The analysis filters are used to exactly halve the frequency range i.e. a low pass filter and a high pass filter. Firstly low pass filter is applied; the first half of the actual frequency range has the output frequencies of this filter. Similarly, high pass filter passes only high frequencies and does not affect high frequencies. High pass filters are usually used for sharpening. A high pass filter tends to retain the high-frequency information within an image while reducing the low-frequency information. The kernel of the high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels. The kernel array usually contains a single positive value at its center, which is completely surrounded by negative values. The following array is an example of a 3 by 3 kernel for a high pass filter: [22], [33].

0	-1/4	0
-1/4	+2	-1/4
0	-1/4	0

The output is again divided into four different sub bands labeled as low–low frequency, low–high frequency, high–low frequency, high–high frequency. Since the low-frequency component contains more about the base of the image only it is concerned. So the iteration of this method applies on a low-low component till the desired level is not reached [27].

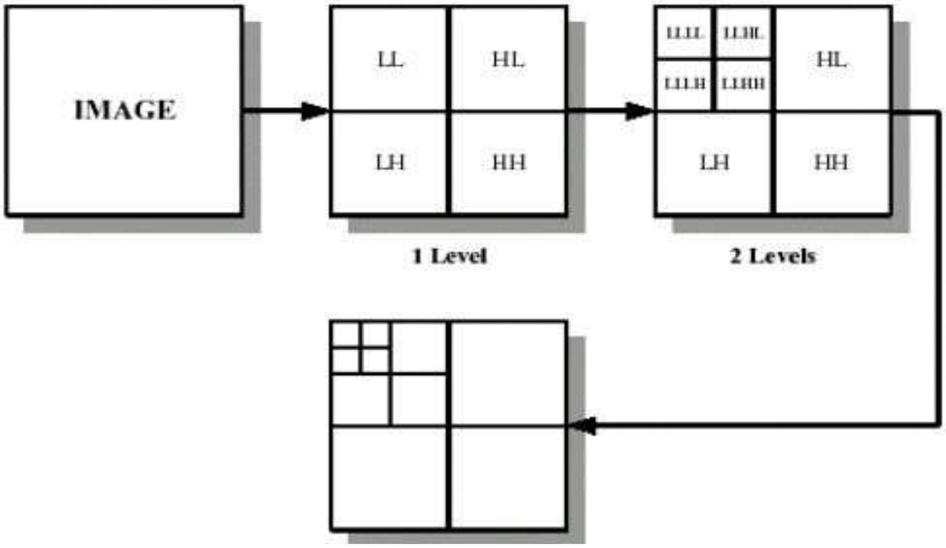


Figure 21: Decomposition of an Image into Sub-Bands

3.1.4. The Technique Used for Shape

3.1.4.1. Edge Histogram[15]

"Sobel" edge operator has been used for finding out the edges and their respective directions. The intensity of edges is not precisely accurate when calculated using the Sobel operator so to tolerate the slight variation, quantization of edge directions is done [8].

The total 0° to 360°degrees of edge directions in this present work has been quantized into 72 levels so that will contain 72 bins in edge histogram. Thus 0° to 5° of edge count merged into a single bin in the shape histogram of an image [4], [27].

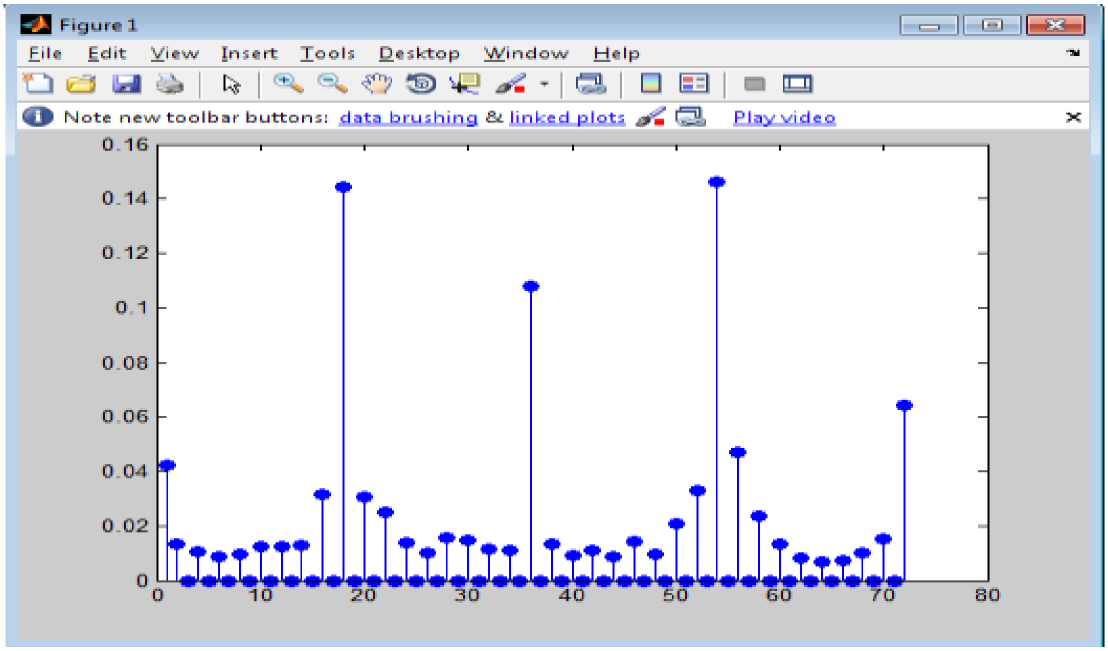


Figure 22: Edge Histogram Obtained for a Trademark Image

3.1.4.2. Sobel Operator

Sobel operator simply calculates the image gradient value. This value is calculated for each pixel. The largest variation and the rate of variation from light to dark values are found out by the Sobel operator. The existence of an edge at any point depends upon the smooth or abrupt change in intensity value for that particular point. Finding an edge in an image is considerably easier than its direction.

