OCTOBER, 2017 M.Sc. in Electronics and Computer Engineering HERSH HAMA

HASAN KALYONCU UNIVERSITY

GRADUATE SCHOOL OF

NATURAL & APPLIED SCIENCES

A COMPARISON STUDY ON IMAGE CONTENT BASED RETRIEVAL SYSTEMS

M. Sc. THESIS

IN

ELECTRONICS AND COMPUTER ENGINEERING

BY

HERSH HAMA

OCTOBER 2017

A Comparison Study on Image Content Based Retrieval Systems

M.Sc. Thesis

In

Electronics and Computer Engineering

Hasan Kalyoncu University

Supervisor

Assist. Prof. Dr. Saed ALQARALEH

by

Hersh HAMA

October 2017

REPUBLIC OF TURKEY
HASAN KALYONCU UNIVERSITY
GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES
ELECTRONICS AND COMPUTER ENGINEERING

Name of the thesis: A Comparison Study on Image Content Based Retrieval Systems

Name of the student: Hersh Mustafa Hama HAMA

Exam date: October, 25, 2017

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Mehmet KARPUZCU
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Bülent HAZNEDAR
Head of Department

This is to certify that we have read this thesis and that in our consensus/majority opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Saed ALQARALEH
Supervisor

Examining Committee Members	Signature
Assist. Prof. Dr. Saed ALQARALEH	……………………..
Assist. Prof. Dr. Bülent HAZNEDAR	……………………..
Assoc. Prof. Dr. Sema Koç KAYHAN	……………………..

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Hersh HAMA

ABSTRACT

A COMPARISON STUDY ON IMAGE CONTENT BASED RETRIEVAL SYSTEMS

HAMA, Hersh Mustafa Hama

M.Sc. in Electronic and Computer Engineering

Supervisor: Assist. Prof. Dr. Saed ALQARALEH

October, 2017, 62 pages

In recent years, multimedia searching has become an important research field. Multimedia files are one of the most important materials on the internet. Unfortunately, even for the state-of-the-art methods and applications based on the access to multimedia on the internet, it is hard to find the required multimedia. The main purpose of this study is to investigate the performance of well-known image content-based retrieval techniques, i.e., Fuzzy Color and Texture Histogram (FCTH), Edge Histogram Descriptor (EHD), Scalable Color Descriptor (SCD), Color Layout Descriptor (CLD), Color and Edge Directivity Descriptor (CEDD), and Speed-Up Robust Feature (SURF) combined with Fast Library Approximate Nearest Neighbor (FLANN). The objective of using these techniques is to find the query's most relevant files and list them at the top of the retrieval list. Several experiments have been conducted and it has been observed that FCTH and SCD outperform other studied techniques. On the other hand, for the SURF combined with FLANN approach, the results of most of the queries were below user expectations. In addition, extracting the feature vectors using this method requires massive amount of memory. Overall, none of the studied CBIR descriptors can be used individually to build a full image retrieval system. In our opinion, multiple descriptors can be used to achieve a more robust system and accurate results.

Keywords: Multimedia search engines, information retrieval, re-ranking algorithm, query by example.

ÖZET

RESİM İÇERİK TABANLI ALMA SİSTEMLERİNE İLİŞKİN BİR KARŞILAŞTIRMA ÇALIŞMASI

HAMA, Hersh Mustafa Hama
Yüksek Lisans Tezi, Elektronik ve Bilgisayar Mühendisliği
Tez Yöneticisi: Yrd.Doç. Dr. Saed ALQARALEH
Ekim, 2017, 62 sayfa

Son yıllarda multimedya arama önemli bir araştırma alanı haline geldi. Multimedya dosyaları, internetteki en önemli materyallerden biridir. Maalesef, internette multimedya erişimine dayanan en yeni yöntem ve uygulamalar için bile, gerekli multimedya bulmak zor. Bu çalışmanın temel amacı, Fuzzy Color ve Texture Histogram (FCTH), Edge Histogram Descriptor (EHD), Scalable Color Descriptor (SCD), Color Layout Descriptor (CLD), Color and Edge Directivity Descriptor (CEDD), ve Speed-Up Robust Feature (SURF) ve Fast Library Approximate Nearest Neighbor (FLANN) ile birleştirilmiş alanda bilinen resim îçerik tabanlı alma sistemlrinin performanslarınn incelenmesidir. Bu teknikleri kullanma amacı, sorgunun en alakalı dosyalarını bulmak ve bunları alma listesinin en üstünde listelemektir. Çeşitli deneyler yapılmış ve FCTH ve SCD'nin diğer incelenen tekniklerden daha iyi performans sergilediği görülmüştür. Diğer yandan, SURF ile FLANN yaklaşımı birleştirildiğinde, sorguların çoğunun sonuçları kullanıcı beklentilerinin altında kaldığı görülmüştür. Buna ek olarak, bu yöntemi kullanarak özellik vektörlerinin çıkarılması muazzam bir bellek gerektirir. Genel olarak, çalışılan CBIR tanımlayıcılarından hiçbiri tam bir görüntü alma sistemi oluşturmak için tek tek kullanılamaz. Görüşümüze göre, daha sağlam bir sistem ve doğru sonuçlar elde etmek için çoklu tanımlayıcılar kullanılabilir.

Anahtar Kelimeler: Multimedya arama motorları, bilgi getirim, yeniden sıralama algoritması, örnekle sorgulama.

To My Parents

ACKNOWLEDGEMENTS

To begin with I would like to thank the “ALLAH” for the virtues of this blessing for implanting the soul of endurance and faith in myself to complete this study.

It is a please to express my deepest gratitude to my supervisor Assist. Prof. Dr. Saed ALQARALEH for his guidance, advice, criticism, continuous encouragements, insight, valuable enthusiastic and unfailing advice throughout the present study.

Special thanks are served for Assoc. Prof. Dr. Sema Koç KAYHAN and Assist. Prof. Dr. Bülent HAZNEDAR for serving on committee and their contributions, suggestions, and comments to improve the quality of thesis.

I want to thank Asst. Prof. Dr. Abdul Hafez ABDUL HAFEZ for his kind help and suggestion to improve the quality of thesis.

Finally, I would like to thank my family and friends to support, encouragement and contribution they have made for my research to be successful.

TABLE OF CONTENT

ABSTRACT .... **v**

ÖZET .... **vi**

ACKNOWLEDGEMENTS .... **viii**

TABLE OF CONTENT .... **ix**

LIST OF FIGURES .... **xi**

LIST OF TABLES .... **xiii**

LIST OF SYMBOLS/ ABBREVIATION .... **xiv**

CHAPTER 1: INTRODUCTION **1**

1.1. Development of Multimedia Search Engines 1

1.2. Main Problems in Multimedia Retrieval 2

1.3. Thesis Organization 3

1.4. Summary 3

CHAPTER 2: RELATED WORKS **4**

2.1 Internet Search Engine 4

2.1.1 Image Retrieval System 5

2.1.2 Multimedia Search Re-ranking 9

2.2 Literature Review 12

2.2.1 Query-Specific Semantic Signature 12

2.2.2 Ranked List Similarity 13

2.2.3 Large Scale Retrieval and Generation of Image Descriptions 13

2.2.4 Prototype Based Re-ranking 14

2.2.5 Exploiting Click Constraints and Multi-view Features for Image Re-ranking 14

CHAPTER 3: CONTENT BASED APPROACHES **15**

3.1 Overview 15

3.2 Feature Extraction 15

3.2.1 Fuzzy Color and Texture Histogram (FCTH) 15

3.2.2 Edge Histogram Descriptor (EHD) 17

3.2.3 Scalable Color Descriptor (SCD) 19

3.2.4 Color Layout Descriptor (CLD) 20
3.2.5 Color and Edge Directivity Descriptor (CEDD) 21
3.2.6 Speed Up Robust Features (SURF) Combined with Fast Library Approximate Nearest Neighbor (FLANN) 22
3.3 Similarity Measurement 23
3.3.1 Tanimoto Coefficient 23
3.3.2 K-Nearest Neighbors 24
3.3.3 Euclidean Distance 24
3.4 Software Environment 24
CHAPTER 4: PERFORMANCE STUDIES 26
4.1. Overview 26
4.2. Experiments on First Database: 26
4.3. Experiments on Second Database: 33
4.4. Experiments on Third Database: 41
4.5. Performance of SURF combing with FLANN 47
4.6. Overall Result Discussion 53
CHAPTER 5: DISCUSSION AND CONCLUSION 56
REFERENCES 59

LIST OF FIGURES

Figure 2.1 [11]: Content Based Image Retrieval System 6

Figure 2.2 [26]: Illustrates a system that integrates three types of image search engines with re-ranking algorithms 11

Figure 3.1 [1]: Block diagram of fuzzy color and texture histogram technique 16

Figure 3.2: Decomposition of the blocks into frequency bands by wavelet transforms 17

Figure 3.3 [35]: Defining sub-images and image-blocks using EHD 18

Figure 3.4 [35]: Five types of edge in EHD 18

Figure 3.5 [3]: A schematic diagram of SCD generation.. 20

Figure 3.6 [4]: A block diagram of the CLD extraction 20

Figure 3.7 [34]: The block diagram of CEDD 21

Figure 4.1: Image queries for the first database 27

Figure 4.2: Number of relevant images for the first database and the compared descriptors using the first 10 results 32

Figure 4.3: Number of relevant images for the first database and the compared descriptors using the first 20 results 32

Figure 4.4: Number of relevant images for the first database and the compared descriptors using the first 30 results 33

Figure 4.5: Image queries for the second database 34

Figure 4.6: Number of relevant images for the second database and the compared descriptors using the first 10 results 39

Figure 4.7: Number of relevant images for the second database and the compared descriptors using the first 20 results 40

Figure 4.8: Number of relevant images for the second database and the compared descriptors using the first 30 results 40

Figure 4.9: Image queries for the third database 41

Figure 4.10: Number of relevant images for the third database and the compared descriptors using the first 10 results 46

Figure 4.11: Number of relevant images for the third database and the compared descriptors using the first 20 results 47
Figure 4.12: Number of relevant images for the third database and the compared descriptors using the first 30 results 47
Figure 4.13: Image queries for testing the SURF combined with FLANN and the first database.. 49
Figure 4.14: Image queries for testing the SURF combined with FLANN and the second database.. 50
Figure 4.15: Image queries for testing the SURF combined with FLANN and the third database 52
Figure 4.16: Retrieval results based on image query of category ‘Cow’ using EHD 53
Figure 4.17: Sample images of category ‘Chair'... 55

LIST OF TABLES

Table 3.1 [2]: Semantic of Local edge bins in EHD 19
Table 4.1: Number of relevant files using the FCTH method and first database....... 28
Table 4.2: Number of relevant files using the EHD method and first database......... 29
Table 4.3: Number of relevant files using the SCD method and first database 30
Table 4.4: Number of relevant files using the CLD method and first database 31
Table 4.5: Number of relevant files using the CEDD method and first database 31
Table 4.6: Number of relevant files using the FCTH method and second database .. 35
Table 4.7: Number of relevant files using the EHD method and second database 36
Table 4.8: Number of relevant files using the SCD method and second database 37
Table 4.9: Number of relevant files using the CLD method and second database 38
Table 4.10: Number of relevant files using the CEDD method and second database 39
Table 4.11: Number of relevant files using the FCTH method and third database ... 42
Table 4.12: Number of relevant files using the EHD method and third database 43
Table 4.13: Number of relevant files using the SCD method and third database 44
Table 4.14: Number of relevant files using the CLD method and third database...... 45
Table 4.15: Number of relevant files using the CEDD method and third database ... 46
Table 4.16: Number of relevant files using the SURF combined with FLANN and the first database. 49
Table 4.17: Number of relevant files using the SURF combined with FLANN and the second database... 51
Table 4.18: Number of relevant files using the SURF combined with FLANN and the third database 52

LIST OF SYMBOLS/ ABBREVIATION

CBIR	Content-Based Image Retrieval
CEDD	Color and Edge Directivity Descriptor
DCT	Discrete Cosine Transform
CLD	Color Layout Descriptor
DoH	Determinant of Hessian Matrix
EHD	Edge Histogram Descriptor
FCTH	Fuzzy Color and Texture Histogram
FLANN	Fast Library Approximate Nearest Neighbor
HSV	Hue, Saturation, and Value
KNN	K-Nearest-Neighbors
LMF	Linear Multimodal Fusion
HH	High-High
HL	High-Low
LH	Low- High
LL	Low-Low
MHL	Multi-view Hyper-graph based Learning
QBE	Query-By-Example
QE	Query Expansion

RGB	Red, Green, and Blue
SCD	Scalable Color Descriptor
SURF	Speed-Up Robust Features
TBIR	Text-Based Image Retrieval

CHAPTER 1

INTRODUCTION

Search engines are websites that have been designed to help users to find out the required information. In general, search engines require the user to write the query keyword (s) into the search box. Then, search engines try to find the candidate results and finally order and rank these outcomes to show the most relevant ones at the top of the results list.

Internet search engines are especially supposed to include more common documents, valuable information, and specific sites and pages**.** The commercial web search engines are working with billions of web pages to retrieve relevant information. In the search engine, the priority is to discover more than one document or page with the user's query. The relevance is the fundamental concept behind information retrieval, because the text of a query keyword of any given user can be an exact match within the text of many documents.

1.1. Development of Multimedia Search Engines

In past decades, videos and images have been developed with the help of social media, and the growth of digital devices. This development played a significant role in the ever increasing use of multimedia data. Indeed electronic communication has advanced rapidly in recent years, mostly because of evolution of information technology.

Visual content is one of the components of multimedia data and contains digital images. As a result, some multimedia search engines have been developed. In general, multimedia search engines can categorized into: First, text based image retrieval (TBIR), this type try to find relevant files based on the query keyword (s). TBIR is used widely in multimedia search engines and it is generally very fast and easy to implement. However, in some cases, captions of multimedia files and/ or link tagging may not be related to the contents of the file itself. This might lead the searcher to obtain irrelevant and undesired multimedia files. Second CBIR, which extract features that describes visual content of multimedia. Content-Based is refers to the shape, texture, color, or any other feature that can be achieved from the content of the images rather than the text. The right features play a significant role in the retrieval system and these features of the images as accuracy and uniquely as suitable. Moreover, the features chosen to describe the objects of the multimedia file need to be discriminative. Overall, the CBIR systems used color, texture, and shape features as three basic means to index multimedia files.

1.2. Main Problems in Multimedia Retrieval

In this thesis, we have investigated the performance of image search engines. Following is a summary of the main problems related to image search engines:

First, the primary challenge for retrieving images using TBIR is that it uses query keywords and surround text such as filenames and content of page that includes the image. This might lead the results of queries to contain irrelevant files.

Second, image web-based searches are neglecting the content of images, oftentimes irrelevant documents are cluttering the results of querying for a specific object.

Third, most CBIR systems are using color, texture or both features to retrieve the intended images from the web and the databases system. However, the shape of objects in images also plays an important role to find similar objects in the images. But, most descriptors are not dependent on shape features because in image retrieval expectations are that the shape description is consuming time, complex processes, and invariant to translation, rotation, and scaling of the object.

Moreover, it is very difficult to achieve the most relevant files using only a single feature type. As a feature work, the study present in this thesis can be further improved by testing the performance of integrating multiple descriptors and building an image retrieval system which uses the best group of descriptors simultaneously.

1.3. Thesis Organization

The remaining of the thesis is organized as follows: Chapter 2 provides the mechanism of search engines. Chapter 3 introduces a comprehensive survey on content-based image retrieval techniques. Experimental studies are presented in Chapter 4, and finally, conclusions are given in Chapter 5.

1.4. Summary

This study has compared the accuracy of six CBIR image techniques. These CBIR techniques are Fuzzy Color and Texture Histogram (FCTH) [1], Edge Histogram Descriptor (EHD) [2], Scalable Color Descriptor (SCD) [3], Color Layout Descriptor (CLD) [4], Color and Edge Directivity Descriptor (CEDD) [5], and Speed-Up Robust Feature (SURF) combined with Fast Library Approximate Nearest Neighbor (FLANN) [6].

As results of this study, FCTH and SCD descriptors were selected as the best descriptors. Hence, they have obtained the most relevant files compared to the other descriptors. On the other hand, for the SURF combined with FLANN approach, the results of most of the queries were below user expectations. In addition, extracting the feature vectors using this method requires a massive memory. Overall, none of the studied CBIR descriptors can be used individually to build a full image retrieval system. In our opinion, multiple descriptors can be used to achieve a more robust system and accurate results.

CHAPTER 2

RELATED WORKS

2.1 Internet Search Engine

Search engines are websites that have been designed to help users to find out the required information. Thus, they can be considered as the backbone of the Internet. Internet search engines are helpful for users to find required information and it has the ability to provide users with documents, files, valuable information, web sites and pages.

Based on [7], the mechanism of search engines is briefly summarized below:

First of all, collecting the website's information is done by a software program which is called "Spider" also known as Crawler, and Robot [7]. Spiders start by visiting a list of websites, and then it downloads a copy of the website pages, and then follows every link extracted from these downloaded pages. In this way, the crawling system rapidly travels, spreading out across the most widely used parts of the Internet.

Secondly, the indexing [7], which is a process of analyzing all web pages that the crawler finds, and storing a copy of every page with related information, such as keywords, metadata, multimedia files, etc., in a huge database(s), this process is also sometimes called cataloging.

On the other hand, when users enter a query keyword (s) into the search box, search engine [7, 8] tries to find relevant files from those pages recorded in the index. Then it ranks the results according to which documents is the most relevant to the query. In the search engines that are especially supposed to include more common documents, valuable information, sites or pages, this assumption has proven fairly successful in terms of people's overall gratification with search results. Popularity and relevance are determined automatically by employing mathematical algorithms to sort results of relevance, and then to rank the results according to popularity.