The image is convolved with the filtering window in both vertical direction and horizontal direction. To calculate the derivatives Sobel operator uses two 3×3 kernel functions for vertical and horizontal variations separately. These kernel functions are convolved with the original image matrix. Suppose **A** is the original source image, the vertical and horizontal derivative approximation values at each point have been shown by **Gy** and **Gx** images. The calculation of Gx and Gy is shown below:

$$\mathbf{G_y} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * \mathbf{A} \quad \textbf{and} \quad \mathbf{G_x} = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

Here*is for a 2-dimensional convolution operation.

An increment in the “down” direction defines by y-coordinate, while the x-coordinate defines an increase in "right"-direction. A gradient approximation is the gradient magnitude value at each pixel in the image matrix which can be obtained, using:

$$\mathbf{G} = \sqrt{\mathbf{G_x^2} + \mathbf{G_y^2}}$$

Gradient’s direction is calculated as:

$$\boldsymbol{\theta = arctan\left(\frac{G_y}{G_x}\right)}$$

Where **Θ** is 0 for a vertical direction edge.

Steps based on shape property for determination of similarity between two trademark images are: [4], [25].

- To compute the edge histogram the images are converted from color to grayscale [5].

- Preprocess the images like smoothening or removal of any noise. This is to avoid any difficulty in the histogram. In this paperwork, Binomial filters are applied to order 8.
- To compute edge intensity and edge direction the image matrix is multiplied with the Sobel operator.[8]
- The directions are quantized from 0^0 to 360^0in 72 levels.
- A histogram is computed in which the quantized level denotes by X-axis and count of edge pixels in an image having that direction level denotes by y-axis.
- Normalized the histogram by dividing each bin value of histogram by the total number of pixels.
- The distance between the two computed normalized histograms is calculated by the Euclidean distance function.

3.2. Similarity Calculation

Between query image and database images to carry out the similarity evaluation for all the features color, texture, shape, and spatial layout, Euclidean space has been used [2], [6], [9].

The Euclidean distance is measured according to the equation:

$$\mathbf{d} = \left|\sqrt{\{\mathbf{D}\}^{2} - \{\mathbf{Q}\}^{2}}\right|$$

Here, d shows the Euclidean distance between database and query images.

Q represents the feature vector of the requested query image.

D represents the feature vector calculated for the database image.

The computed result lies between 0 and 1 after the normalization of the distance value between query and database images. Weights allocated to each feature are associated to these normalized values hence it is necessary to normalize these values [8].

Assigning a weight to features is essential as a trademark database is diverse the images of trademark varied from company to company. Since it becomes harder to correctly retrieve similar images by using a single feature or having the same priority to every feature [13].

3.3. Combining the Features

Use the only single feature to get the retrieval result may produce an inefficient result. The retrieved images may not similar to the query image. Hence a combination of several features instead of using a single feature has been used to produce efficient results [26].

The similarity between query and database image calculated from three types of features, which include color, texture, and shape features. The distance between the query image and the image in the database for different features calculated as follows:

$$d_1 = \sum_{i=1}^{d} \frac{\left|f_c^Q - f_c^T\right|}{\left|f_c^Q\right| + \left|f_c^T\right|}$$

Where,

d = dimension of the vector
f_c^Q = color feature vector of the query image.
f_c^T= color feature vector of the database image

$$d_2 = \sum_{i=1}^{d} \frac{\left|f_g^Q - f_g^T\right|}{\left|f_g^Q\right| + \left|f_g^T\right|}$$

Where,

d = dimension of the vector
f_g^Q = texture feature vector of the query image.
f_g^T= texture feature vector of the database image.

$$d_3 = \sum_{i=1}^{d} \frac{\left|f_s^Q - f_s^T\right|}{\left|f_s^Q\right| + \left|f_s^T\right|}$$

Where,

d = dimension of the vector
$\mathrm{f}_{\mathrm{s}}^{\mathrm{Q}}$ = shape feature vector of the query image
$\mathrm{f}_{\mathrm{s}}^{\mathrm{T}}$ = shape feature vector of the database image

During similarity measures, we have used appropriate weights to combine the feature. We have selected different and random weight factors assigned to distance values. The reason behind this concept is, in order to compare sub features, it is essential to normalize the similarity distance. The overall similarity(**S**) is the sum of weighted similarities for normalized distance. The overall similarity is calculated by using the below formula where features are combined using variable weights.

$$\boldsymbol{S} = \frac{\boldsymbol{W_{C}.D_{C} + W_{T}.D_{T} + W_{S}.D_{S}}}{\boldsymbol{W_{C} + \ W_{T} + W_{S}}}$$

$W_T, W_C \& W_S$ are the weight factors associated with a texture, color and shape feature respectively and assigned to distance values D_T for texture or D_C for color and D_S for shape respectively [4], [9].

3.4. Database

The Trademark image database used for this research is not a standard database rather a predefined database. We designed this database by gathering similar images from the internet for this research. The trademark images in the database are actual real trademarks issued to various service providers, companies and other legal entities. Official websites are maintained by governments where these registered and issued trademarks are weekly uploaded and updated. The trademark images contained in the database are 8-bit uncompressed images and they are properly indexed in the database. The total number of trademark images are 175 and all trademark images have JPG format and are uniform in size of 256*256. The reason behind to take uniform size for each trademark image is to avoid discrepancy in the retrieved result due to different sizes of images [2].

We used the MATLAB tool to implement our experiment for this research. The calculation of color histogram, GLMC, wavelet transform, edge histogram and spatial layout of the color feature have already been done for all the trademark images. Trademark images saved in the database are ready for the comparison with any query image.

The database containing trademark images further have been classified into various classes and journals for better maintenance. Trademark images issued for goods are in Class 16 while Class 35-45 comprised of trademarks images issued to services [2].

Indexing of images done manually in alphabetically order for example A (1), A (2) (having the same class trademark and) B (1), B (2) (having other same class trademarks) etc. This indexing helps in computing the systems recall and precision values which indirectly judge the performance of this Multi Feature CBIR Model [3], [21]. The system can be judged as good if images of the same class equivalent to the query image retrieved by the system automatically [6].

4. RESULT AND DISCUSSION

This chapter describes the effectiveness, reliability, performance, robustness, and efficiency of the work done during the whole project time. Recall and Precision values are a key component for measurement of Performance whereas average response time is a key component for effectiveness and efficiency. The Recall-Precision graph depicts the correctness of the proposed Multi Feature CBIR Model over the existing Universal CBIR Model [3], [18].