2.1.1 Image Retrieval System

In the last decade, multimedia systems have rapidly been developed and widespread accessibility of visual content has created a surge in research activities related to visual searches. Multimedia retrieval is widely needed and it is one of the larger problems of user concern. The key issue is retrieval of visual documents (such as web pages including images, videos and images), that are relevant to a given query or intention of people on the internet. There are two main problems in generating an efficient multimedia search technique. First, how to represent queries and index visual documents, second, how to map the representations of queries and visual documents and find the relevance between queries and visual documents. An effective and efficient system is required for managing huge databases. In the following, the two main image retrieval systems are summarized:

2.1.1.1 Text-Based Image Retrieval (TBIR)

In this technique a user is required to enter a query keyword(s) as a text to obtain images from database systems. Then, a search engine returns relevant images in a ranked list that contains tags of the query keyword, and the score of ranking is done according to a similar measurement between the keyword and the textual features of relevant images [9]. Text based image retrieval has been used widely in multimedia search engines and it can be easy implemented. The main challenge for retrieving images using TBIR is that multimedia databases are built using the images surrounding text, such as filenames, metadata, link tags and content of the web pages that contain the indexed image. However, in some cases, multimedia surrounding text has no relation to the contents of the files itself. This might lead to obtaining irrelevant, duplicate, and undesired multimedia files.

2.1.1.2 Content-Based Image Retrieval (CBIR)

Content based search engines (CBIR), which are used to find visual information on the Internet based on the content of visual documents has now been developed. CBIR is employing visual content of images, such as color, texture, and shape [10], to retrieve the relevant images from the web and the databases system. Query-by-example (QBE) is a query technique that allows the user to search for multimedia files based on an example. This technique can be used when a user has an image, and he/ she is looking for similar files.

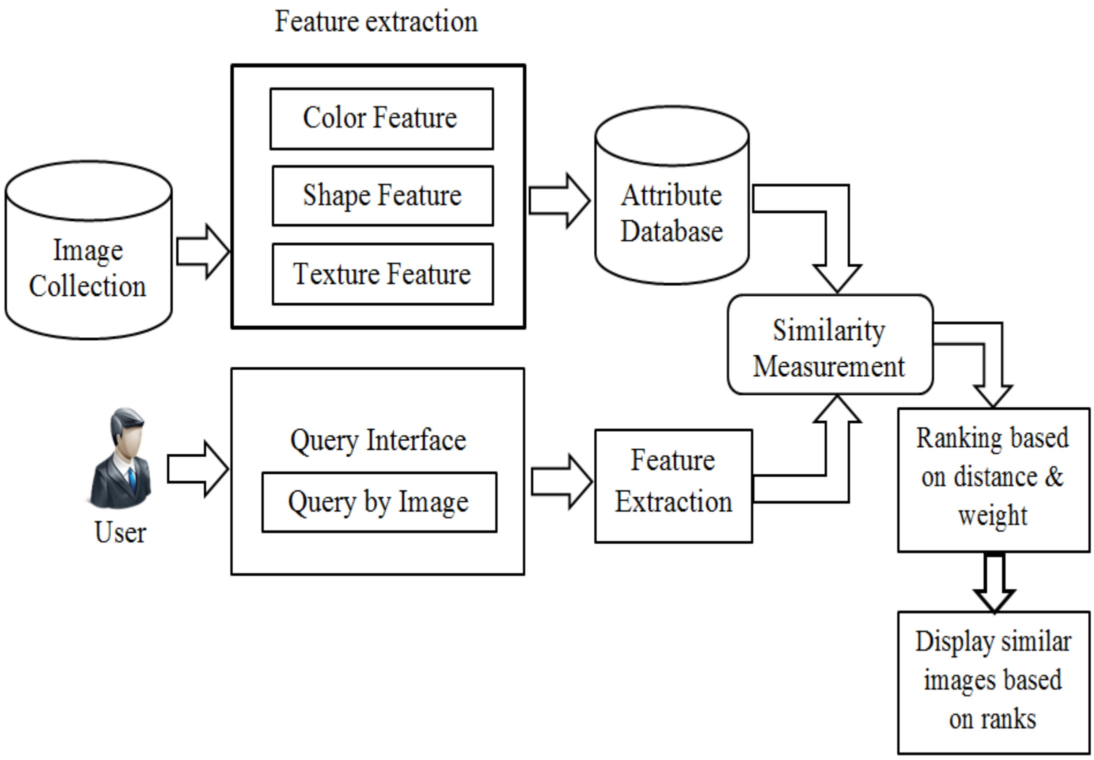

Figure 2.1 [11]: Content Based Image Retrieval System.

As shown in Figure 2.1 [11, 12], the CBIR starts by extracting and storing the images features into a features database. Users are asked to upload an image then the system will extract the same set of features used to build the database from the uploaded image. The similarities/distances calculated between the feature vectors of an uploaded image and database images; a small distance means more similarity and relevant. On the other hand, images that have more distance are called irrelevant. The top ranking list of images contains relevant images and the irrelevant images are shown at the bottom of the list.

Mainly, Multimedia features are categorized as: color, texture, and shape features, and its details are summarized next:

a) Color Feature

Color feature [10], is widely used in images retrieval because of its simplicity and it does not rely on an image's orientation or size. However, the intensity of light and camera viewpoint plays an important role in the relevance of query results. In the following; two different color feature techniques are summarized:

Color histogram [12], is a representation of the distribution of colors in an image. The image will be represented by a number of vectors, which is done based on the used color-space such as RGB (Red, Green, and Blue). For instance, for the RGB color system, the number of vectors is three, where each vector is representing the number of pixels that have a specific color. Although, it is the most common method because of it is simplicity to compute, it requires the compared images to have the same scale. In addition, it is sensitive to the changes in the camera view.

Color moment [13, 14], is another type of color feature that divides an image into several blocks. The color moments of the image blocks are exported and they are clustered into a number of classes based on the algorithm of a fast non-iterative clustering [14]. The central vector of each class is considered as a primitive of the image and these are used as feature vectors. The similarity among color moments is measure by Euclidean Distance.

b) Texture Feature

In general, texture feature [15], is the process of integrating features such as smoothness, coarseness, and regularity. Additionally, it refers to visual patterns and plays a significant role in people's interpretations and visual perceptions. Texture features can be categorized into the following three techniques. 1) Statistical approaches [16], supposes texture by averages (means) of statistical gray level properties of image points. 2) Spectral methods [16], are based on global periodicity of the grey levels of a surface in frequency domain and power density function. And 3) Structural methods [16], where texture features consist of texture elements that are called "texels". These texture elements are then organized on a surface based on some specific placement rules.

Gabor filter [17, 18], is a one of the most popular techniques for extracting texture features from images. In addition, it is an efficient method that can be used in image retrieval and classification systems. Gabor filters [19], consist of a bank of wavelets, that combine together and each wavelet captures energy at a particular scale and in a particular direction. It is provides a localized frequency description and captures the energy/ local features of the signal. Then, texture information can be exported from collected energy distributions.

The main advantage of texture features is that they are able to distinguish between the objects of the images that have the same shape and color such as snow and cotton or cloud.

c) Shape Feature

Shape is referred to a particular region in an image which contains all the geometrical visual information that could be sought out [20]. Shape feature extraction includes two major steps; object segmentation and shape representation [16]. Object segmentation in images is classified into several segments and their shape features can be represented. Shape representation is classified into two types. First, region-based methods where all of the pixels in the object region are taken into account to acquire the shape representation [21]. The Grid based method [22], is one of popular region based approaches. The basic idea of this approach is to represent each shape in the image by a binary vector. For each shape the image pixels are scanned and will be assigned one as a value if it is a located in the shape and a value of zero otherwise. The similarities between two shapes are measurement by the binary 'Hamming distance' [23].

Secondly, boundary-based methods are these methods that use only the pixels that represent the object edges to describe the shape [21]. Canny edge [24], is one of the good performance boundaries based technique that is widely used to detect an image's objects. The Canny edge detector works as follows. 1) Canny is sensate to noisy, therefore image noise is eliminated by smoothing the image with Gaussian filter. 2) Gradient magnitudes and direction are computed using a sobel edge detector [25], at each pixel. The gradient magnitude at any pixel determines whether it lies on an edge or not. If the magnitude of a gradient is high, it implying an edge, otherwise it is not an edge. Thirdly, if the gradient is a maximum at any point, edges will occur at this point and suppress any point that is not at the maximum (non-maximum suppression). Finally, hysteresis the canny uses low and high thresholds, where pixel gradient is accepted as an edge, if it's value is higher than the upper threshold. In addition, if pixel gradient value is below the lower threshold, then it is rejected.

Furthermore, if the pixel gradient is between both thresholds, then it will be accepted only if it is connected to a pixel that is above the upper threshold.

In recent years, re-ranking algorithms, which are used to reorder original results of queries to show the most relevant files at the top of the ranked lists, has been widely used.

2.1.2 Multimedia Search Re-ranking

Most search engines like Bing, Yahoo, and Google, build multimedia databases based on text search approaches, i.e., using text such as surrounding text, user provided tags, description, and title of visual content. However, this type of search technique neglects the visual content, and its performance can be unsatisfying for retrieving multimedia files.

Visual re-ranking is a process used for re-arranging and improving performance in the initial search ranked list based on visual information of documents. This information also consists of multimodal cues that can be any auxiliary knowledge. It can be features extracted from each visual document. Mainly, visual search re-ranking has three challenging problems: First, unsatisfactory initial search performance, which means that the majority of initial query results are usually irrelevant documentation, and second, the lack of available knowledge or context for re-ranking. For instance, users do not agree to provide their profiles or visual query examples. Third, large-scale dataset, most of the techniques have been tested on a small dataset, however re-ranking approaches are frequently asked to deal with large scale datasets. In the following the four main types of multimedia content-based re-ranking systems [26], i.e., self-re-ranking, example based re-ranking, crowd re-ranking and interactive re-ranking are summarized.

a) Self-Re-ranking Methods

Self-re-ranking methods [26], assumes that the top *n* ranked documents are relevant files, and it assumes that last retrieved *n* documents are irrelevant. Then the query's initial results will be re-ranked based on their similarity to these two groups.

The first step in self-re-ranking is discovering and eliminating the noise in the initial query results, i.e., irrelevant files. Then, it tries to improve the order of initial query results based on their similarity to the two groups mentioned above. However, self-re-ranking methods are dependent on the files that appear at the top of the query initial results. If relevant documents on that list are few, this technique fails to improve the query results. To overcome this problem, some self-re-ranking methods [26], employ other modalities such as audio, linkage of web pages, and text that are also worth taking into consideration in order to improve the quality of the new re-ranked results.

b) Example Based Re-ranking Methods

Similar to QBE, this technique requires a few query examples along with the textual query to mine the relevant information. Two of the main example-based re-ranking techniques are concept based re-ranking, and linear multimodal fusion, and their details are summarized below.

1. Concept based re-ranking [26], integrates text description with low level feature description to eliminate the semantic gap between what information can be obtained from a low level feature and what users really want.
2. Linear Multimodal Fusion (LMF) [26], uses multiple single modality based techniques that work based on initial search results and each single modality returns a ranked list.

Figure 2.2 illustrates three kinds of search results from text, visual, and concept modalities and linear combination, which is used to fuse different ranked lists. The relevance scores for each file are combined and the files are re-ordered according to their combined scores. In addition, as shown in [26], Query Expansion (QE) is used to reformulate the user query for reducing the number of irrelevant documents retrieved by knowledge retrieval; and it is used to provide the user with more relevant images or documents. In multimedia, query expansion a given image or object query based on top *n* ranked documents searches in the database to retrieve a set of image regions that match with the query, then collecting those regions along with the main query to form a richer latent model of the object. After that, it queries the dataset using this expanded set of matching regions, finally repeating this operation as necessary.

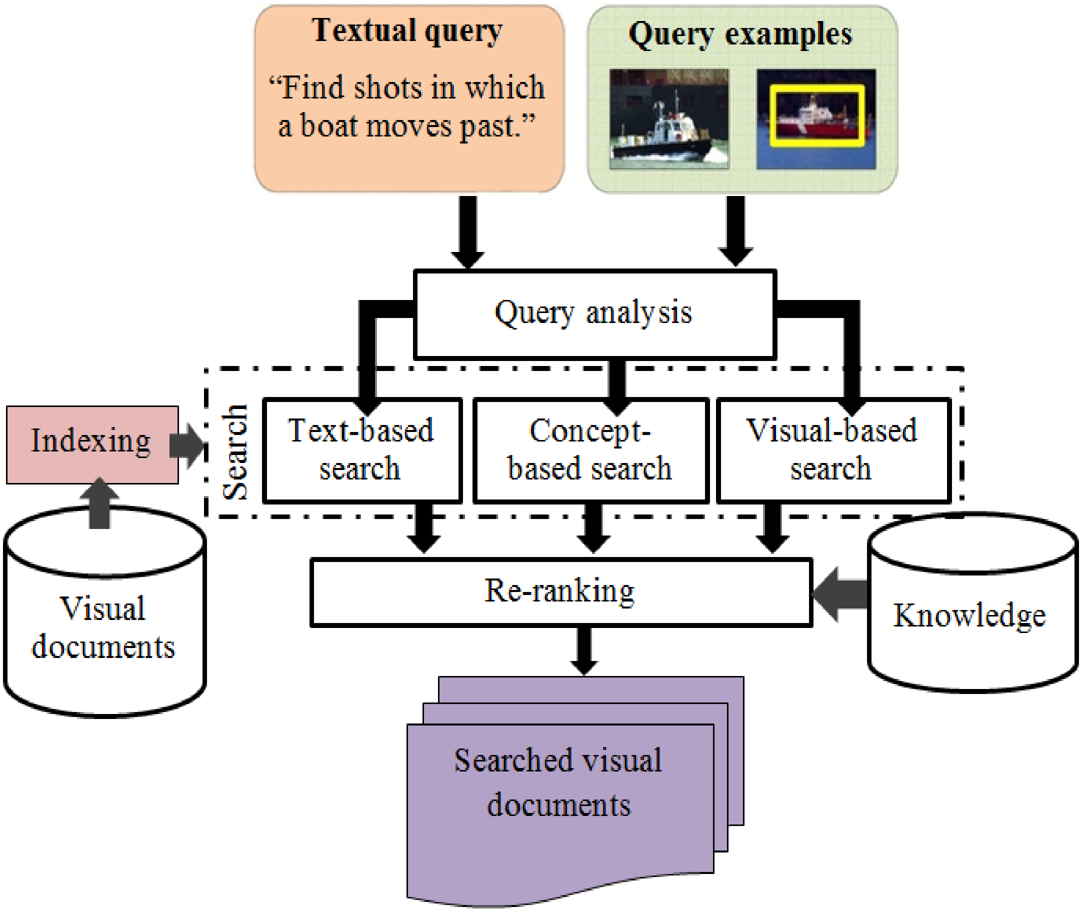

Figure 2.2 [26]: Illustrates a system that integrates three types of image search engines with re-ranking algorithms.

c) Crowd Re-ranking Methods

In general, crowd re-ranking combines results of multiple search engines in order to find and increase the number of relevant files. It assumes that the re-ranking process can be significantly improved due to the rich information involved. After collecting query results from multiple sources, crowd re-ranking tries to find representative visual patterns, as well as the relations between the collected results. Experiments showed that this method can improve the quality of retrieved results. On the other hand, processing time and the noisy nature of web information is a key issue in crowd re-ranking.

d) Interactive Re-ranking Methods

In this approach users are required to give feedback about a portion of the initial ranked list, whether they are relevant or irrelevant [26]. Then, the search initial results are re-ranked accordingly, to show the most relevant files at the top of the new results list. Unfortunately, this approach is not preferred for most of the users, as they do not like to spend more time on performing extra actions.

2.2 Literature Review

Nowadays, multimedia retrieval systems are one of the important research fields. Many researchers are trying to improve the efficiency of getting multimedia files through the Internet. In this section, we have summarized recent developments related to image retrieval.

2.2.1 Query-Specific Semantic Signature

The query specific semantic signature [27-29], is a unique image representative that will be produced from the image features. The main advantages of semantic signature are: 1) the signatures are shorter than the image visual features, and 2) all duplicated images have the same signatures. In addition, this approach can be applied to re-ranking images without picking query images. It assumes that query initial results have the dominant object and the images belonging to that object should have a higher score in the ranked list. Furthermore, the query specific semantic signatures also play a significant role in reducing the semantic gap [28] they also help in predicting the image's topics. This can be done based on the similarity of images signatures, which employs applying the following two steps:

First step, which is called the offline step, classes that include different concepts, related to the query keyword are automatically found, i.e., these classes are known as references. Some reference classes have similar semantic meanings and their training sets are visually similar. The redundant reference classes are removed to refine the efficiency of online image preprocessing. According to query keyword, the semantic signature is extracted for an image by calculating the similarities between the image and the reference classes of the query keyword.

Second step, which is called the online step, after a user is submitting a query, a pool of images that are associated with the query keyword and have a semantic signature in the same semantic space will be retrieved. In addition, whenever a user selects any of the query results, this approach has the ability to re-rank the query result based on its similarities to the selected image. As mentioned before, this approach is efficient because the semantic signature is shorter than the original image feature vector, however the size of the retrieval image pool in this technique predefined to a fixed size (e.g., including 1000 images).

2.2.2 Ranked List Similarity

In [30], similarities algorithms of query results work on the conjecture that contextual information has encoded the similarity to provide resources for refining the effectiveness of CBIR descriptors. An iterative approach is employed to refine the effectiveness of ranked lists. Generally, two images are considered for distance calculation, and the position of both images might be incorrect in the initial ranked lists. Additionally, many distances must be computed to bring the most relevant files to the top of list. The contextual information provided by the ranked lists can be used for improving the incorrect scores. Moreover, two images have the same ranked lists only if they are duplicate.

2.2.3 Large Scale Retrieval and Generation of Image Descriptions

In [31], the proposed approach utilizing query image to generate relevant description; such is acquired by computing the global similarity of a query image to a huge web collection of captioned images. Large scale database of pictures that are associated with descriptive text is one key requirement of this approach. In this approach, after the relevant description is generated, it finds the nearest matching images by calculating the global similarity of a query image to huge collection of captioned images. In addition, measuring visual similarity is done based on: first, gist feature, which is a global image descriptor related to perceptual dimensions of scenes such as ruggedness, roughness, naturalness etc., and the second descriptor which is also a global image descriptor, computed by changing size of the image into a “tiny image”, size of each thumbnail contains 32×32. The similarity of query image with images in the dataset computed by summation of gist similarity and tiny image color similarity. Note that the approach presented in [31] is not useful in the case of, first, having a small number of captioned images in the database, and second when the caption of images has no relation to the content of images itself.

2.2.4 Prototype Based Re-ranking

In [32], the approach consists of online and offline stages. The offline part is done by learning the re-ranking model [32], using user label training data, which is constructed from the text based search results. In the learning stage, a score vector is calculated for each image and it is a corresponding query. In the online stage, user should be submitting a text query into a search box engine. Then, the user will be able to obtain the initial ranked results. The approach of [32], obtains a score vector for each of the top-N images in the initial ranked list. This vector contains the score for all the meta-re-rankers. Note that meta-re-rankers are prepared by examining visual similarities for images that have been re-ranked previously.

2.2.5 Exploiting Click Constraints and Multi-view Features for Image Re-ranking

This method is useful to deploy the clicked data to efficiently justify the relevant images, and to adaptively associate different features and find the convenient ones. However, sometimes users click and view some images which are not relevant to the required files. This leads the re-ranked list to contain large number of irrelevant files. The second problem is related to low click counts of new uploaded images, which prevents new uploaded relevant images from being shown in the query results.

In [33], the multi-view hyper-graph based learning technique (MHL) adaptively associated click data and diverse visual features have been developed. There are three main steps in this approach. First step, achieving the query independent semantic representation, this process is done by classifying images as relevant and irrelevant to the query. This approach assumes that the queries have a strongly relevant relation with high click counts images and it assumes that the semantic similarities among these images are high. Second step, hyper-graph learning [33], is used to construct a group of manifolds for different visual features, where a set of vertices is connected by hyper-edge in a hyper-graph [33]. Third, the semantic of the manifold click data associated with multiple visual manifolds is calculated using the graph-based learning framework.

CHAPTER 3

CONTENT BASED APPROACHES

3.1 Overview

The concept behind CBIR is to retrieve images from databases that are relevant to a given query based on the content of the files themself. Hence, relevancy is equitably dependent on the content of the images. Initially, the features are extracted from images and their values are stored in a database. Then, the attributes of the images are compared with the attributes of a user's query according to a similarity measurement. Finally, the images are ranked, based on relevancy to the query, to show the most relevant files at the top of the query results.

3.2 Feature Extraction

In CBIR system, feature extraction is one of the most significant components that is used to detect a set of visual features to represent content of the local, global, and regional features. In this study, several CBIR algorithms have been compared and its details are summarized below:

3.2.1 Fuzzy Color and Texture Histogram (FCTH)

The FCTH descriptor, extracts multiple attributes which include color and texture that are combined in one histogram. The size of FCTH is limited to 72 bytes per image, hence, it reduces a high dimensional vector of images and is more suitable for large databases. In addition, this method is appropriate for accurately retrieving images that have noise or distortion and can handle and be used for rotating images [1].

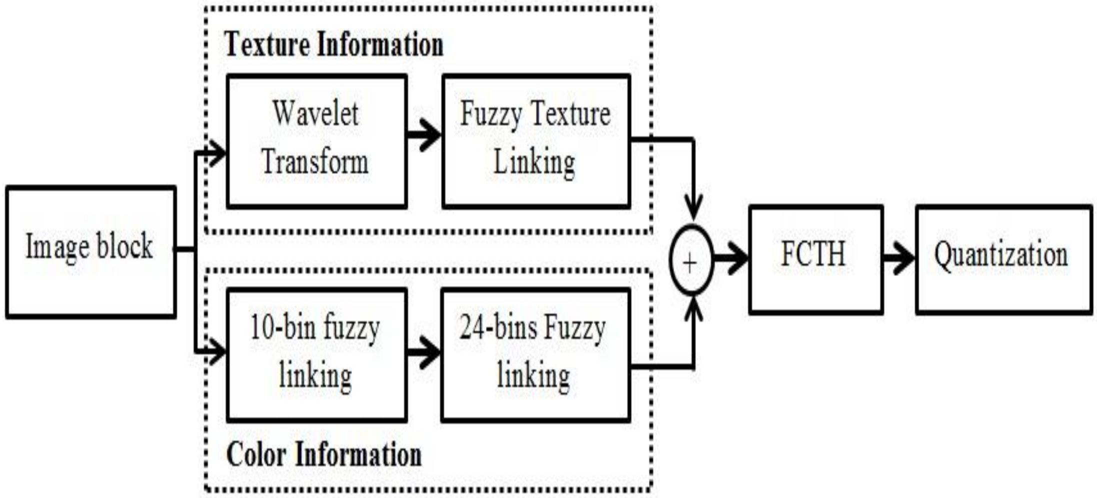

Figure 3.1 [1]: Block diagram of fuzzy color and texture histogram technique.

In FCTH, the image is firstly divided into a number of blocks and each block is consecutively passed into three fuzzy units to extract visual content. A block diagram of FCTH is shown in Figure 3.1. Mainly, FCTH works on extracting texture and color information. This process is done using the following units:

First, Fuzzy Color Segmentation: This step contain two units, first unit uses the three channels of HSV color space, i.e., Hue, Saturation, and Value of the image as an inputs, then 10 bins color histogram is produced as output, where first bin, i.e., bin number 0 represents black color, bin 1 represents gray, bin 2 represents white, bin 3 represents red, bin 4 represents orange, bin 5 represents yellow, bin 6 represents green, bin 7 represents cyan, bin 8 represents blue and bin 9 represents magenta. These colors are presented based on the fuzzy-linking histogram that is introduced in [1].

In the second unit, the resulting bins of the previous unit, in addition to the value of S and V of the HSV color space in each pixel, are used to achieve a 24 bins histogram. Each color bin in the previous unit classifies into one of three hue areas that are labeled dark color, color, and light color where color is obtained in the first 10 bins unit.

Second, Fuzzy Texture Segmentation: For extracting texture feature from the images, three features were used to represent energy in high frequency bands of wavelet transforms. The texture elements of each image block are achieved based on applying “Haar wavelet transform” on Y component, which represent the brightness in the YIQ color space. After applying 1-level Haar wavelet transform on a 4×4 block that is decomposed into 4 frequency bands. Each band includes 2×2 coefficients as shown in Figure 3.2. The motivation for using the features exported from high frequency band is that these features are reflected texture properties. In various frequency bands, moments of wavelet coefficients have proven to be effective for representing texture [34]. In different frequency bands, the coefficients are shown variation in different directions. For example, the **HL** shows activities in the horizontal direction. An image with vertical strips thus has high energy in the **HL** band and low energy in the **LH** band.

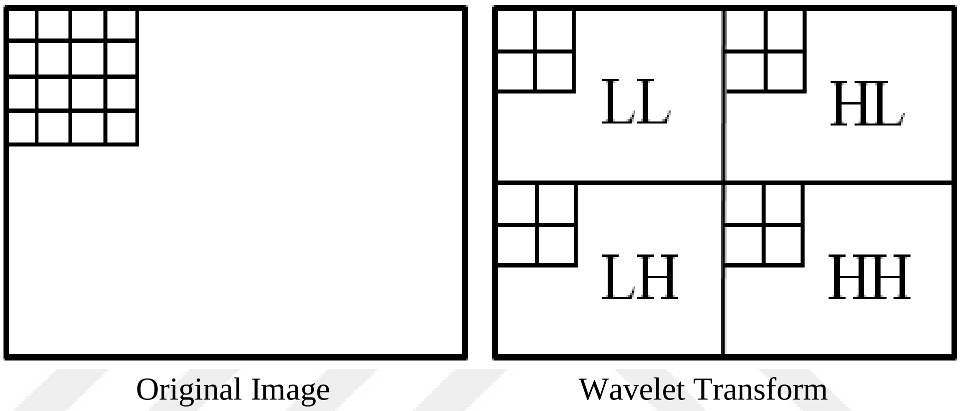

Figure 3.2: Decomposition of the blocks into frequency bands by wavelet transforms.

These elements $F_{LH}, F_{HL},$ and F_{HH} are used as input in the fuzzy units that form a histogram of eight bins regions as output. These regions are analyzed as follows: (0) Low Energy Linear area, (1) Low Energy Horizontal activation, (2) Low Energy Vertical activation, (3) Low Energy Horizontal and Vertical activation, (4) High Energy Linear area, (5) High Energy Horizontal activation, (6) High Energy Vertical activation, (7) High Energy Horizontal and Vertical activation.

3.2.2 Edge Histogram Descriptor (EHD)

Edges are considered as a significant feature representing the content of the images. In this descriptor, a histogram is used to represent edge features. The size of histogram is fixed, i.e., only 80 bins. This make it suitable to be used for huge databases and it is also more effective for retrieving images that have different sizes and have been rotated.

Mechanism of EHD

Initially, regardless the original size of the image, it will be classified into 4 × 4 equal size local areas that are called sub-images as shown in Figure 3.3. Next, for each sub-image, a histogram of edge distribution is generated. In this descriptor, as shown in Figure 3.4 [2], the edges are categorized as follows: vertical edge, horizontal edge, 45-dgree edge, 135-dgree edge, and non-directional edge.

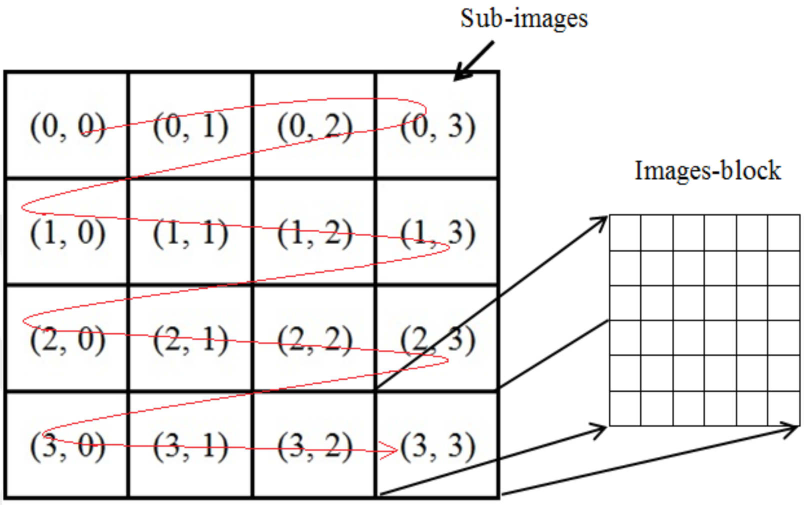

Figure 3.3 [35]: Defining sub-images and image-blocks using EHD.

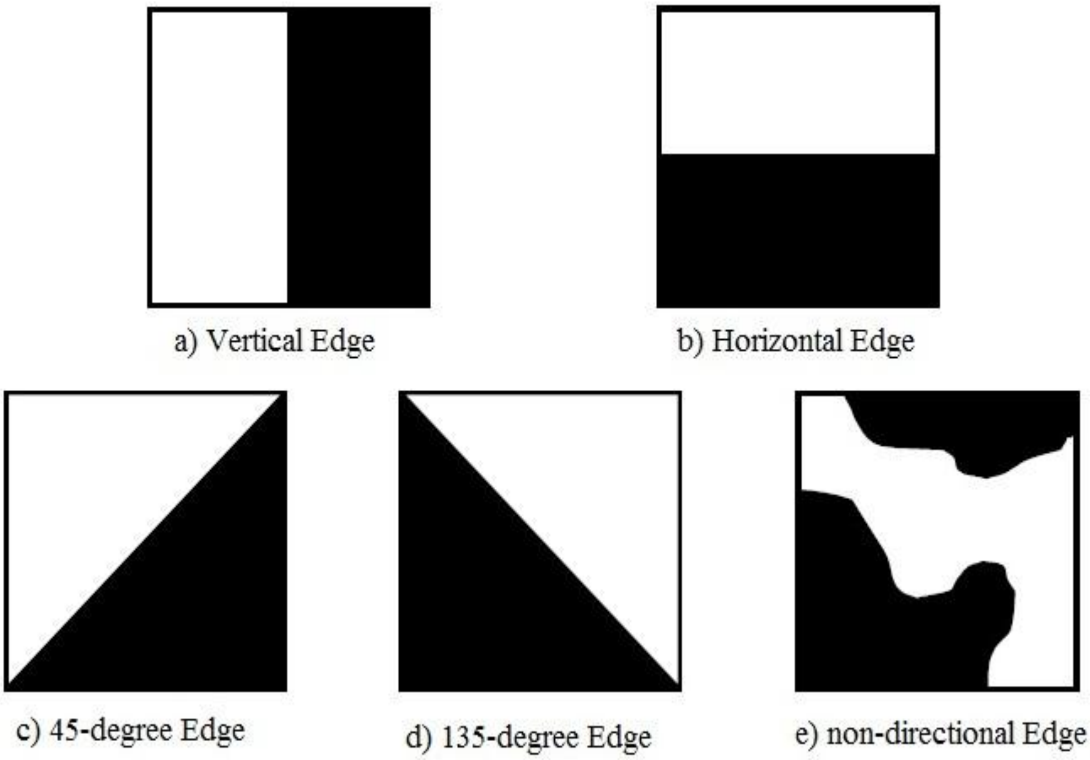

Figure 3.4 [35]: Five types of edge in EHD.

Note that each image is divided into 16 sub-images, and for each sub-image, five histogram bins are dedicated to represent the frequency of occurrences of the five edge types. Thus, $5 \times 16 = 80$ bins are produced and represent the edge histogram of each image. Table 3.1 [2], shows information of the 80 bin produced by EHD.

Table 3.1 [2]: Semantic of Local edge bins in EHD.

Histogram bins	Semantics
Bin#[0]	Vertical edge of first sub-image
Bin#[1]	Horizontal edge of first sub-image
Bin#[2]	45-degree edge of first sub-image
Bin#[3]	135-degree edge of first sub-image
Bin#[4]	Non-directional edge of first sub-image
Bin#[5]	Vertical edge of second sub-image
: :	: :
Bin#[75]	Vertical edge of last sub-image
Bin#[76]	Horizontal edge of last sub-image
Bin#[77]	45-degree edge of last sub-image
Bin#[78]	135-degree edge of last sub-image
Bin#[79]	Non-directional edge of last sub-image

3.2.3 Scalable Color Descriptor (SCD)

The Scalable Color Descriptor is derived from a color histogram defined in the Hue-Saturation-Value (HSV) color space with fixed color space quantization, i.e., 256 bins. In addition, it uses a Haar transform coefficient encoding, allowing scalable representation of description, as well as complexity scalability of feature extraction and matching procedures. Furthermore, Hue component is quantized to 16 bins, Saturation component is quantized to 4 bins, and Value component is quantized to 4 bins. Then, based on the desired accuracy, the binary representation is scaled to specific number of bins. Figure 3.5: Shows the semantic diagram of SCD.

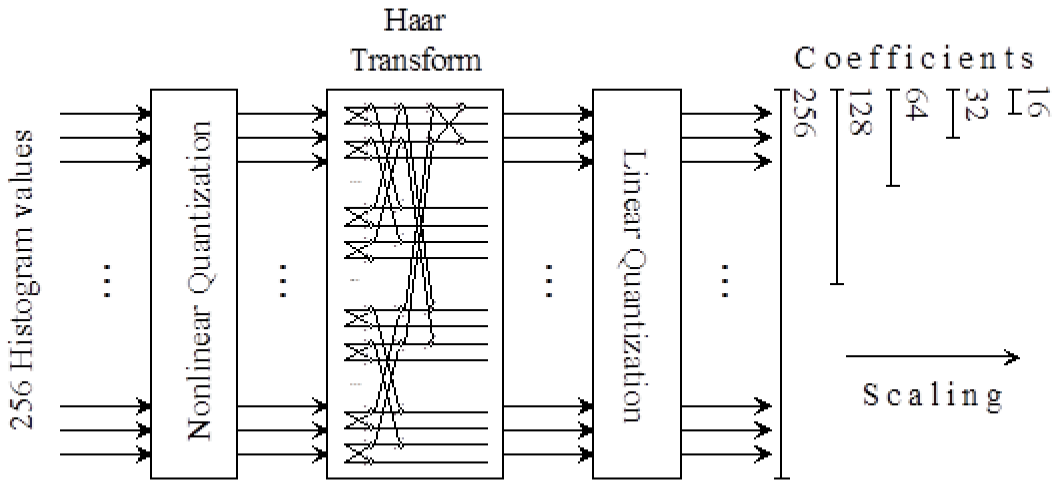

Figure 3.5 [3]: A schematic diagram of SCD generation.

3.2.4 Color Layout Descriptor (CLD)

This descriptor [36], represents the spatial distribution of color features is found in an image by dividing the image to sub-images and generating a thumbnail (64 blocks) for each part. In more details, the process of CLD extraction [4], consists of 4 steps that are illustrated in Figure 3.6.