4.1. Performance

As mentioned earlier recall and precision value measures the performance of the CBIR system**.** The recall is the percentage of relevant images retrieved from the database among all possible relevant images while Precision is the ratio of relevant images retrieved from the database to the total number of retrieved images. The reliability, effectiveness, and efficiency of the CBIR system depend on the good precision and recall values [24].

The way of organization of the database and its content has a great role in the CBIR system after all images are retrieved from the database [7]. For example, if a database contains images of trains only then no matter what is the query image, the retrieved result as output will only contain images of trains. This reduces the efficiency of the system as output retrieved by the system has a lesser number of relevant images [21].

We have carefully selected different database images through various sources of previous work and the internet and made sure those are relevant to our main query images.

Relevant DB images determination depends on which low-level features are used to characterize image contents, the criterion used to compare image features, how database objects are ranked with respect to the query (based on either a quantitative measure of similarity or qualitative preferences) and finally, on whether the user is interested in the whole query image or only in a part of it. All of these aspects strongly influence the query evaluation process.

The other case is if the database is too diverse which contains different kinds of images and only a few images can be retrieved which will closely appear to the query image and will not be sufficient for the desired goal, which also reduces the efficiency of the system [6].

Thus to achieve good recall and precision values the database should be large in size with diversity and it should contain a good amount of images similar to the requested query image.

4.2. Recall-Precision Graph

The effectiveness of the Multi-Feature CBIR Model can be easily judged by looking at the Precision and Recall values. We are using the multi-feature CBIR method to get effective and better results to compare to the universal CBIR method. The experiments are conducted on a set of 175 images that have been collected from the internet. The database consists images of a different class but under approximately constant illumination conditions and the image, the resolution is resized to 256 x 256 pixels irrespective of its prior resolution. A query image is provided by the user then similar images from the database are selected and displayed. We have used three different techniques for the retrieval of images. They are as follows:

1) Using only Color (HSV)

2) Using Color (HSV) and Texture

3) Using color, Texture, and shape

Fig 23 shows the graph plotting between the precision and recall and the results are obtained by issuing different query images to the CBIR system [24].

$$precision = \frac{No.of\ relevant\ images\ retrived}{Total\ No.of\ Images\ retrived}$$

$$recall = \frac{No.of\ relevant\ images\ retrived}{Total\ No.of relevant\ Images\ in\ the\ Database}$$

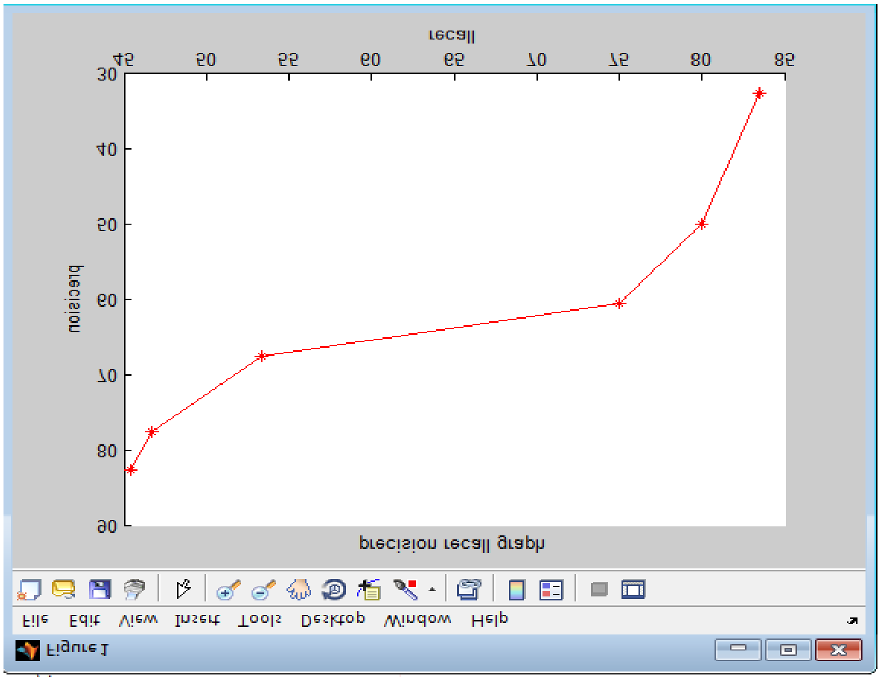


Figure 23: Precision Recall Graph of the Multi-Feature CBIR Model

Table 3: Precision and Recall Values in %

Item Method	Precision	Recall	Average Response Time(Sec)
Color	60%	80%	2.51
Color and Texture	65%	72%	151
Color, Texture and Shape	72%	84%	277

The above table shows the results of Precision and Recall values in %. These results are based on three different methods such as 1) Color, 2) Color and Texture and 3) Color, Texture and Shape and results are in Average Response Time (Sec).

4.3. Average Response Time

The universal model has used only two feature such as color and texture to find out the similarity between images but Multi feature CBIR model is using four features such as color, texture, shape and spatial layout for finding the similarity between images. In the multi-feature CBIR model, we have 11 different types of query images and a database of 175 images that are similar to query images. As we are using more features, the result of similarity matching of images is more efficient than the universal model.

We have done multiple experiments using our different types of query images based on four different combinations of features as:

1) Color
2) Shape
3) Texture
4) Color and Texture
5) Color and Spatial Layout
6) Color, Texture, Shape and Spatial Layout

Along with the feature combination, we have also focused on the average time of finding the relevant images by the system.