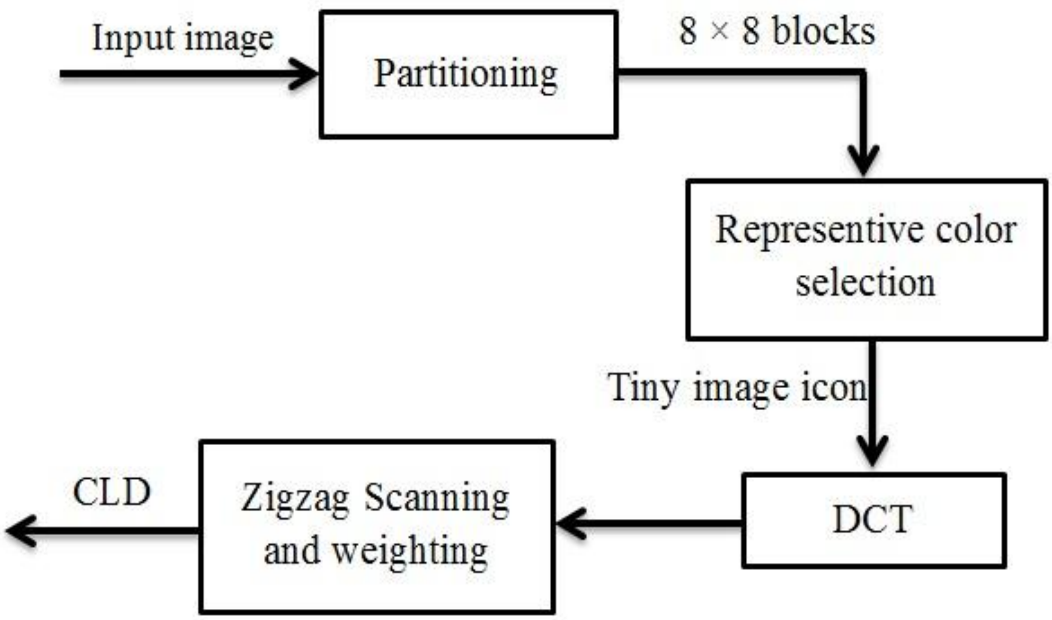

Figure 3.6 [4]: A block diagram of the CLD extraction.

In the first step, which is called image partitioning, the image is divided into a number of blocks, also known as tiny image icons, and each block contains 8×8 pixels. Note that if the size of the image is not divisible by eight, this method neglects and ignores the image's outermost pixels. In the second step, a single dominant color is selected as the representative color from each tiny image. Many techniques can be used for this purpose, however, as mentioned in [4, 36], the average of the values of all pixels is recommended and accurate. In the third step, each block is transformed to the Y/Cb/Cr color space, then each component of the color is transformed by a 8×8 Discrete Cosine Transform (DCT) which produces three set of 64 DCT coefficients for each component. Fourth step, the CLD is shaped by reading the DCT coefficients in zigzag scanned order [36]. Finally, the DCT coefficients of all components are arranged in a 64 one-dimensional vector.

3.2.5 Color and Edge Directivity Descriptor (CEDD)

The CEDD descriptor extracts multiple attributes that include color and texture which are combined in one histogram. The attribute CEDD results from the combination of three-fuzzy units. The size of CEDD is limited to 54-bytes per image, which reduces the high dimensional vector of images and is more suitable for usage in large image databases.

In the CEDD descriptor, the image is divided into a number of blocks and each block is consecutively passed into a three fuzzy unit to extract visual content. The color unit is the unit incorporated with the color extraction, and similarly the unit incorporated with the extraction of color information is called texture unit. The histogram of this descriptor consists from six areas that are determined by 24 individual areas, emanating from the color unit [5].

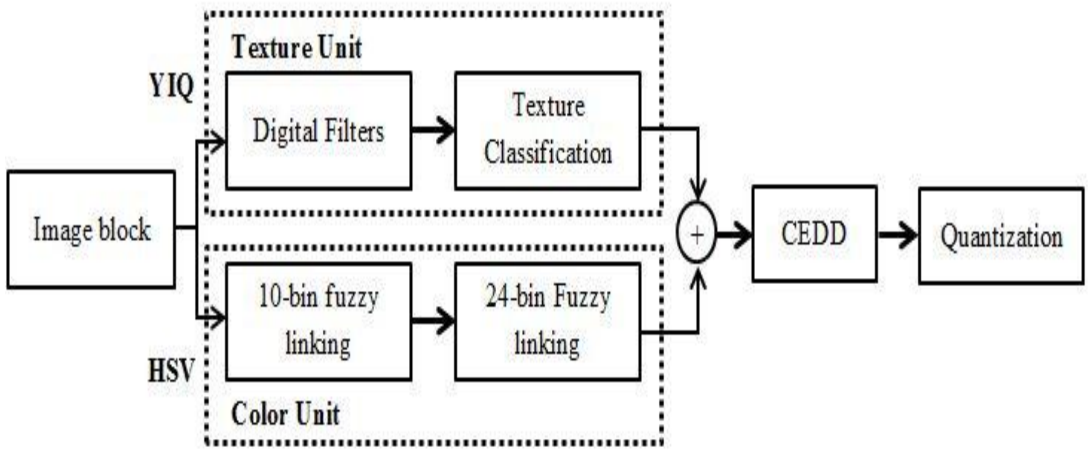

Figure 3.7 [34]: The block diagram of CEDD.

A block diagram of CEDD is shown in Figure 3.7. It is important to note that CEDD is similar to FCTH with a few minor differences in the way of extracting texture features. The process of extracting texture and color features is done using the following units:

First, Fuzzy Color Segmentation: This step is done exactly as described in FCTH method.

Second, Fuzzy Texture Segmentation: In this part, the 5-digital filters are used to export the texture features. These digital filters are able to characterize and detect the five type of edges described in section 3.2.2. In addition, texture information is extracted from each image's blocks, and each block includes 4 sub-blocks. Then, the edge magnitudes for each block are found, and all magnitudes are normalized. Finally, each of the image-block is classified into texture block or a non-texture block. This classification process is done as follows. First, find the largest value among the five magnitudes found in the previous step, and second, if the max value is greater than a predefined threshold, the image block is classified as a texture block, otherwise it will be classified as a non-texture block. Furthermore, all image-blocks are classified based on the 5 types of edges, where each block can be classified into one or more types, and each edge type has a threshold. Hence, all blocks that have magnitude value greater that the edge threshold will be classified into that type.

3.2.6 Speed Up Robust Features (SURF) Combined with Fast Library Approximate Nearest Neighbor (FLANN)

SURF [6], is a visual descriptor that uses local feature detection to extract interest points, which are points that can be used to identify objects in the images. In addition, these points are the same for the rotated and scaled copy of the original file, and can provide reliable matching between different viewpoints of the same images.

In general, the SURF descriptor is extracted by constructing a square region center aligned to a vertical and horizontal orientation. This region is classified into smaller 4×4 square sub-windows, for each sub-window, Haar wavelet responses are calculated. Then, the sum of the values of the responses are extracted for both vertical and horizontal orientation, furthermore, the sum of the absolute values of both responses are extracted. Thus, each sub-window has a four-dimensional descriptor vector. In addition, the mechanism of SURF is explained in detail below.

First, detecting the interest point: this approach focuses on blob-like structures [6] to detect the interest points in the image, which are allocated at junctions, speckles, and corners of objects and at locations where the determinant is maximal. The determinant of the Hessian matrix (DoH), which includes the various Gaussian filters at each point for scale selection, is convolved with the source image. Furthermore, to speed up the implementation of SURF, it use integral image, which is defined by the sum of all pixel values for all rectangular regions within the processed image.

Second, interest point description: The SURF describes the content of intensity distribution in the key-point neighborhood to provide a unique and robust description. The Haar wavelet is calculated for window regions within a circular area of the key-point's neighborhood, and the sum of both wavelet responses is computed. Next, the orientation of the processed window region is changed by 60°, and the Haar wavelet is computed again.

3.3 Similarity Measurement

In CBIR, after extracting the features of images, it is very important and required to use an appropriate method (metric) to compute the similarity between images in the database and the query image. This process is done to rank the images based on their similarity to query and shows the most relevant ones at the top of query results. In the following three of the similarity metrics are summarized.

3.3.1 Tanimoto Coefficient

Tanimoto coefficient [37], is used for calculating the distance (*D*) in CBIR between two image descriptors, and can be expressed as follows.

$$D = t(a,b) = \frac{a^T b}{a^T a + b^T b - a^T b} \tag{3.1}$$

Where a and b are two vectors of image descriptors, and $a^T \; and \; b^T$ are a transposition of the vectors. The range of Tanimoto coefficient is within the range {0, +1}, *D* is equal to zero, when the compared images are duplicated, otherwise the *D value* is rising whenever the similarity is decreasing.

3.3.2 K-Nearest Neighbors

The process of comparing the descriptor of a query image with the descriptors of all images stored in a database is time consuming. To solve this issue, clustering techniques, which are grouping the images, where images in the same group are more similar to each other than to those in other groups [38], can be used. For instance, Fast Library Approximate Nearest Neighbor (FLANN), which uses the nearest neighbor clustering technique, has been introduced to improve the performance of SURF in dealing with large datasets. In this mechanism, the SURF descriptors is calculated for all images and combined in a matrix. Then, a FLANN index is constructed by dividing the images into groups where each group contains the most similar files. Then, using the query file, the K-Nearest-Neighbors (KNN) finds the most similar group(s) and computes the distance between all of the images in the selected groups and the query descriptor, to find and show the user the most similar files.

3.3.3 Euclidean Distance

The Euclidean distance [39], is another distance metric that can be used for similarity measurement in multimedia retrieval. The similarity between two image vectors is calculated by computing the square root of the sum of squared absolute differences. It can be computed as follows:

$$D(x, y) = \sqrt{\sum_{i=0}^{n} (x_i - y_i)^2} \tag{3.2}$$

Where x, y are n-dimensional vectors of image descriptors.

3.4 Software Environment

In this study, C# language has been used to implement the six methods studied in this thesis, i.e. FCTH, EHD, SCD, CLD, CEDD, and SURF combined with FLANN. It is important to note that in the case of FLANN, which works based on SURF, the EmguCV has been integrated with C#. In addition, a web page which is used to show the query results was created using ASP.NET. Furthermore, Postgresql 9.1, which is one of the most advanced open source database systems, and can be easily integrated with C#, has been used to create the needed databases. Last but not least, the server used in this thesis has the following properties: Intel(R) Core(TM) i3, CPU M 350 at 2.27 GHz (4CPUs), 4 GB RAM, and Windows Seven 32 bit Operating System.

CHAPTER 4

PERFORMANCE STUDIES

4.1. Overview

In this study, the performance of the FCTH, EHD, SCD, CLD, CEDD and SURF combined with FLANN content-based methods have been investigated. To ensure the robustness of test results, multiple databases that contain images which belong to different group of subjects, different sizes, i.e., small, medium, and large, and belong to different extensions such as, '.jpg', '.jpeg', '.png', '.tif', and '.bmp' have been used in this study. These databases contain images that have been collected from different resources such as, Wang [40, 41], MIR Flickr [42], UCID [43], MSRCROID [44], etc. It is important to note that in case of FCTH, EHD, SCD, CLD, and CEDD the used databases were built using 10.000, 50.000, and 100.000 images. On the other side, due to the lack of super servers, i.e., high quality computers, SURF combined with FLANN was tested using different databases that contained 1000, 2500 and 5000 images respectively. In addition, for each database, 10 query images were chosen randomly and the performance of the above methods was computed. Furthermore, after executing every query, we have tried to find the number of relevant files using the first ten, first twenty, and first thirty results. This process is done by two humans through checking the query results, and then counting the number of relevant files. Hence, as shown in [45, 46, 47], it has been found through delivering a questioner to a group of search engine's users, that most of the users check only the first twenty results of the query [45, 46, 47], and few users may check the first thirty files.

4.2. Experiments on First Database:

In this section, the performance of FCTH, EHD, SCD, CLD, and CEDD were tested on the first database, which contains 10,000 images. Figure 4.1 show the 10 queries used that have been selected randomly, to compare the accuracy of the above mentioned methods, and the results for these methods is shown below:

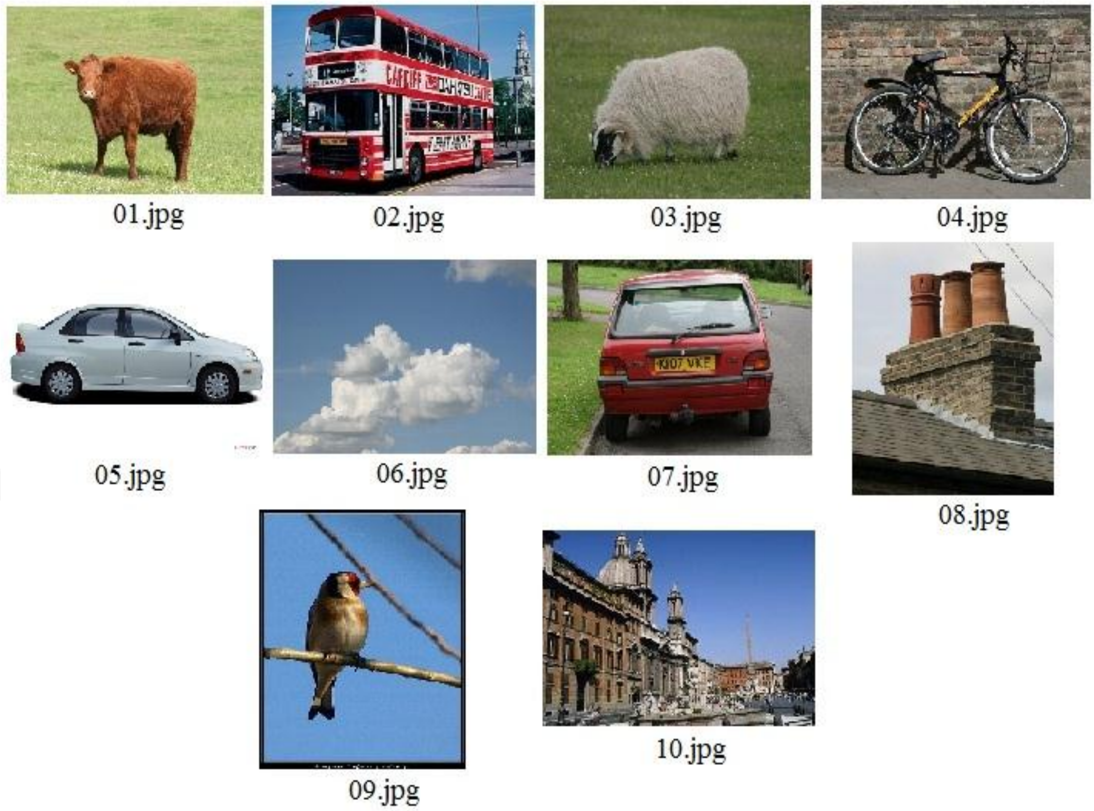

Figure 4.1: Image queries for the first database.

Experiment #1: Using FCTH Method

In this experiment the performance of the FCTH method was investigated, and the number of relevant files for each query was found using the first 10, 20 and 30 results. The results of this experiment are shown in Table 4.1. As shown in Table 4.1, the number of relevant files using the first 10 files was at least 80%, and for the sixth query which belongs to the 'cloud' category, the percentage of relevant files was 100 %.

Table 4.1: Number of relevant files using the FCTH method and first database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	01.jpg	10	19	24
Query #2	02.jpg	9	15	20
Query #3	03.jpg	9	19	26
Query #4	04.jpg	9	13	19
Query #5	05.jpg	8	13	19
Query #6	06.jpg	10	20	30
Query #7	07.jpg	10	16	22
Query #8	08.jpg	9	18	25
Query #9	09.jpg	8	18	26
Query #10	10.jpg	9	14	21

Experiment #2: EHD Method

In this experiment, the EHD descriptor was tested. Table 4.2 shows the number of relevant files for the 10 designated queries. It has been observed that the results of most of the query were below the user's expectations, for instance, in the case of the first query, only two files out of ten were relevant. This is because of the fact that this descriptor only describes and uses edge based on local edge distribution. On the other hand, the performance of this method can be accepted, if the query image has few objects and the borders of these objects are clear, i.e. represented by enough number of pixels, for instance as seen in the fourth query of the category 'bicycle', and fifth and seventh queries of the category 'car'.

Table 4.2: Number of relevant files using the EHD method and first database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	01.jpg	2	3	4
Query #2	02.jpg	8	12	19
Query #3	03.jpg	7	11	15
Query #4	04.jpg	10	17	23
Query #5	05.jpg	9	17	21
Query #6	06.jpg	4	5	7
Query #7	07.jpg	10	19	29
Query #8	08.jpg	4	6	6
Query #9	09.jpg	6	11	12
Query #10	10.jpg	2	4	6

Experiment #3: SCD Method

As shown in [3], SCD approach is one of most appropriated techniques for retrieving images. In this experiment, the performance of the SCD method was investigated, and the number of relevant files for each query was found using the first 10, 20 and 30 results. As shown in Table 4.3, the number of relevant files using the first 10 files was at least 90% and 85% for the first twenty files. In addition, for some queries such as the third and six queries, the percentage of relevant files was 100%.

Table 4.3: Number of relevant files using the SCD method and first database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	01.jpg	10	19	27
Query #2	02.jpg	10	20	29
Query #3	03.jpg	10	20	30
Query #4	04.jpg	9	17	24
Query #5	05.jpg	10	19	28
Query #6	06.jpg	10	20	30
Query #7	07.jpg	10	18	25
Query #8	08.jpg	10	18	21
Query #9	09.jpg	10	17	24
Query #10	10.jpg	10	20	24

Experiment #4: CLD Method

In general, CLD represents any image using the distribution of colors in that image. This means that this method can show two images that contain totally different objects as similar, if its objects have the same colors. In this experiment, the CLD descriptor was tested, and number of relevant files for the 10 used queries is shown in Table 4.4. Overall, as explained before, the number of irrelevant images can be high when objects have the same colors. On the other hand, whenever objects are represented by different colors, such as the first and ninth query, the results of this method can be acceptable.