We have done the experiments for all 11-query images to see the comparison of the average time for finding the similarity between images. Our experiments are as follows:

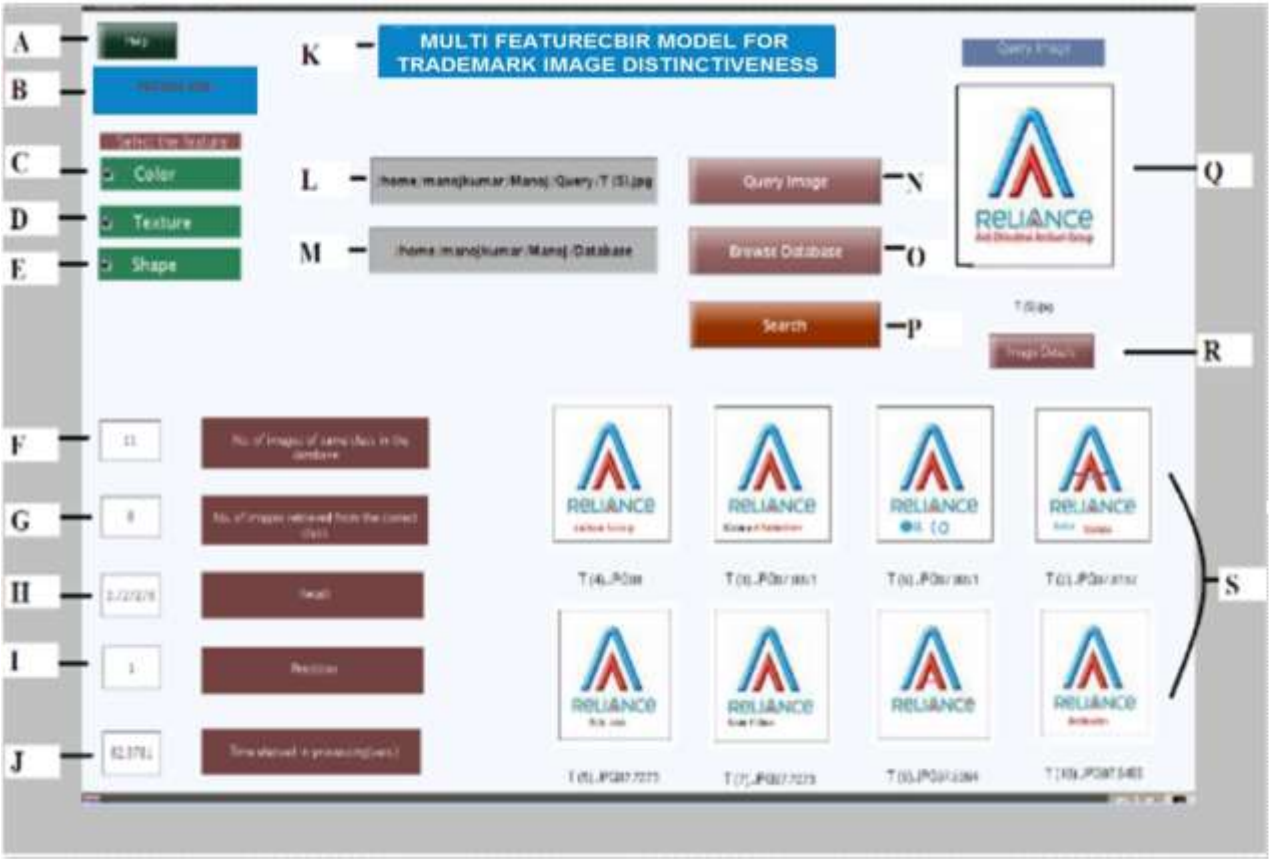
- When we compared the query with database images based on **color feature**, the system took **2.51 seconds** of average retrieval time to show the relevant images using a multi-feature CBIR model but with the Universal model, it was **3.1 seconds** for the color feature.
- When we compared the query images with the database images based on the **shape feature**, the system took **19.59 seconds** of average retrieval time to show the relevant images using a multi-feature CBIR model but the Universal model did not use the shape feature.
- When we compared the query images with the database images based on **texture feature**, the system took **121 seconds** of average retrieval time to show the relevant images using a multi-feature CBIR model but with the Universal model, it was **140 seconds** for texture feature.
- When we compared the query images with the database images based on **color and Texture feature**, the system took **133 seconds** of average retrieval time to show the relevant images using the multi-feature CBIR model but with the Universal model, it was **154 seconds** for the color feature.
- When we compared the query images with the database images based on **color and spatial layout feature**, the system took **151 seconds** of average retrieval time to show the relevant images using the multi-feature CBIR model but the Universal model did not use spatial layout feature.
- When we compared the query images with the database images based **on color, texture, shape and spatial layout feature**, the system took **292 seconds** of average

retrieval time to show the relevant images using the multi-feature CBIR model but the Universal model did not use this combination of features.

Hence, the results of our expanded multi-feature CBIR model show that the average run time for providing relevant images was improved and it is more efficient than the universal model as we are using four different features and their combinations.

4.4. Graphical User Interface

Below screenshots of designed GUI explains all the functionalities in detail about the Multi-Feature CBIR system. For a user it is very easy to work on the user interface which has been designed in MATLAB and MATLAB programming has been used for this. The meaning and purpose of the user interface are discussed below: [2], [3].



/home/manojkumar/Manoj/Query/T (5).jpg

/home/manojkumar/Manoj/Database

Figure 24: Snapshot of GUI (1)

How the system work is described in ‘help.txt’ file which opens when pushed this button.

This panel displays the name of the developer of the system.

When the "color" feature is selected this checkbox is to be checked.

When the "Texture" feature is selected this checkbox is to be checked.

When the "Shape" feature is selected this checkbox is to be checked.

The text box displays the total no of images of the same class present in the database.

The text box displays the total no of images retrieved correctly of the same class as of the query image present in the database [6].

The text box displays the Recall value of the system.

The text box displays the Precision value of the system.

The text box displays the time duration taken by the system to find out the result.

This panel shows the name of the designed system.

This text box displays the path of the query image.

Displays the path of the database.

A pop-up window is displayed to select the query image when pushed this button.

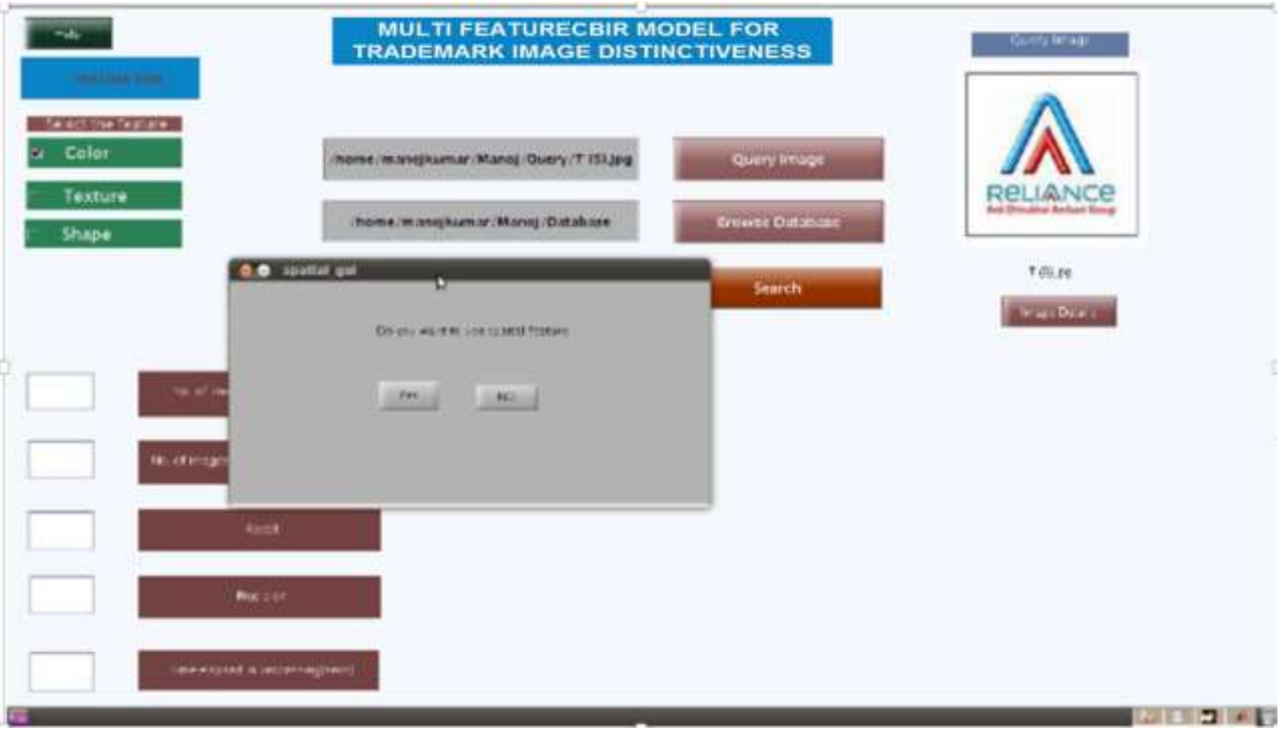
To select the database pop up window is displayed when pushed this button.

When this button has pushed the system processed to find the eight similar images present in the database to the query image [6].

Displays the input query image.

This push button invokes the imtool which shows the detail of the image.

The resulted output is displayed as eight similar images to the query image fetched by the system from the database.



/home/manojkumar/Manoj/Query/T (5).jpg

/home/manojkumar/Manoj/Database

Do you want to use spatial feature

Figure 25: Snapshot of GUI (2)

The above figure shows a pop-up window which popped up when the color feature is checked which asked the user whether it is interested to select the spatial layout feature or not.

4.5. System Specification

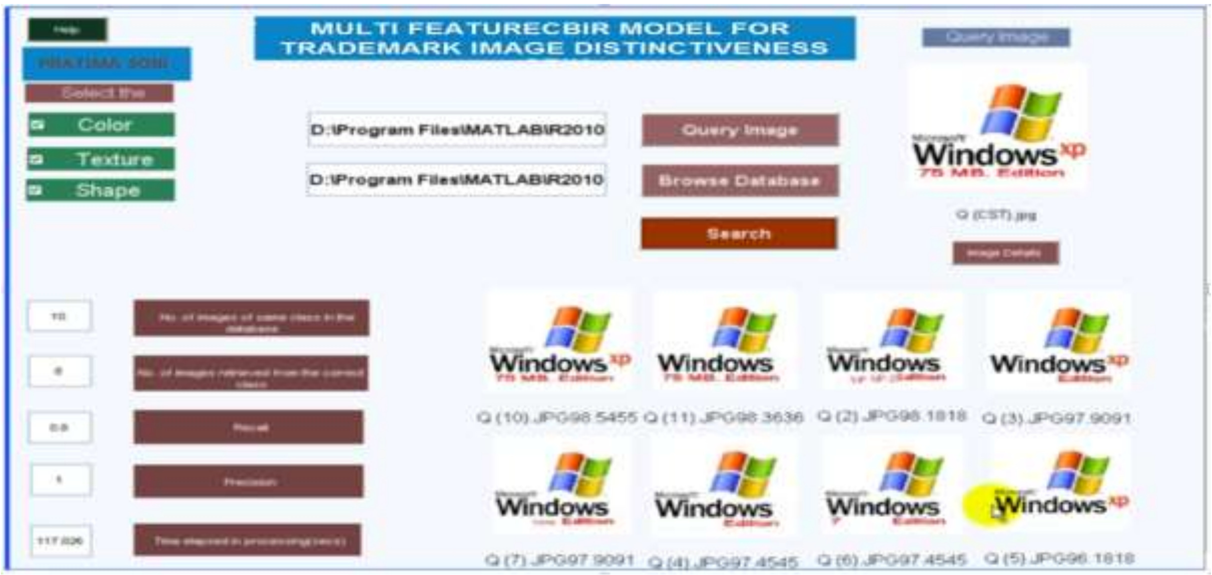
Table 4: System Requirements Tabulation

Sr. No	Feature Description	System Specification
1	Size of Database	175
2	System loading time	Real Time
3	Image format supported	Jpeg/jpg
4	Average Retrieval time with color alone	2.51 sec
5	Time is taken for Retrieval of images with shape alone	19.59sec
6	Time is taken for Retrieval of images with texture alone	121 sec
7	Average Retrieval time with spatial layout + color	151 sec
8	Retrieval time with texture+ shape + color+ spatial layout	292 sec
9	Retrieval time with color+ spatial layout+ texture	277 sec
10	No. of images retrieved	8
11	Average Recall	61.75
12	Average Precision	64.1

4.6. Hardware and Software Requirement

- Operating System- Ubuntu, Windows XP/ Windows 7 or advance.
- Tools/ Software- MATLAB 7.6.0 (R2008a) or advance.
- Processor- Intel or AMD 32/64 bits processor
- Primary memory- Minimum 1GB RAM
- Secondary memory-1 GB External memory for MATLAB product. 3-4 GB for a typical installation.