Table 4.4: Number of relevant files using the CLD method and first database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	01.jpg	10	16	20
Query #2	02.jpg	5	5	9
Query #3	03.jpg	10	18	28
Query #4	04.jpg	8	13	16
Query #5	05.jpg	9	17	22
Query #6	06.jpg	9	18	28
Query #7	07.jpg	9	16	21
Query #8	08.jpg	7	8	8
Query #9	09.jpg	8	14	20
Query #10	10.jpg	4	5	5

Experiment #5: CEDD Method

In general, the CEDD descriptor, extracts multiple attributes that include color and texture which are combined in one histogram. In this experiment, the CEDD descriptor was tested. Table 4.5 shows the number of relevant files for the 10 used queries. As a result, the number of relevant files overall is good. In addition, when checking the first 10 files this method was able to achieve 100% relevant files for most of the queries.

Table 4.5: Number of relevant files using the CEDD method and first database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	01.jpg	10	20	28
Query #2	02.jpg	10	15	23
Query #3	03.jpg	10	19	26
Query #4	04.jpg	8	12	16
Query #5	05.jpg	8	12	16
Query #6	06.jpg	10	20	28
Query #7	07.jpg	10	20	28
Query #8	08.jpg	10	15	18
Query #9	09.jpg	8	16	24
Query #10	10.jpg	9	16	20

Figures 4.2, 4.3 and 4.4 show the number of relevant files using the compared descriptors i.e. the FCTH, EHD, SCD, CLD, and CEDD and the first database (10,000 Images) using the first ten, first twenty, and first thirty results, respectively.

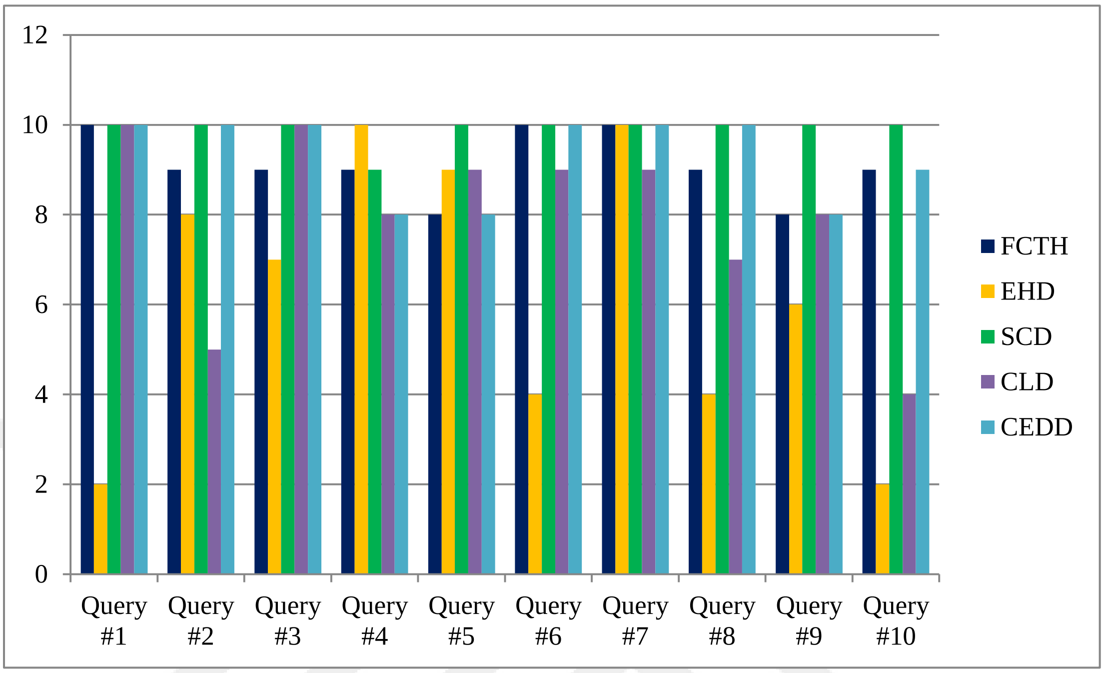

Figure 4.2: Number of relevant images for the first database and the compared descriptors using the first 10 results.

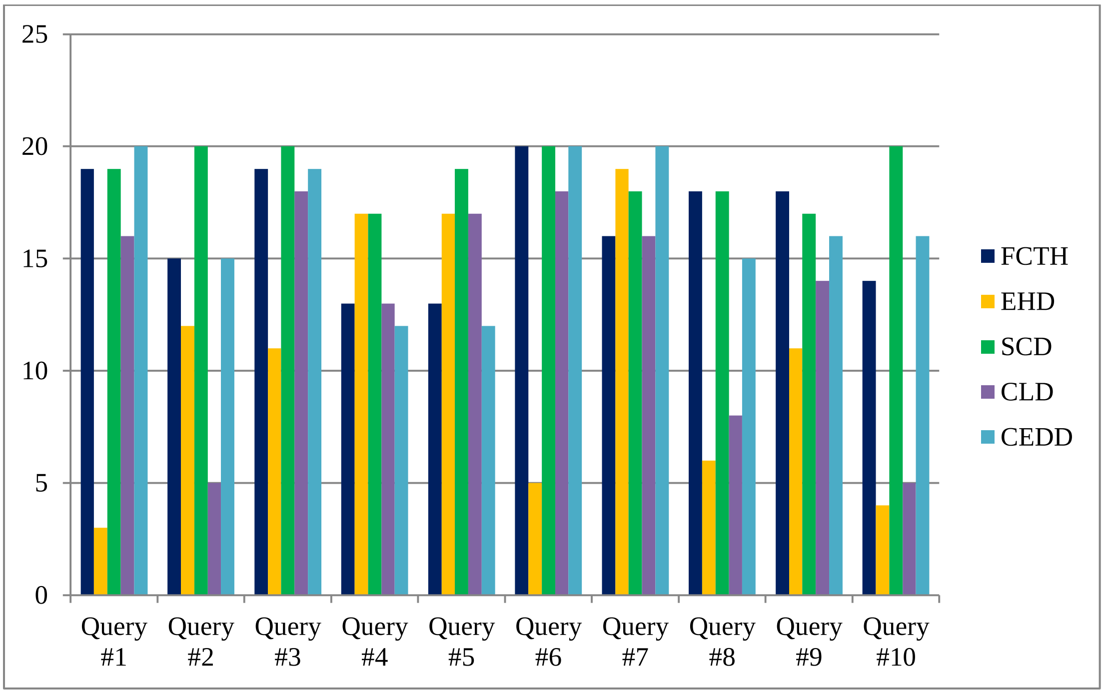

Figure 4.3: Number of relevant images for the first database and the compared descriptors using the first 20 results.

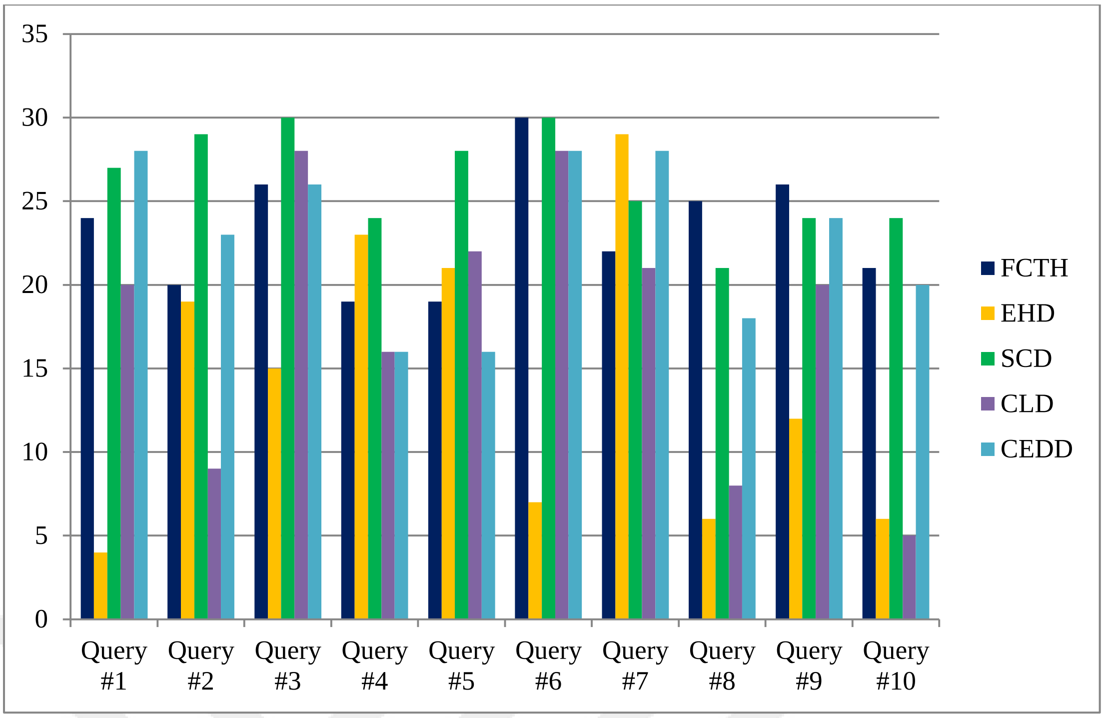

Figure 4.4: Number of relevant images for the first database and the compared descriptors using the first 30 results.

4.3. Experiments on Second Database:

In this section, the performance of FCTH, EHD, SCD, CLD, and CEDD were tested on the second database, which contains 50,000 images. Figure 4.5 show the 10 queries used that have been selected randomly, to compare the accuracy of the above mentioned methods, and the results for these methods is shown below:

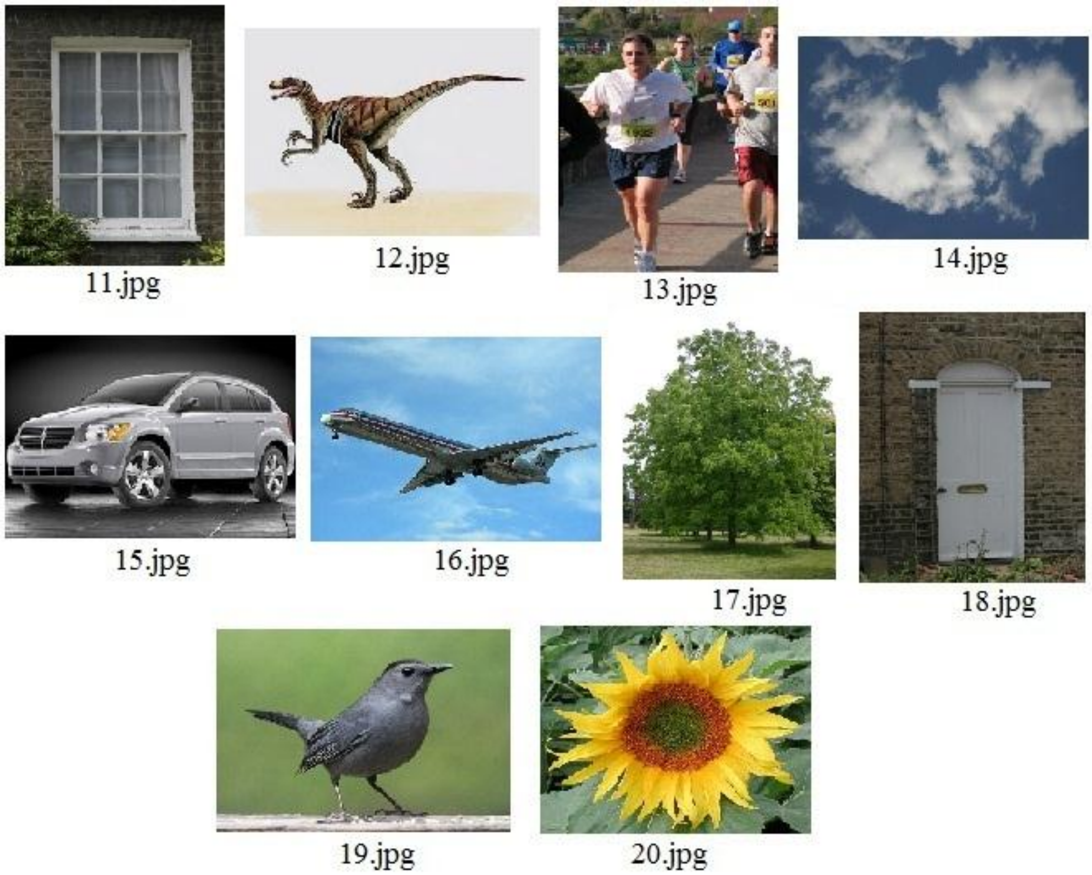

Figure 4.5: Image queries for the second database.

Experiment #1: Using FCTH Method

In this experiment, the number of relevant images using the FCTH descriptor was found using first ten, twenty, and thirty results. The results of this experiment are shown in Table 4.6. It is clear from Table 4.6 that by increasing the numbers of images in the database, the numbers of irrelevant images are increased for some of the queries. Hence, the performance of FCTH has been decreased.

Table 4.6: Number of relevant files using the FCTH method and second database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	11.jpg	7	14	21
Query #2	12.jpg	10	20	29
Query #3	13.jpg	7	13	16
Query #4	14.jpg	10	20	30
Query #5	15.jpg	8	14	21
Query #6	16.jpg	4	7	11
Query #7	17.jpg	9	14	19
Query #8	18.jpg	3	6	10
Query #9	19.jpg	10	18	27
Query #10	20.jpg	10	17	26

Experiment #2: Using EHD Method

In this experiment the performance of the EHD approach was investigated. The numbers of relevant files are shown in Table 4.7 for the 10 queries. Similar to the first database, the performance of this approach can be accepted, if borders of the object of the query image were regular, and/or, if the query image has few objects and the borders of these objects were clear, i.e. represented by enough number of pixels, for instance, fifth query of category ‘car’. However, the overall performance of this method was below expectations, for instance, the fourth, eighth, and ninth queries produced zero relevant files. This is because of and as mentioned before, the EHD method divides the image into 16 sub-images and these sub-images are further divided into a number of image-blocks that are largely affected on the content of the image, as they contain less information about the object(s). In addition, based on predefined thresholds, each image-block is represented by only one type of edge, however most of image’s blocks contain more than one type of edges.

Table 4.7: Number of relevant files using the EHD method and second database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	11.jpg	2	2	3
Query #2	12.jpg	7	14	16
Query #3	13.jpg	1	1	1
Query #4	14.jpg	0	0	0
Query #5	15.jpg	9	16	21
Query #6	16.jpg	2	2	4
Query #7	17.jpg	6	13	20
Query #8	18.jpg	0	0	0
Query #9	19.jpg	0	0	0
Query #10	20.jpg	6	6	10

Experiment #3: Using SCD Method

In this experiment, the performance of the SCD technique was tested on the second database, and the results of this experiment are shown in Table 4.8. Then number of the relevant files for each query was found using 10, 20, and 30 results. As a result, the overall performance was good, and this method was able to get 100% relevant files using the first 10 files for some queries, such as the second query which belongs to the ‘dinosaur’ category, the third query of ‘sport’ category, fourth query of ‘cloud’ category, fifth query of ‘car’ category, and seventh query of ‘tree’ category.

Table 4.8: Number of relevant files using the SCD method and second database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	11.jpg	9	18	25
Query #2	12.jpg	10	17	25
Query #3	13.jpg	10	16	24
Query #4	14.jpg	10	20	29
Query #5	15.jpg	10	18	27
Query #6	16.jpg	5	10	11
Query #7	17.jpg	10	17	27
Query #8	18.jpg	7	7	7
Query #9	19.jpg	6	12	16
Query #10	20.jpg	9	16	25

Experiment #4: Using CLD Method

In this experiment, the CLD descriptor was tested using the second database. Table 4.9 shows the number of relevant files for the 10 queries. The overall performance of CLD has been improved by using a larger database as compared with the results of first database; for instance, the percentage of relevant files was 100%, for the first query category ‘window’. In addition, the number of relevant images is very high for some queries such as the seventh, ninth, and tenth queries and the results for some other queries are also good such as the second, fourth, and fifth queries. However, this method might inadvertently retrieve some images that have the same colors of the query, even if these images actually have totally different objects. This will increase the number of irrelevant retrieved images, as shown in the results of the third, sixth, and eighth queries.

Table 4.9: Number of relevant files using the CLD method and second database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	11.jpg	10	20	30
Query #2	12.jpg	9	18	20
Query #3	13.jpg	1	1	1
Query #4	14.jpg	8	15	22
Query #5	15.jpg	8	16	21
Query #6	16.jpg	4	6	9
Query #7	17.jpg	9	18	28
Query #8	18.jpg	5	9	13
Query #9	19.jpg	9	18	27
Query #10	20.jpg	10	20	29

Experiment #5: Using CEDD Method

Table 4.10 shows the number of relevant files for the 10 queries. As shown in Table 4.10 the percentage of relevant files was 100% for the ninth query of the category 'bird'. Additionally, most of the results are acceptable. However, this descriptor cannot differentiate between two objects that have the same color and bounders, for instance the number of relevant files for the sixth query of category 'airplane' is low, and most the of retrieved images were from the category 'birds'.

Table 4.10: Number of relevant files using the CEDD method and second database.

Query #	Query Image	Number of Relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	11.jpg	9	18	26
Query #2	12.jpg	8	18	26
Query #3	13.jpg	6	9	10
Query #4	14.jpg	9	16	22
Query #5	15.jpg	10	20	28
Query #6	16.jpg	4	6	9
Query #7	17.jpg	7	13	22
Query #8	18.jpg	8	9	10
Query #9	19.jpg	10	20	30
Query #10	20.jpg	10	16	25

Figures 4.6, 4.7 and 4.8 show the number of relevant files using the compared descriptors i.e. the FCTH, EHD, SCD, CLD, and CEDD and the second database (50,000 Images) using the first ten, first twenty, and first thirty results, respectively.