4.7. Result Snapshots



D:\Program Files\MATLAB\R2010

D:\Program Files\MATLAB\R2010

10 No. of images of same class in the database

8 No. of images retrieved from the correct class

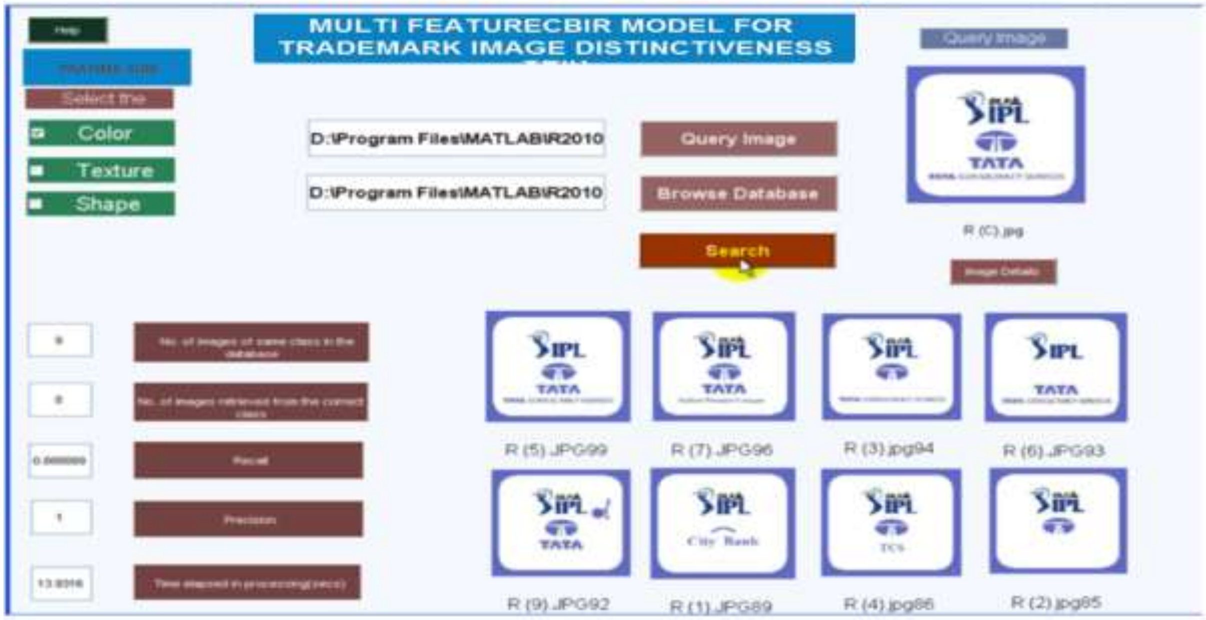
0.8 Recall

1 Precision

117.026 Time elapsed in processing(secs)

Figure 26: Snapshot of Result (1)

The above figure shows how the interface looks with an input query image and with all features selected.



D:\Program Files\MATLAB\R2010

D:\Program Files\MATLAB\R2010

9 No. of images of same class in the database

8 No. of images retrieved from the correct class

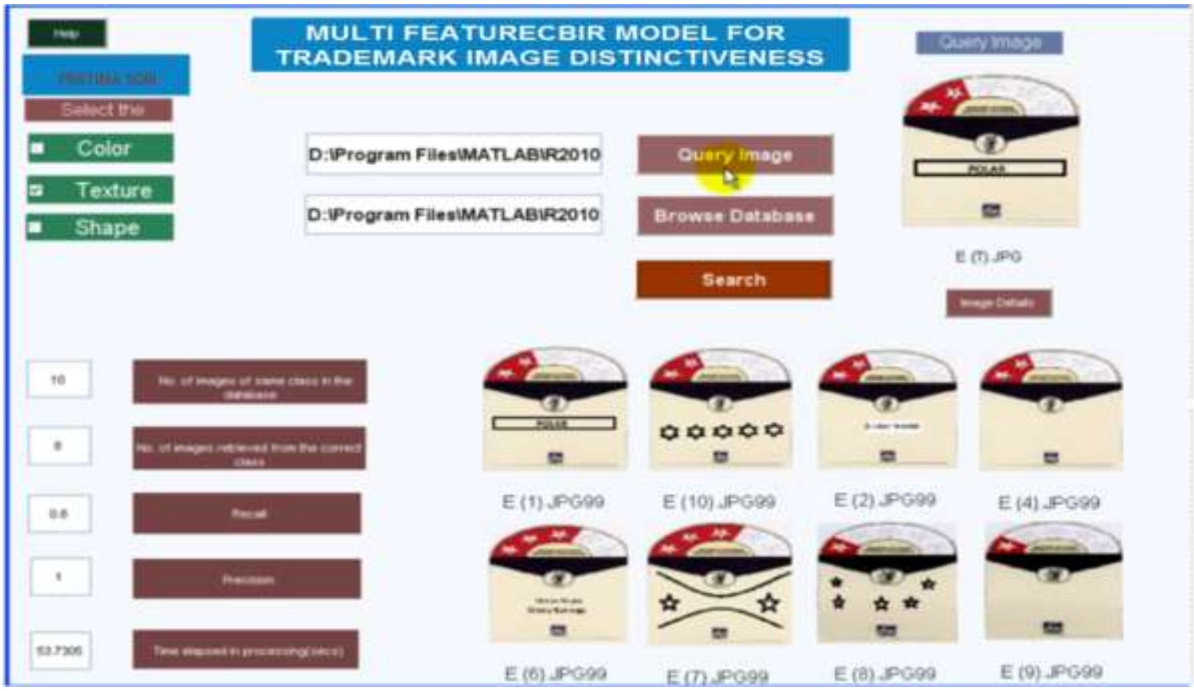
0.888889 Recall

1 Precision

13.8316 Time elapsed in processing(secs)

Figure 27: Snapshot of Result (2)

The above figure shows an input query with an only Color feature selected.