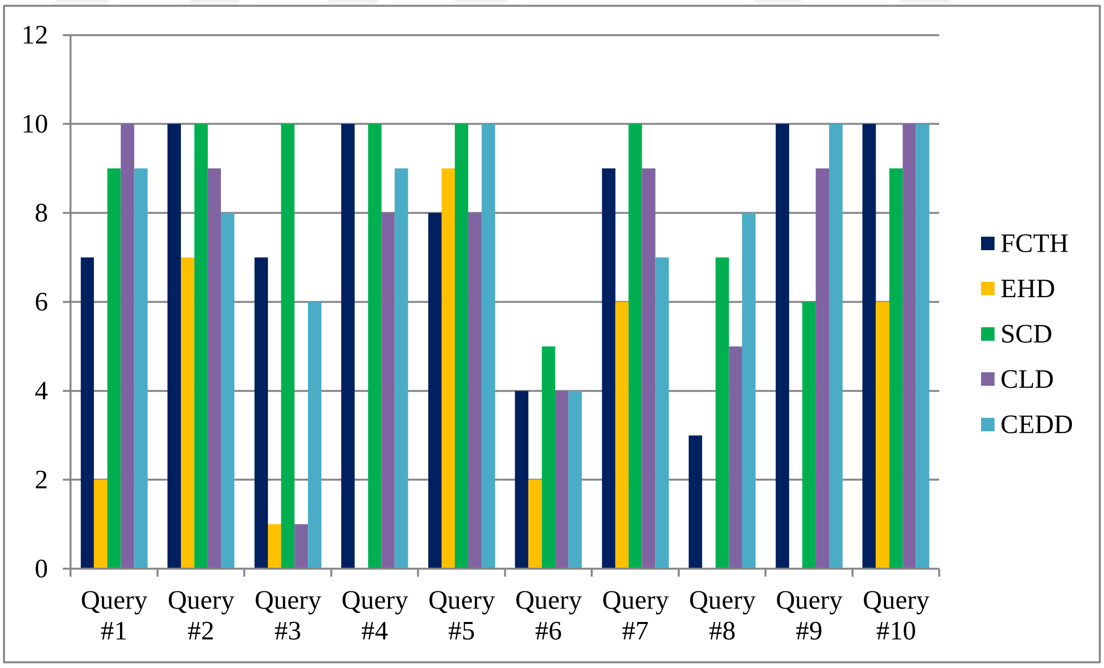

Figure 4.6: Number of relevant images for the second database and the compared descriptors using the first 10 results.

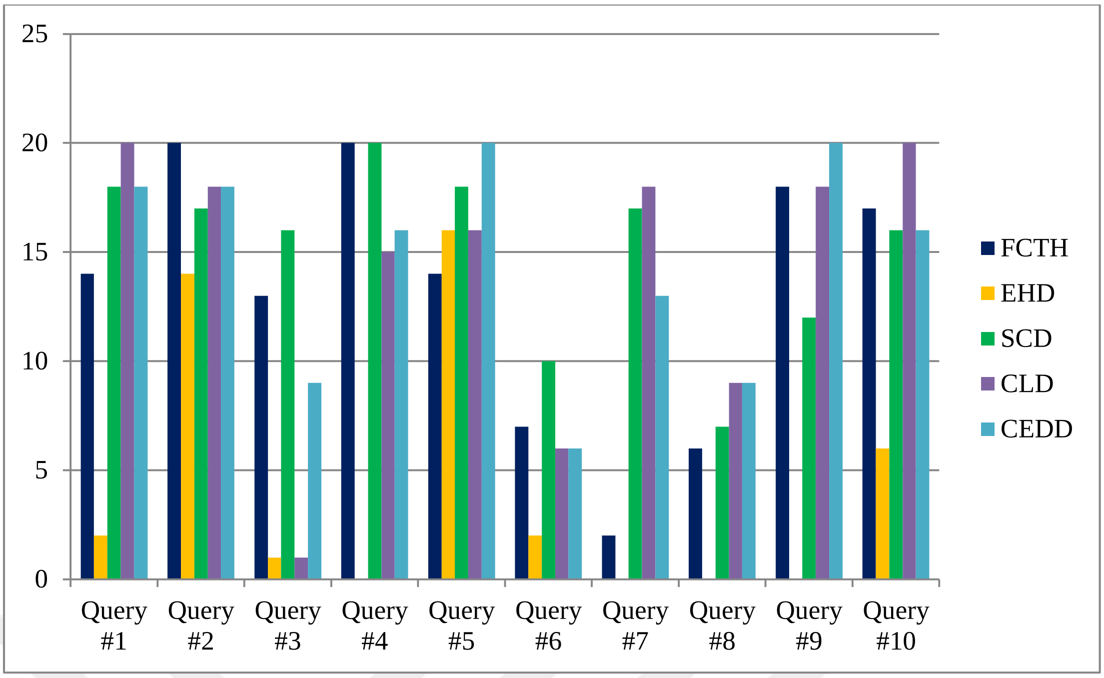

Figure 4.7: Number of relevant images for the second database and the compared descriptors using the first 20 results.

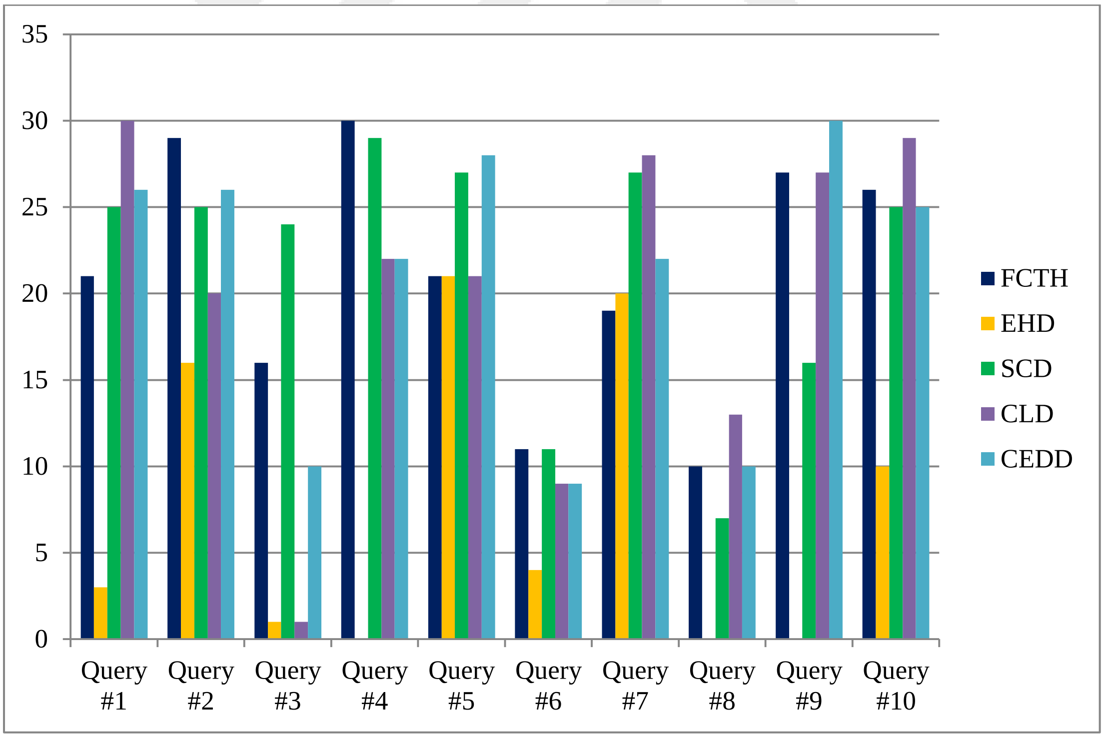

Figure 4.8: Number of relevant images for the second database and the compared descriptors using the first 30 results.

4.4. Experiments on Third Database:

To ensure the robustness of the test results, in this section FCTH, EHD, SCD, CLD, and CEDD were tested further using a database that contained 100,000 images. Figure 4.9 show the 10 queries that have been randomly selected. The results for these methods are described in the following:

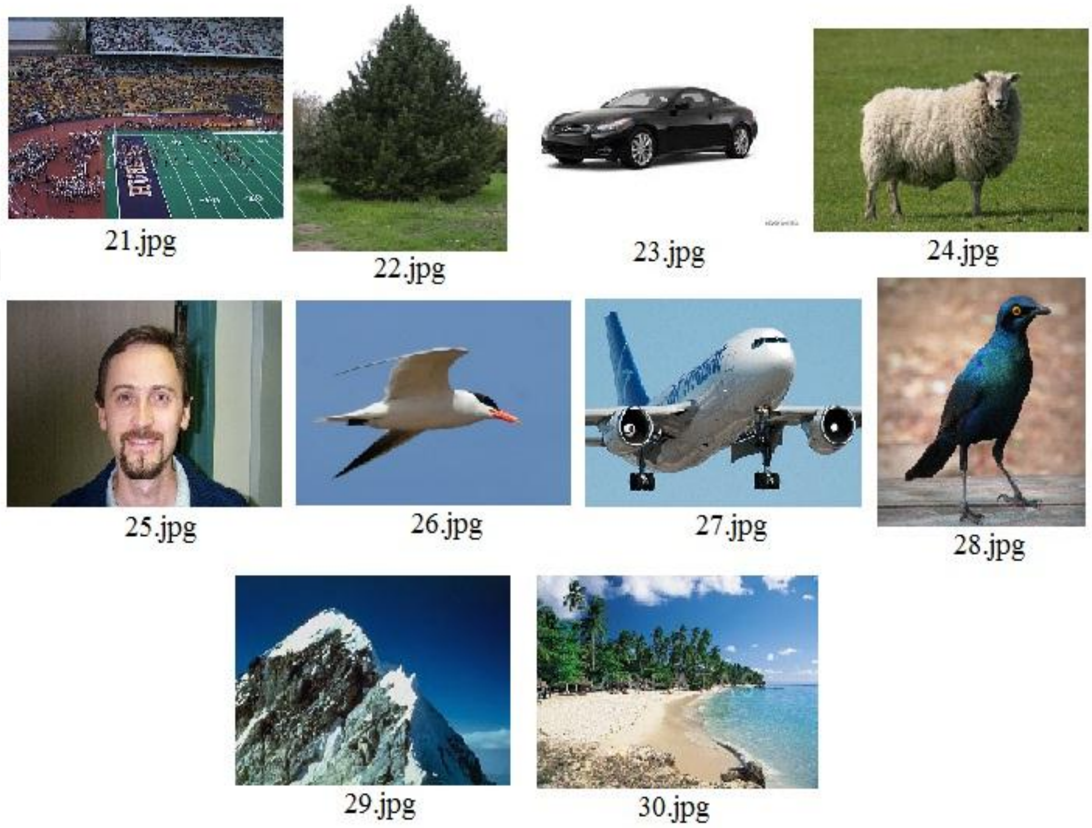

Figure 4.9: Image queries for the third database.

Experiment #1: Using FCTH Method

In this experiment the performance of FCTH method was investigated, and the number of relevant files for each query was found using first 10, 20 and 30 results. The results of this experiment are shown in Table 4.11. Although, the percentage of relevant files was 100% for the first query of the category ‘stadium’, the number of irrelevant images was the highest for the seventh query of the category ‘airplane’, and most of the retrieved images belonged to ‘birds’, ‘sky’, ‘balloons’, and ‘sea’ categories. However, they have in common with the query image the same background color. In addition, most of the retrieved images for the ninth query ‘mountain’ category are not relevant, as most of the images in this category were taken in different seasons. For example some images of snow-capped mountains were taken in the winter and some others were taken in the summer. Furthermore, some retrieval images contain buildings that look somehow similar to mountains.

Table 4.11: Number of relevant files using the FCTH method and third database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	21.jpg	10	20	30
Query #2	22.jpg	3	5	9
Query #3	23.jpg	10	19	26
Query #4	24.jpg	8	14	19
Query #5	25.jpg	3	7	13
Query #6	26.jpg	10	15	22
Query #7	27.jpg	2	2	3
Query #8	28.jpg	4	6	7
Query #9	29.jpg	3	3	3
Query #10	30.jpg	6	8	10

Experiment #2: Using EHD Method

Successful retrieval systems need to extract the right features that represent the content of images and are sufficient in describing objects. It is very difficult to achieve the most relevant files using only a single feature type. Therefore, as this descriptor only describes and use edge based on local edge distribution, the results of most of the query were below user’s expectations as shown in Table 4.12. In addition, it has been observed that this descriptor cannot distinguish between queries that have some common properties, such as color and boundaries, as shown in the results of the sixth the seventh queries. As mentioned before, the number of relevant files can be high, if the query image has few objects and the borders of these objects are clear, i.e. represented in enough number of pixels, such as third query, which of category ‘car’.

Table 4.12: Number of relevant files using the EHD method and third database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	21.jpg	2	2	2
Query #2	22.jpg	3	7	10
Query #3	23.jpg	10	19	29
Query #4	24.jpg	1	1	1
Query #5	25.jpg	7	12	16
Query #6	26.jpg	1	1	1
Query #7	27.jpg	1	1	1
Query #8	28.jpg	4	11	14
Query #9	29.jpg	0	0	0
Query #10	30.jpg	0	1	2

Experiment #3: Using SCD Method

In this experiment the performance of the SCD descriptor was investigated, and once again the number of relevant files for each query was found using the first 10, 20 and 30 results. Table 4.13 shows the number of relevant files for the 10 queries. In general, it has been noticed that this descriptor is not accurate for huge databases, and the results for some of the queries are irrelevant. Hence, using the color feature only might be not sufficient to accurately describe the image's objects, for example most of the images retrieved for the seventh query contained different objects that are colored in blue. Furthermore, it is difficult to retrieve similar images for the eighth query of the 'bird' category based on the color feature only, as most of the query similar images were taken in different places that affect the content of the images. However, the percentages of the relevant images were 100%, for the first query of the category 'stadium' and the fourth query of the category 'sheep'. This is because of the fact that the images in each of these categories almost all have the same objects and colors. Hence, the color information can provide sufficient information for these particular queries. In addition, this method uses the HSV color space, which is developed to approximate the same process through which humans perceive and manipulate color.

Table 4.13: Number of relevant files using the SCD method and third database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	21.jpg	10	20	30
Query #2	22.jpg	10	17	25
Query #3	23.jpg	10	20	29
Query #4	24.jpg	10	20	30
Query #5	25.jpg	2	4	7
Query #6	26.jpg	8	16	23
Query #7	27.jpg	2	4	5
Query #8	28.jpg	3	5	5
Query #9	29.jpg	2	3	3
Query #10	30.jpg	9	16	21

Experiment #4: Using CLD Method

In this experiment the performance of the CLD method was investigated, and the number of relevant files for each query was found using the first 10, 20 and 30 results. The results of this experiment are shown in Table 4.14. As expected, since this method focuses only on color features and it cannot provide the texture information of the objects in the images, the results for some queries are less than 50%. However, for certain queries, color features can provide enough information, for example, the first and third queries.

Table 4.14: Number of relevant files using the CLD method and third database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	21.jpg	10	20	24
Query #2	22.jpg	8	15	21
Query #3	23.jpg	10	19	29
Query #4	24.jpg	8	15	20
Query #5	25.jpg	4	5	5
Query #6	26.jpg	8	17	24
Query #7	27.jpg	1	5	9
Query #8	28.jpg	9	16	24
Query #9	29.jpg	2	4	4
Query #10	30.jpg	5	9	15

Experiment #5: Using CEDD Method

Table 4.15 shows the number of relevant images for ten queries using the CEDD descriptor. The results for some queries are highly relevant such as the first, third, fourth, and sixth queries; additionally the results for the second and fifth queries are quite good. On the other hand, the performance of the texture part of this descriptor decreases when it is required to deal with objects that have unsymmetrical boundaries such as the ninth and tenth queries. Moreover, the number of irrelevant images is high for the seventh query of the category ‘airplane’, as most of the retrieved files were images of the ‘birds’ category that have edge and color similarity to the query image.

Table 4.15: Number of relevant files using the CEDD method and third database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	21.jpg	10	20	24
Query #2	22.jpg	9	13	16
Query #3	23.jpg	8	18	28
Query #4	24.jpg	10	19	25
Query #5	25.jpg	7	14	16
Query #6	26.jpg	9	17	26
Query #7	27.jpg	2	2	2
Query #8	28.jpg	3	6	9
Query #9	29.jpg	2	5	7
Query #10	30.jpg	3	6	9

Figures 4.10, 4.11 and 4.12 show the number of relevant files using the compared descriptors, i.e. the FCTH, EHD, SCD, CLD, and CEDD and the third database (100,000 Images) using the first ten, first twenty, and first thirty results, respectively.