Select the

□ Color

☑ Texture

□ Shape

D:\Program Files\MATLAB\R2010

D:\Program Files\MATLAB\R2010

10 No. of images of same class in the database

8 No. of images retrieved from the correct class

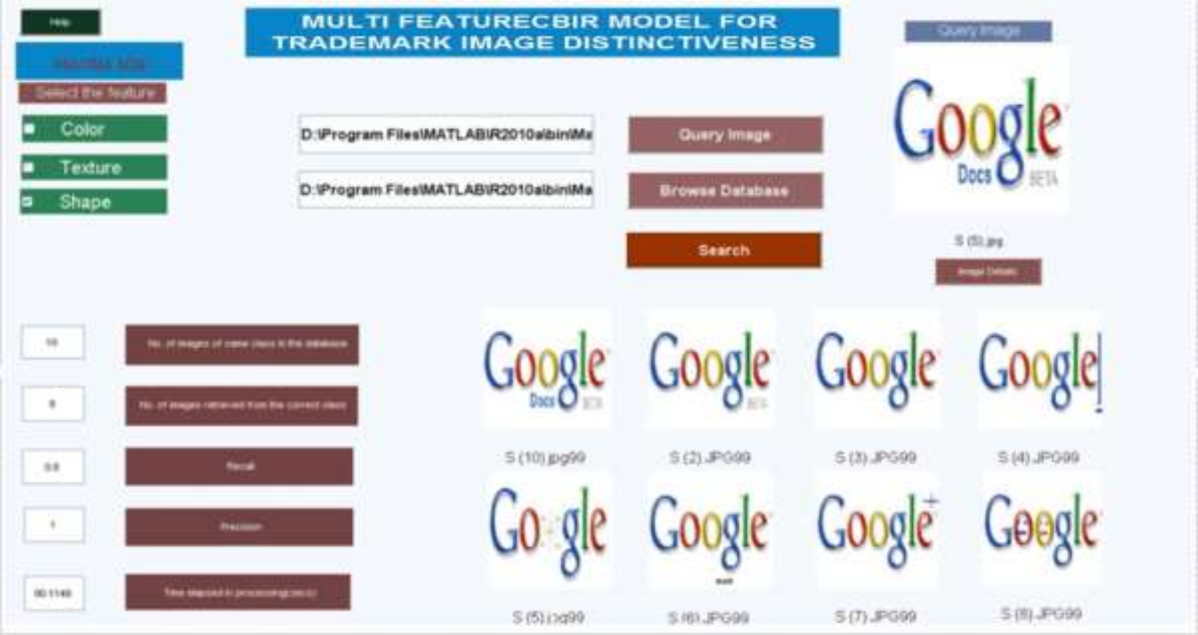
0.8 Recall

1 Precision

53.7306 Time elapsed in processing(secs)

Figure 28: Snapshot of Result (3)

The above figure shows an input query with only the Texture feature selected.



Select the feature

□ Color

□ Texture

☑ Shape

D:\Program Files\MATLAB\R2010a\bin\Ma

D:\Program Files\MATLAB\R2010a\bin\Ma

10 No. of images of same class in the database

8 No. of images retrieved from the correct class

0.8 Recall

1 Precision

60.1149 Time elapsed in processing(secs)

Figure 29: Snapshot of Result (4)

The above figure shows an input query with only the Shape feature selected.

5. CONCLUSION AND FUTURE WORK

5.1. The Conclusion of the Work

This paperwork suggests the prevention of trademark infringement by improving the trademark image retrieval system by providing write-ups that extended the Universal CBIR Model. Previously for any new trademark registration, manual similarity matching has been done with the help of the Universal CBIR Model which is cumbersome and time-consuming task [3]. The weighted approach makes the system efficient compared to those which give equal priority to all feature values of the image [28]. This model is comprised of four features namely color, texture, shape and the important one spatial layout of color with the facility of flexible feature techniques and prioritization approach [25]. With the help of these write-ups, computational work and calculations performed, recall and precision values show the performance of the system and its success for finding out similar trademarks images [2], [4], [9]. With the help of comparison over the existing Universal Model values, we also found that the average response time of the software using the Multi-Feature model is also increased.

Instead of good results still, there is a need for more research work to enhance the performance and efficiency of the Multi-Feature CBIR Model. But the objective of this paperwork is very well achieved with satisfactory results with the possibility of future work [18].

5.2. Future Work

User feedback will play an important role in future extensions of the reported work. More overextending this work with the other possible set of different feature techniques and more flexibility of database images will provide better results and advancement of the Multi-Feature CBIR Model. Higher recall and precision values are expected after doing this.

If images in the database are well classified then there is a tremendous decrement in time for searching a particular image in a huge database. So automatically database classification of the images and insertion of new images in the database will be a challenging task for future work.

REFERENCES

[1] S. Nandagopalan, Dr. B. S. Adiga, and N. Deepak, “A Universal Model for Content-Based Image Retrieval” World Academy of Science, Engineering and Technology, Vol. 46, Issue 1, 2008, pp. 141-151.

[2] J. C. Wu, C. P. Lam, B. E. Mehtre, Y.J. Gao, A. D. Narasimhalu, “Content-based retrieval for trademark registration” Multimedia Tools and Applications, Vol. 3, Issue 3, November 1996, pp. 245-267.

[3] J. P. Eakins, M.E. Graham, and J.M. Boardman, “Evaluation of a trademark image retrieval system” Information Retrieval Research, the 19th Annual BCS-IRSG Colloquium on IR Research, Vol. 4, Issue 5, April 08-09, 1997, pp. 210-260.

[4] Heng Qi, KeqiuLi, Yangmingshan, WenyuQu, “An effective solution for trademark image retrieval by combining shape description and feature matching” Elsevier Ltd, Pattern Recognition, Vol.43, Issue 6, June 2010, pp. 2017–2027.

[5] Smith, J., and Chang, S.-F, “VisualSEEk: A Fully Automated Content-based Image Query System” ACM Multimedia, Boston, Vol. 96, Nov 18 – 22, 1996, pp. 87-98.