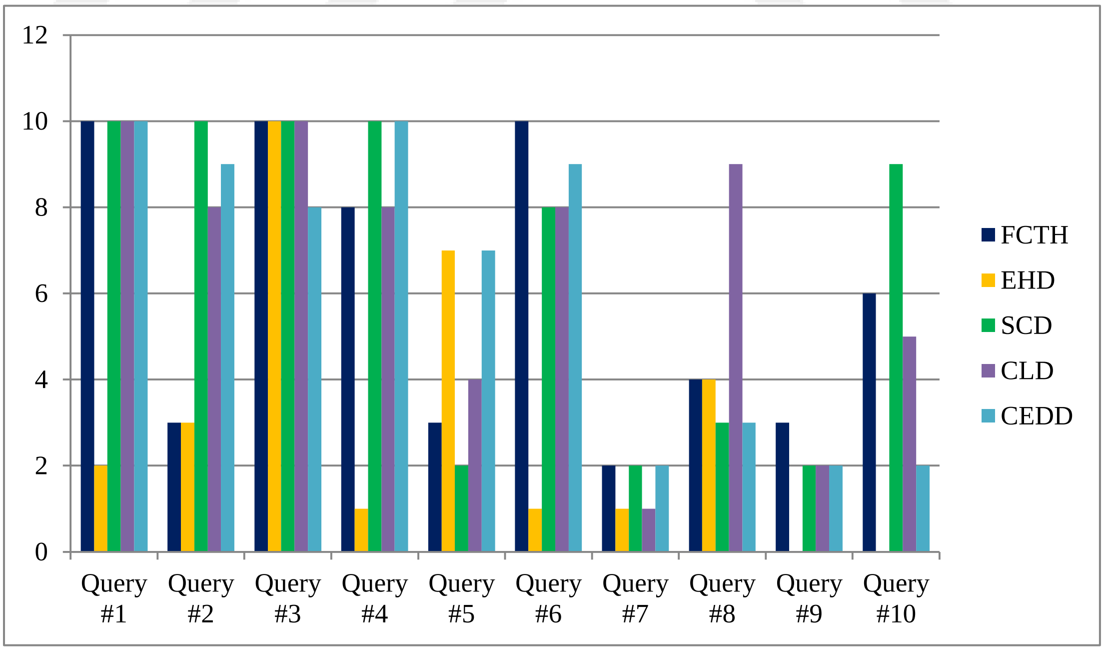

Figure 4.10: Number of relevant images for the third database and the compared descriptors using the first 10 results.

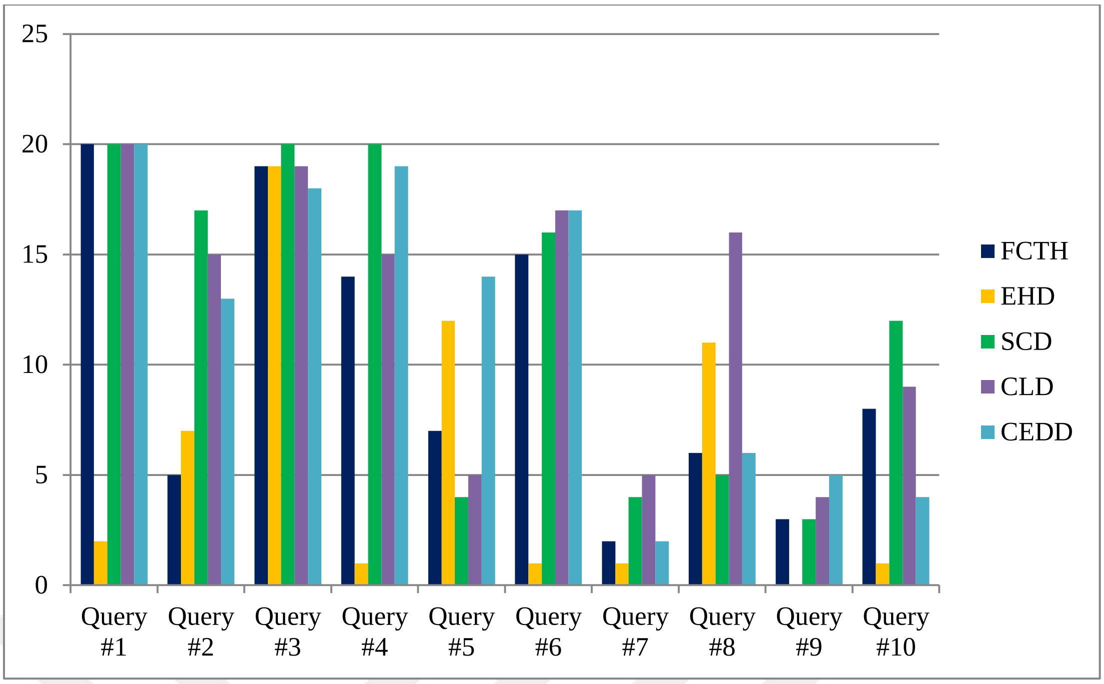

Figure 4.11: Number of relevant images for the third database and the compared descriptors using the first 20 results.

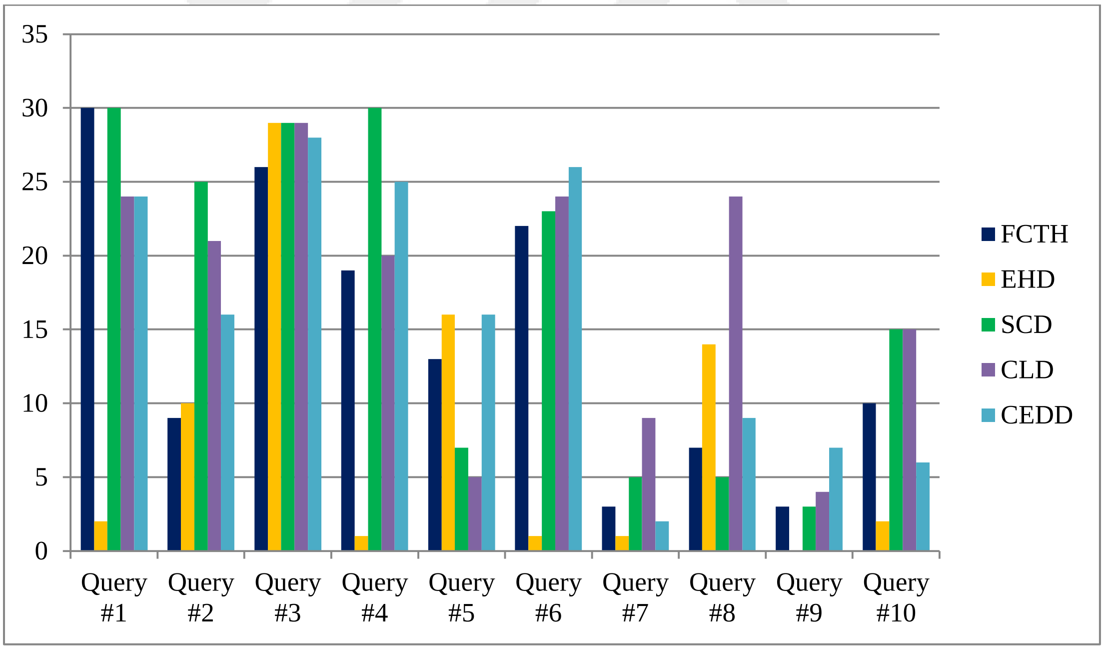

Figure 4.12: Number of relevant images for the third database and the compared descriptors using the first 30 results.

4.5. Performance of SURF combing with FLANN

In this section, the performance of SURF combed with FLANN method was investigated using three databases. The databases used were built using 1000, 2500, and 5000 images. In addition for each database, 10 query images were chosen randomly and the performance of this method is shown below.

Experiment #1: Performance using first database (1000 images)

In this experiment, the performance of SURF combined with FLANN was investigated, and the number of the relevant files was found using the first ten, twenty, and thirty results for each query. The query images for this database are shown in Figure 4.13, and the results of this experiment are shown in Table 4.16. It is clear that the number of the relevant images for the fifth query of the category 'tree' and the sixth query of the category 'window' are acceptable. However, the number of irrelevant images for some queries is high, as this approach neglects using the color, texture, and shape features individually, and instead it uses the interest points that are located at junctions or corners of the objects, however these interest points are often not sufficient to find the similarities among the objects within the images that can be represented by multiple models and/or shapes. For instance, the number of relevant files for the eighth query of the category 'motorcycle' was very low as it has multiple models and shapes.

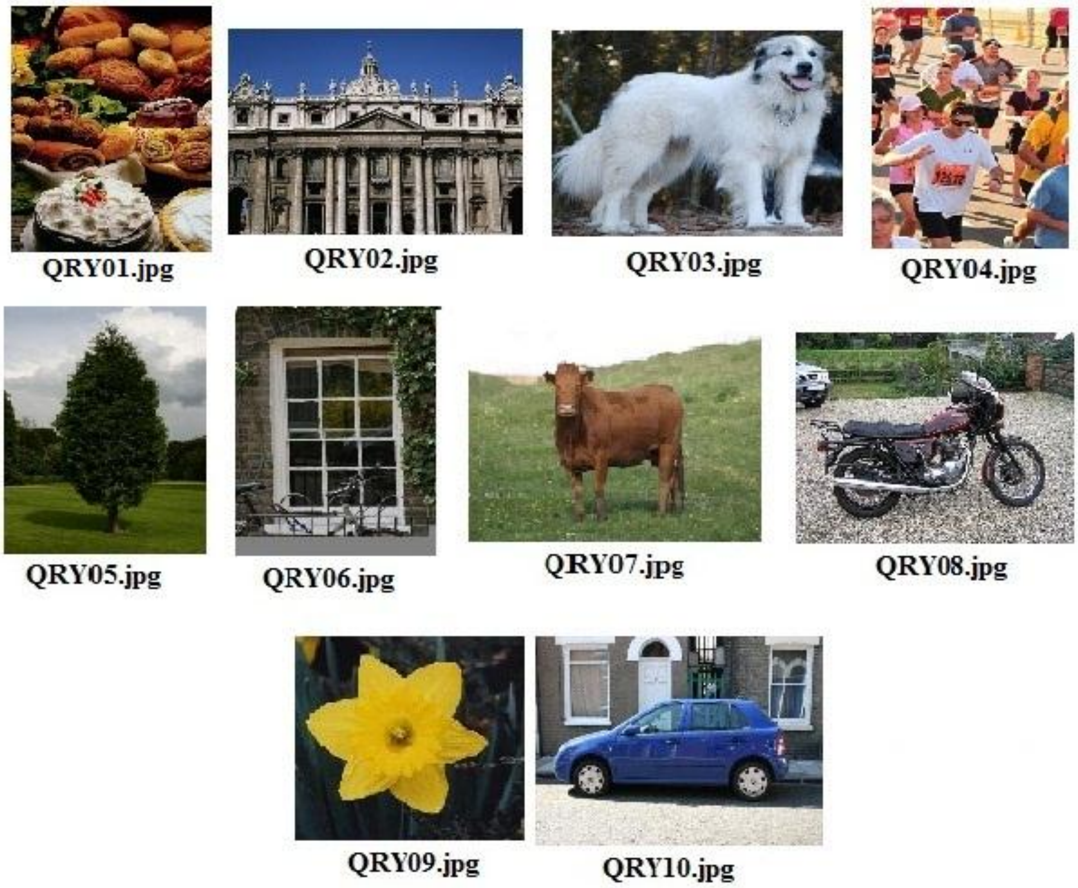

Figure 4.13: Image queries for testing the SURF combined with FLANN and the first database.

Table 4.16: Number of relevant files using the SURF combined with FLANN and the first database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	QRY01.jpg	2	4	6
Query #2	QRY 02.jpg	7	12	15
Query #3	QRY 03.jpg	1	1	1
Query #4	QRY 04.jpg	3	3	3
Query #5	QRY 05.jpg	8	17	26
Query #6	QRY 06.jpg	10	20	28
Query #7	QRY 07.jpg	5	9	11
Query #8	QRY 08.jpg	0	2	3
Query #9	QRY 09.jpg	5	7	10
Query #10	QRY 10.jpg	6	9	11

Experiment #2: Performance using second database (2500 images)

In this experiment, we have used a database that contains 2500 images, the query images of this database are shown in Figure 4.14. Table 4.17 shows the number of relevant image for the 10 queries. Generally, this approach only focuses on the interest point in the images. The percentages of the relevant files for most of the queries were less than 50%. This method cannot find the exact interest point for the same objects that have been rotated, as is the case of the ninth and tenth queries. On the other hand, this method cannot deal with objects that have multiple and different shapes. For instance, the number of relevant files for the eighth query of the category ‘cloud’ was low, as clouds can have different random shapes, which means that images of the category ‘cloud’ are represented by different interest points.

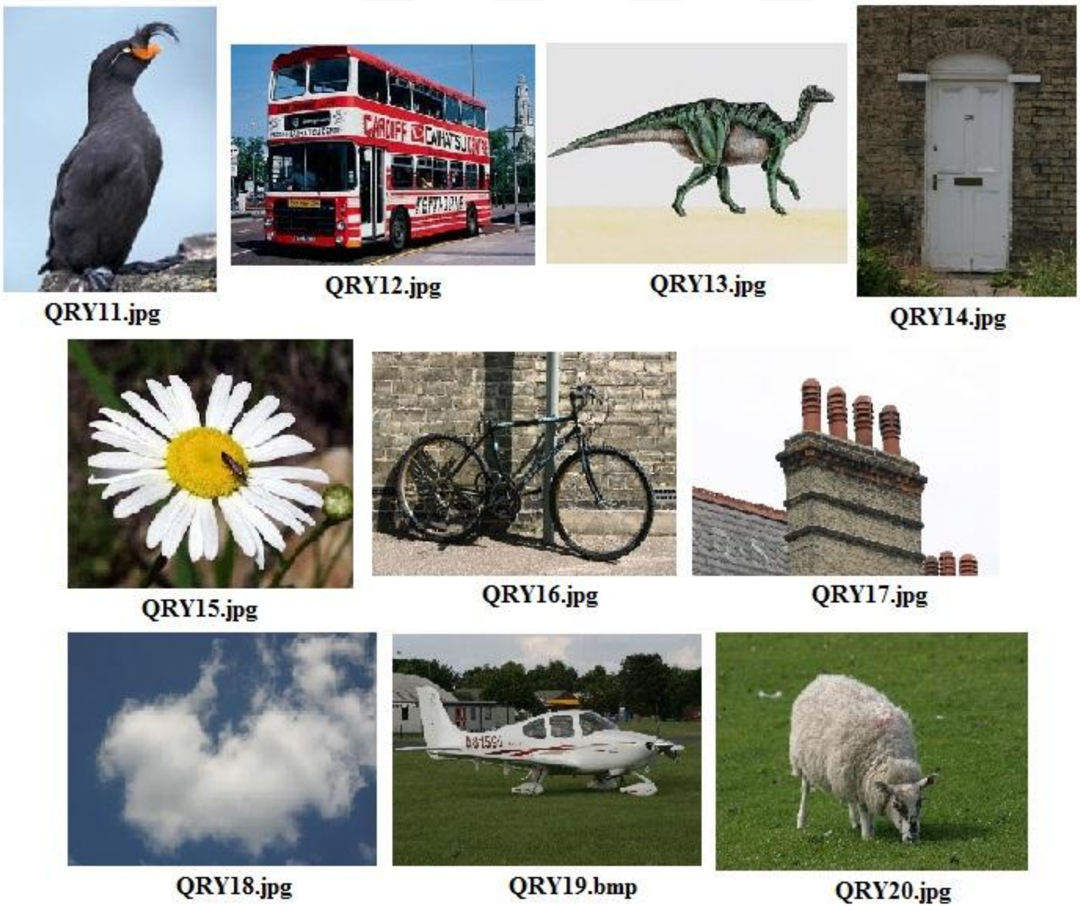

Figure 4.14: Image queries for testing the SURF combined with FLANN and the second database.

Table 4.17: Number of relevant files using the SURF combined with FLANN and the second database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	QRY11.jpg	2	3	4
Query #2	QRY 12.jpg	2	4	5
Query #3	QRY 13.jpg	8	12	17
Query #4	QRY 14.jpg	8	13	16
Query #5	QRY 15.jpg	10	16	18
Query #6	QRY 16.jpg	3	8	12
Query #7	QRY 17.jpg	4	11	13
Query #8	QRY 18.jpg	2	3	3
Query #9	QRY 19.jpg	5	7	9
Query #10	QRY 20.jpg	2	3	3

Experiment #3: Performance using a 5000 images database

In this experiment, we have used a database that contains 5000 images, the query images of this database are shown in Figure 4.15. The number of relevant files for each query was found, and shown in Table 4.18. The numbers of relevant files of all queries were under user expectations. Hence, by increasing the size of the database, the number of irrelevant images also increases.

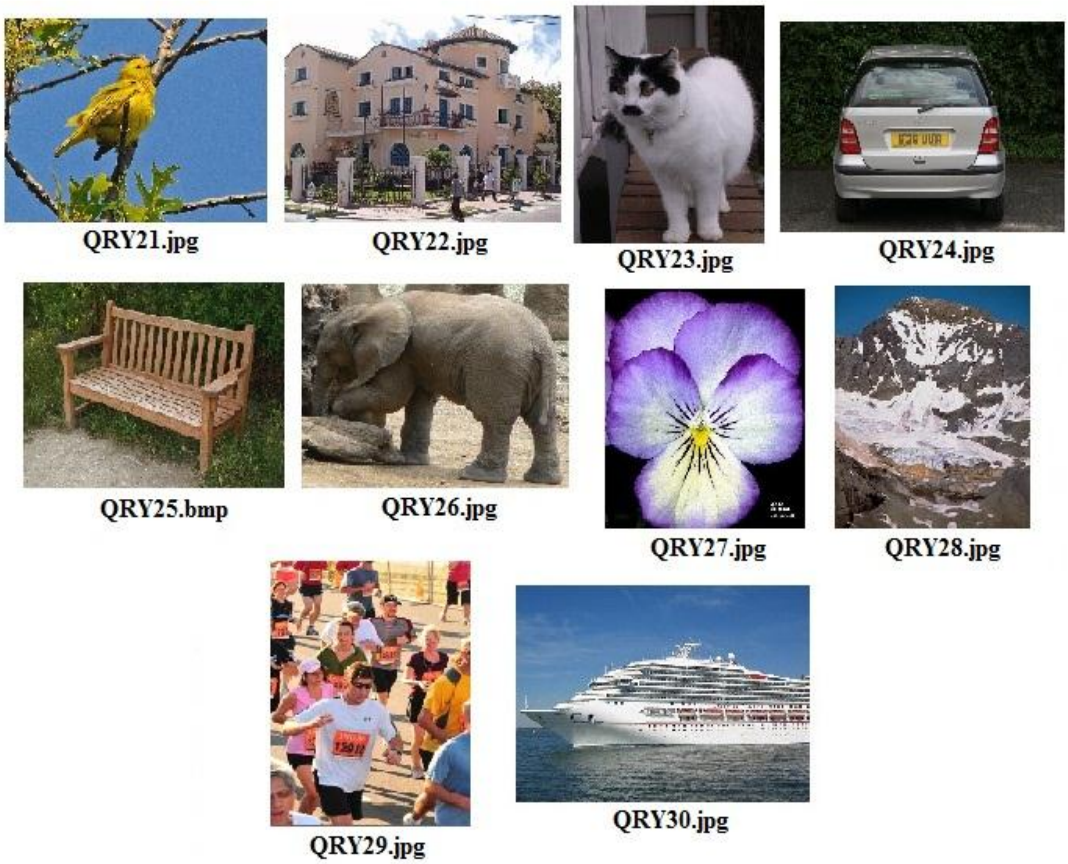

Figure 4.15: Image queries for testing the SURF combined with FLANN and the third database.

Table 4.18: Number of relevant files using the SURF combined with FLANN and the third database.

Query #	Query Image	Number of relevant files		
		First 10 files	First 20 files	First 30 files
Query #1	QRY21.jpg	1	1	1
Query #2	QRY 22.jpg	2	4	5
Query #3	QRY 23.jpg	0	0	1
Query #4	QRY 24.jpg	4	7	8
Query #5	QRY 25.jpg	3	7	8
Query #6	QRY 26.jpg	1	1	1
Query #7	QRY 27.jpg	1	2	2
Query #8	QRY 28.jpg	0	1	1
Query #9	QRY 29.jpg	1	3	5
Query #10	QRY 30.jpg	3	7	8

4.6. Overall Result Discussion

In general, the EHD uses only texture features in image retrieval. However, the edges are significant features to represent the content of images because the human eyes are sensitive to boundaries of features for perception of images. But, the edge vectors fail to discriminate the constituent parts of the images. For example, in the first database for the first query of the category 'cow', some of the retrieved images belonged to the 'sheep' category, an animal which has almost the same shape as that of a 'cow' as shown in Figure 4.16. Hence, the EHD cannot distinguish between different objects that have almost the same shape. Therefore, the edge feature is not sufficient to be used alone in a retrieval system.

Figure 4.16: Retrieval results based on image query of category 'Cow' using EHD.

On the other side, to our knowledge there is no edge based descriptor that can simultaneously filter directions of all an image's all edges. For example, in the third database, for the third query of the category 'car', the number of relevant images is the highest compared to other queries. This is because the component of the query image regular.

In addition, CEDD and FCTH methods incorporate color texture and color features in a single histogram, whose colors information in both techniques is the same, but texture information will mainly determine the capability and suitability of each approach. In this study, it has been shown that the performance of the FCTH descriptor overcame that of the CEDD descriptor. However, the performance of texture features in FCTH can be corrupted when it is required to deal with images that have many objects.

Moreover, the SCD and CLD descriptors use color information in image retrieval. On one side, the CLD descriptor considers local color information, and does not consider edge information, however, it still has adequate accuracy for many queries. On the other hand, the SCD does not consider local information and edge information, and still the accuracy of this descriptor is good. It is important to note that color descriptors cannot find similar objects in the compared images if the objects have different colors. For instance, if the compared images have colored cows such as black, white, or brown, it is impossible to retrieve all of these images based on color.

Furthermore, the SURF focuses on local attribute detection to extract interest points that are not change by fluctuating rotation and scale, and can be used to identify similarities between the images objects. However, this method cannot detect similar objects that have different shapes and it is difficult to find resemblances among such objects based on this approach.

It is important to note that all the studied techniques does not have the ability to show accurate similarities for objects that have multiple shapes and designs, for instance, these techniques could not show images shown in Figure 4.17, as a result for chair query.

Based on the above results, among all compared descriptors, the SCD and FCTH have the most relevant results in all databases. However, it has been observed that the accuracy of these descriptors is decreased by increasing the size of databases and/or increasing the number of image categories.

Figure 4.17: Sample images of category 'Chair'.

In conclusion, the CBIR systems use color, texture, and shape features as three basic means to obtain these goals. The right features play a significant role in retrieval system and these features of the images as accuracy and uniquely as suitable. In addition, the features are chosen to describe the objects of the images that are needed as discriminative. In my opinion, each of the studied descriptors has pros and cons, however, none of these descriptors can be used alone to build a truly accurate image retrieval system.

CHAPTER 5

DISCUSSION AND CONCLUSION

Multimedia data has been growing rapidly in recent years, mostly due to the evolution of information technology. Nowadays, the image search engine is widely used with the proliferation of capture devices and the growth in digital images on the internet. The main role of this type of search engine is to retrieve and show the user relevant images. Text based image retrieval (TBIR), finds relevant files based on the query keyword (s). Finding images based on the text has some problems, for instance, sometimes the captions of images and/ or the corresponding link tagging may not actually be related to the content of the images themselves. This might lead the users to obtain irrelevant, duplicate, and undesired images. To solve such problems, the CBIR system, which extracts features that describe the visual content of multimedia has been developed. In general, CBIR systems use color, texture, and shape features to describe the visual content of multimedia.

In this study, the performance of FCTH, EHD, SCD, CLD, CEDD, and SURF combined with FLANN content based techniques have been investigated. The first five methods were tested using three databases that contained 10,000; 50,000; and 100,000 images and, due to the lack of super servers, i.e., high quality computers, the SURF combined with FLANN was tested using smaller databases that contained 1000, 2500, and 5000 images. In addition, for each database, 10 query images were chosen randomly and the performance evaluation of all the methods was computed using the first 10, 20, and 30 image results for each query.

The FCTH descriptor combines color information and texture information in a single histogram, which is helpful to get good results. However, this technique cannot distinguish between objects that have almost the same properties such as shapes and borders.

The EHD method uses the edge feature to extract and detect object boundaries. The performance of this approach can be acceptable, if the borders of the object of the query image are regular, and clear, and/or if the query image has few objects. However, the overall performance of this method was below expectations. This is because of the fact that the EHD method divides the image into 16 sub-images and these sub-images are further divided into a number of image-blocks. This process may lead to inadvertently discarding important parts and features of the image objects. In addition, based on predefined thresholds, each image-block is represented by only one type of edge, however most times the image's blocks contain numerous types of edges. Hence, the performance of this descriptor decreases when it is required to deal with objects that have unsymmetrical boundaries.

The SCD descriptor uses only color information in image retrieval. Overall, the accuracy of this technique is good. However, it cannot find similar objects that have different colors. In addition, SCD has some difficulties in distinguishing between different objects that have the same color.

The CLD method represents any image using the distribution of colors in that image. The CLD descriptor uses only a color feature. This method can find the rotated and scaled file, which are relevant to the query image. However, the number of irrelevant images can be high when the compared objects have the same colors.

The CEDD descriptor, extracts multiple attributes that include color and texture information which are combined in one histogram. Therefore, the results of this descriptor were widely accepted than the methods that use only a single feature type, such as EHD and CLD descriptors. On the other hand, as CEDD incorporates color and texture features in a single histogram, the drawback of this histogram is that it cannot precisely describe an image's complicated objects. Addition, it cannot differentiate between two objects that have the same color and boundaries.

The SURF combined with FLANN uses the interest points that are located at junctions, speckles, and corners of the objects. These interest points can be used in CBIR systems, because they are not changed by the fluctuating rotation and possible differences in size. However, it taken a massive memory and most times the interest points are not sufficient enough to find similarities among the image's objects that have and can be represented by multiple models and/or shapes.

Based on the results obtain using all the databases, FCTH and SCD descriptors were selected as the best descriptors. Hence, they have obtained the most relevant files compared to the other descriptors. On the other hand, for the SURF combined with FLANN approach, the results of most of the queries were below user expectations. In addition, extracting the feature vectors using this method requires a massive memory. Overall, none of the studied CBIR descriptors can be used individually to build a full image retrieval system. In our opinion, multiple descriptors can be used to achieve a more robust system and accurate results.

As future work, the study presented in this thesis can be further improved by testing the performance of integrating multiple descriptors and building an image retrieval system which uses the best group of descriptors simultaneously.

REFERENCES

[1] Chatzichristofis, S. A., & Boutalis, Y. S. (2008, May). FCTH: Fuzzy color and texture histogram-a low level feature for accurate image retrieval, In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on* (pp. 191-196). IEEE.

[2] Won, C. S., Park, D. K., & Park, S. J. (2002). Efficient use of MPEG-7 edge histogram descriptor, *ETRI journal*, **24**(1), 23-30.

[3] Manjunath, B. S., Ohm, J. R., Vasudevan, V. V., & Yamada, A. (2001). Color and texture descriptors, *IEEE Transactions on circuits and systems for video technology*, **11**(6), 703-715.

[4] Jalab, H. A. (2011, September). Image retrieval system based on color layout descriptor and Gabor filters, In *Open Systems (ICOS), 2011 IEEE Conference on* (pp. 32-36). IEEE.

[5] Kumar, P. P., Aparna, D. K., & Rao, K. V. (2012, October). Compact Descriptors For Accurate Image Indexing And Retrieval: Fcth And Cedd, *International Journal of Engineering Research and Technology (IJERT),* **1**(8).

[6] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF), *Computer vision and image understanding*, **110**(3), 346-359.

[7] Bruce, J. (2013, April 17). *How Do Search Engines Work?* Retrieved from http://www.makeuseof.com/tag/how-do-search-engines-work-makeuseof-explains/

[8] Search Engine Watch. (2007, March 14). *How Search Engines Work?* Retrieved from https://searchenginewatch.com/sew/news/2065173/how-search-engines-works

[9] Duan, L., Li, W., Tsang, I. W. H., & Xu, D. (2011). Improving web image search by bag-based reranking, *Image Processing, IEEE Transactions on*, **20**(11), 3280-3290.

[10] Syed H. S., Arif I. U., Saeeda N., Noor ul A. Kh., Muhammad I. R., & Bandar AlHaqbani., (2016) Content-Based Image Retrieval Using Texture Color Shape and Region, *International Journal of Advanced Computer Science and Applications (IJACSA),* **7**(1), 418-426.

[11] Joshi, M. D., Deshmukh, R. M., Hemke, K. N., Bhake, A., & Wajgi, R. (2014). Image retrieval and re-ranking techniques-a survey, *Signal & Image Processing*, **5**(2), 1.

[12] Pass, G., & Zabih, R. (1996, December). Histogram refinement for content-based image retrieval, In *Applications of Computer Vision, 1996. WACV'96., Proceedings 3rd IEEE Workshop on* (pp. 96-102). IEEE.

[13] Singh, S. M., & Hemachandran, K. (2012, September). Content-based image retrieval using color moment and Gabor texture feature, *International Journal of Computer Science Issues (IJCSI),* **9**(1), (299-309).

[14] Jau-Ling, S., & Ling-Hwei, C. (2002). Color image retrieval based on primitives of color moments, In *Recent Advances in Visual Information Systems*, 88-94.

[15] M.Tech V. V., Jagadeesan J., & Isaac R. A., (2014). Web Image Search Reranking Using CBIR, *International Journal of Computer Science & Engineering Technology (IJCSET)*, **5** (2229-3345), 348-360.

[16] Khokher, A., & Talwar, R. (2012, April). Content-based Image Retrieval: Feature Extraction Techniques and Applications, *In Conference proceedings, Proceedings published in International Journal of Computer Applications® (IJCA)*, 9-14.

[17] ping Tian, D. (2013). A review on image feature extraction and representation techniques, *International Journal of Multimedia and Ubiquitous Engineering*, ***8***(4), 385-396.

[18] Chow, T. W., & Rahman, M. K. M. (2007). A new image classification technique using tree-structured regional features, *Neurocomputing*, ***70***(4), 1040-1050.

[19] Grigorescu, S. E., Petkov, N., & Kruizinga, P. (2002). Comparison of texture features based on Gabor filters, *Image Processing, IEEE Transactions on*, ***11***(10), 1160-1167.

[20] Kekre, D. H., Thepade, S. D., Mukherjee, P., Wadhwa, S., Kakaiya, M., & Singh, S. (2010). Image retrieval with shape features extracted using gradient operators and slope magnitude technique with BTC, *International Journal of Computer Applications (IJCA)*, **6**(8), 28-33.

[21] Amanatiadis, A., Kaburlasos, V. G., Gasteratos, A., & Papadakis, S. E. (2011). Evaluation of shape descriptors for shape-based image retrieval, *Image Processing, IET*, ***5***(5), 493-499.

[22] Zhang, D., & Lu, G. (2004). Review of shape representation and description techniques, *Pattern recognition*, **37**(1), 1-19.

[23] Norouzi, M., Fleet, D. J., & Salakhutdinov, R. R. (2012). Hamming distance metric learning, In *Advances in neural information processing systems* (pp. 1061-1069).

[24] KABADE, A. L. & Sangam, V. G. (2016, May). Canny edge detection algorithm, *International Journal of Advance Research in Electronics and Communication Engineering (IJARECE)*, **5**(5), 1292-1295.

[25] Gao, W., Zhang, X., Yang, L., & Liu, H. (2010, July). An improved Sobel edge detection, *In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference*, **5**, 67-71

[26] Mei, T., Rui, Y., Li, S., & Tian, Q. (2014). Multimedia search reranking: A literature survey, *ACM Computing Surveys (CSUR)*, ***46***(3), 38.

[27] Jain, H., Lodha, R., & Deulkar, K. (2014). Web-based Image Search using Image Re-Ranking Technique: A Review Paper, *International Journal of Current Engineering and Technology*, **4**(5), 3300-3303.

[28] Wang, X., Qiu, S., Liu, K., & Tang, X. (2014). Web Image Re-Ranking Using Query-Specific Semantic Signatures, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, ***36***(4), 810-823.

[29] Wang, X., Liu, K., & Tang, X. (2011, June). Query-specific visual semantic spaces for web image re-ranking, *In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (pp. 857-864). IEEE.

[30] Pedronette, D. C. G., & Torres, R. D. S. (2013). Image re-ranking and rank aggregation based on similarity of ranked lists, *Pattern Recognition*, ***46***(8), 2350-2360.

[31] Ordonez, V., Han, X., Kuznetsova, P., Kulkarni, G., Mitchell, M., Yamaguchi, K., Stratos, K., Goyal, A., Dodge, J., Mensch, A., & Daumé III, H. (2015). Large scale retrieval and generation of image descriptions, *International Journal of Computer Vision*, **119**(1), 46-59.

[32] Yang, L., & Hanjalic, A. (2012). Prototype-based image search reranking, *Multimedia, IEEE Transactions on*, **14**(3), 871-882.

[33] Yu, J., Rui, Y., & Chen, B. (2014). Exploiting click constraints and multi-view features for image re-ranking, *Multimedia, IEEE Transactions on*, **16**(1), 159-168.

[34] Chatzichristofis, S. A., Zagoris, K., Boutalis, Y. S., & Papamarkos, N. (2010). Accurate image retrieval based on compact composite descriptors and relevance feedback information, *International Journal of Pattern Recognition and Artificial Intelligence*, **24**(2), 207-244.

[35] Park, D. K., Jeon, Y. S., & Won, C. S. (2000, November). Efficient use of local edge histogram descriptor, *In Proceedings of the 2000 ACM workshops on Multimedia* (pp. 51-54). ACM.

[36] Balasubramani, R., Kannan, D. V. (2009). Efficient use of MPEG-7 color layout and edge histogram descriptors in CBIR systems, *Global Journal of Computer Science and Technology*, ***9***(4), 157-163.

[37] Yaegashi, Y., Tateoka, K., Fujimoto, K., Nakazawa, T., Nakata, A., Saito, Y., Abe, T., Yano M., & Sakata, K. (2012). Assessment of similarity measures for accurate deformable image registration, *Journal of Nuclear medicine and Radiation Therapy*, ***3***(4).

[38] Petkos, G., Schinas, M., Papadopoulos, S., & Kompatsiaris, Y. (2017). Graph-based multimodal clustering for social multimedia, *Multimedia Tools and Applications*, **76**(6), 7897-7919.

[39] Malik, F., & Baharudin, B. (2013). 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*, **25**(2), 207-218.

[40] Li, J., & Wang, J. Z. (2003). Automatic linguistic indexing of pictures by a statistical modeling approach, *IEEE Transactions on pattern analysis and machine intelligence*, ***25***(9), 1075-1088.

[41] Wang, J. Z., Li, J., & Wiederhold, G. (2001). SIMPLIcity: Semantics-sensitive integrated matching for picture libraries, *IEEE Transactions on pattern analysis and machine intelligence*, ***23***(9), 947-963.

[42] Huiskes, M. J., Thomee, B., & Lew, M. S. (2010, March). New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative, In *Proceedings of the international conference on Multimedia information retrieval* (pp. 527-536). ACM.

[43] Einstein, A., Podolsky, B., & Rosen, N. (1935). Can quantum-mechanical description of physical reality be considered complete?, *Physical review*, ***47***(10), 777-780.

[44] Microsoft. (2016 December 30). Microsoft Research Cambridge Object Recognition Image Database. Retrieved from "https://www.microsoft.com/en-us/download/details.aspx?id=52644".

[45] Alqaraleh, S., & Ramadan, O. (2015). Utilizing Query by Example for Fast and Accurate Multimedia Retrieval, *Applied Mathematics & Information Sciences*, ***9***(1), 125.

[46] Alqaraleh, S., & Ramadan, O. (2014). Elimination of repeated occurrences in multimedia search engines, *Int. Arab J. Inf. Technol.*, ***11***(2), 134-139.

[47] Alqaraleh, S., Ramadan, O., & Salamah, M. (2015). Efficient watcher based web crawler design. *Aslib Journal of Information Management*, ***67***(6), 663-686.