[6] M. Flickner, H. Sawhney, W. Niblack, “Query by image and video content” Proceedings of IEEE, Vol. 28, Issue 9, Sep 1995, pp. 23-32.

[7] R M Harlick, K. Shanmugam, and I. Dinstein, “Textural features for image classification” IEEE Trans. Sys. Man Cybern, Vol. 3, Issue 6, Nov 1973, pp. 610-621.

[8] F. Long, H, Zhang, H. Dagan, “Fundamentals of content-based image retrieval,” Multimedia Information Retrieval and Management, Technological Fundamentals and Applications, Berlin, Heidelberg, New York, Vol. 8, Issue 2, 2003, pp. 1-26.

[9] Aarti Kochhar, Amanbir Sandhu, “Content-Based Image Retrieval using Texture, Color and Shape for Image Analysis” International Journal of Computers & Technology, Vol. 3, Issue 1, Aug 2012, pp 300-326.

[10] N. S. Vassilieva, “Content-based Image Retrieval Methods” ISSN 0361-7688, Programming and Computer Software, Vol. 35, Issue 3, May 2009, pp 158-180.

[11] Ranjeet Kumar, R.C. Tripathi, M.D. Tiwari “A Comprehensive Study on Content-based Trademark Retrieval System” International Journal of Computer Applications (0975 – 8887), Vol. 13, Issue 6, January 2011.

[12] W.Y. Ma, and B.S Manjunath, “Netra: A Toolbox for Navigating Large Image Databases” Proc. of the Int.Conf. On Image Processing, Santa-Barbara, CA, USA, Vol. 1, Oct 26-29, 1997, pp 568–571.

[13] Rajshree S. Dubey, RajnishChoubey, Joy Bhattacharjee, “Multi-Feature Content-Based Image Retrieval” International Journal on Computer Science and Engineering, Vol. 02, Issue 06, 2010, pp 2145-2149.

[14] Toshikazu Kato, “Database Architecture for content-based image retrieval" International Society of Optical Engineering, April 1, 1992, pp. 112-123.

[15] Mohamed Abdel-Mottaleb, “Image Retrieval Based on Edge Representation” International Conference on Image Processing, IEEE, Vancouver, BC, Canada, Vol.3, 10-13 Sept. 2000.

[16] Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing, 3rd Ed., Pearson. Retrieved September 26, 2018, from https://www.academia.edu/8240316/Digital_Image_Processing_Gonzaleze_and_Woods_3rd_Ed._Pearson

[17] Mr. Kondekar V. H., Mr. Kolkure V. S., Prof.Kore S.N, “Image Retrieval Techniques based on Image Features: A State of Art approach for CBIR” International Journal of Computer Science and Information Security, Vol. 7, Issue 1, 2010.

[18] Image Retrieval. (n.d.). In Wikipedia. Retrieved August 18, 2018, from en.wikipedia.org/ wiki/Image retrieval

[19] Chromaticity Diagram. (n.d.). Retrieved January 12, 2018, from www.efg2.com /Lab/ Graphics/Colors/Chromaticity.htm

[20] Introduction to Content-Based Image Retrieval tutorial. (n.d.). Retrieved May 10, 2017, from www.nonrelevant.net/2010/11/introduction-to-content-based-image-retrieval-tutorial/

[21] Swati Agrawal, A.K. Verma, Preetvanti Singh, “Content-Based Image Retrieval Using Discrete Wavelet Transform Edge Histogram Descriptor” International Conference on Information Systems and Networks, IEEE, Mathura, India, Vol. 1, Issue 6, 6 June 2013.

[22] Tianhorng Chang and C.-C. Jay Kuo, Senior member IEEE, “Texture analysis and classification with tree-structured wavelet transform” IEEE transactions on image processing, Vol. 2, Issue 4, October 1993, pp. 429-441.

[23] Content-Based Image Retrieval System. (n.d.). In Wikipedia. Retrieved from March 15, 2017, from https://en.wikipedia.org/wiki/Content-based_image_retrieval

[24] Jain, A.K., & Vailaya, "Shape-based retrieval: A case study with trademark image databases" IEEE Transactions on Pattern Recognition, Vol. 31, Issue 9, September 1998, pp. 1369-1390.

[25] Akriti Nigam, Arpit Kumar Garg, R.C. Tripathi, “Content-Based Image Retrieval by integrating shape with color and texture information” International Journal of Computer Application, Vol. 22, Issue 7, May 2011.

[26] Shih-Fu Chang, Thomas S. Huang, Yong Rui, "Image retrieval: promising directions, current techniques, and open issues" Journal of Visual Communication and Image Representation, Vol.10, Issue 1 March 1999, pp. 39–62.

[27] R H Goudar, Neeshu Chaudhary, Roshi Choudhary, Nikita Raina, Rashmi Chauhan "An Integrated Approach to Content-Based Image Retrieval" International Conference on Advances in Computing, Communications and Informatics, IEEE Xplore, New Delhi, India, Vol. 5, Issue 5, Dec 1, 2014.

[28] K. Hemachandran, S. Mangijao Singh, "Content-based image retrieval using color moment and texture feature" International Journal of Computer Science Issues, Vol. 9, Issue 5, September 2012.

[29] Neha Sharma, "Retrieval of the image by combining the histogram and HSV features along with algorithm" International Journal of Engineering Trends and Technology, Vol-4, Issue 7, July 2013.

[30] Gonzalez, R. C., & Woods, R. E. (2002). Digital Image Processing, 2nd Ed., Chapter-6, Pearson. Retrieved May 20, 2018, from https://www.academia.edu/8240316/Digital_ Image_Processing_Gonzaleze_and_Woods_3rd_Ed._Pearson

[31] 2D Daubechies Wavelets. (n.d.). Retrieved April 26, 2018, from http://www.mathworks.com /matlabcentral/mlcdownloads/downloads/submissions/9554/versions/5/previews/numerical-tour/wavelet_daubechies2d/index.html

[32] High-Pass Filtering. (n.d.). Retrieved July 20, 2018, from www.diffractionlimited.com/help/ maximdl/High-Pass_Filtering.htm