IMAGE CLASSIFICATION FOR CONTENT BASED INDEXING

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

SERDAR TANER

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

DECEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan Özgen
Director

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

Prof. Dr. Mübeccel Demirekler
Head of Department

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

Prof. Dr. Mete Severcan
Supervisor

Examining Committee Members

Assoc. Prof. Dr. Aydın Alatan _____

Prof. Dr. Mete Severcan _____

Prof. Dr. Yalçın Tanık _____

Assoc. Prof. Dr. Gözde Bozdağı Akar _____

M.Sc. Şener Yılmaz _____

ABSTRACT

IMAGE CLASSIFICATION FOR CONTENT BASED INDEXING

TANER, Serdar
M.Sc. , Department of Electrical and Electronics Engineering
Supervisor: Prof. Dr. Mete Severcan

December 2003, 97 Pages

As the size of image databases increases in time, the need for content based image indexing and retrieval become important. Image classification is a key to content based image indexing. In this thesis supervised learning with feed forward back propagation artificial neural networks is used for image classification. Low level features derived from the images are used to classify the images to interpret the high level features that yield semantics. Features are derived using detail histogram correlations obtained by Wavelet Transform, directional edge information obtained by Fourier Transform and color histogram correlations. An image database consisting of 357 color images of various sizes is used for training and testing the structure. The database is indexed into seven classes that represent scenery contents which are not mutually exclusive. The ground truth data is formed in a supervised fashion to be used in training the neural network and testing the performance. The performance of the structure is tested using leave one out method and comparing the simulation outputs with the ground truth data. Success, mean square error and the class recall rates are used as the performance measures. The performances of the

derived features are compared with the color and texture descriptors of MPEG-7 using the structure designed. The results show that the performance of the method is comparable and better. This method of classification for content based image indexing is a reliable and valid method for content based image indexing and retrieval, especially in scenery image indexing.
Keywords: Artificial neural network, wavelet transform, histogram correlation, directional edge information, leave one out method, MPEG-7

ÖZ

İÇERİK TABANLI İNDEKSLEME AMAÇLI GÖRÜNTÜ SINIFLANDIRMA

TANER, Serdar
Yüksek Lisans , Elektrik-Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Mete Severcan

Aralık 2003, 97 sayfa

Görüntü veritabanlarının hacmi arttıkça içerik tabanlı görüntü indekslemeye olan ihtiyaç önem kazanmıştır. Görüntü sınıflandırma içerik tabanlı görüntü indeksleme ve bulmaya bir çözüm yoludur. Bu tezde görüntü sınıflandırması için ileri beslemeli geri üremeli yapay sinirsel ağlar kullanılmıştır. Görüntülerden elde edilen düşük seviyeli belirleyici nitelikler görüntülerin anlam içeren yüksek seviyedeki özelliklerinin çıkarılması amacıyla görüntü sınıflandırmasında kullanılmıştır. Özellikler Wavelet dönüştürmesiyle elde edilen detay histogram korrelasyonları, Fourier dönüştürmesiyle elde edilen yönlü kenar çizgileri bilgisi ve renk histogram korrelasyonları kullanılarak elde edilmiştir. Çeşitli boyutlardaki 357 renkli resimden oluşan bir görüntü veri bankası yapının eğitilmesi ve test edilmesinde kullanılmıştır. Veri bankası birbirinden tamamen ayrık olmayan manzara içeriklerini belirten 7 sınıfa indekslenmiştir. Temel doğru data, yapay sinirsel ağın eğitilmesi ve test edilmesi için denetlemeli bir şekilde oluşturulmuştur. Yapının performansı, birini dışarıda bırakma yöntemi kullanılarak, simülasyon sonuçları ve temel doğru datanın karşılaştırması yöntemiyle yapılmıştır. Performans

ölçüleri olarak başarı, hatanın karesinin ortalaması ve sınıf geri çağırma oranları kullanılmıştır. Geliştirilen belirleyici niteliklerin performansları, MPEG-7 resim ve doku tanımlayıcı performansları ile tasarlanan yapı kullanılarak karşılaştırılmıştır. Sonuçlar yöntemin performansının karşılaştırılabilir ve daha iyi olduğunu göstermektedir. Bu içerik tabanlı indeksleme amaçlı görüntü sınıflandırma metodu içerik tabanlı görüntü indekslemek ve bulmak, özellikle de manzara görüntüsü indekslemek için güvenilir ve geçerli yöntemdir.

Anahtar Kelimeler: yapay sinirsel ağlar, wavelet dönüşümü, histogram korrelasyonları, yönlü kenar çizgisi bilgisi, birini dışarıda bırakma yöntemi, MPEG-7

to my family, Ege and Tuna

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Prof. Dr. Mete Severcan for sharing his invaluable knowledge, guidance and support in this research I would like to thank to Aselsan A.Ş. for letting me to involve in this thesis study and my work friends for their helps.

Finally I would like to thank my family and friends, for their great support, help and understanding.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

2D : Two dimensional
CBIIR : Content Based Image Indexing and Retrieval
CD : Diagonal Detail Coefficients
CH : Horizontal Detail Coefficients
CHC : Color Histogram Correlations
CLD : Color Layout Descriptor
CSD : Color Structure Descriptor
CV : Vertical Detail Coefficients
Db3 : Daubechies 3
DCD : Dominant Color Descriptor
DF : Directional Filtering
DFT : Discrete Fourier Transform
DHC : Detail Histogram Correlations
DWT : Discrete Wavelet Transform
EHD : Edge Histogram Descriptor
HMMD : Hue Max Min Diff
HSV : Hue Saturation Value
HTD : Homogeneous Texture Descriptor
MPEG-7 : Standard for Multimedia Content Description Interface
NNET : Artificial Neural Network
PCA : Principal Component Analysis
RGB : Red Green Blue
S1 : Number of outputs in the hidden layer of a NNET
SCD : Scalable Color Descriptor
TBD : Texture Browsing Descriptor
YCbCr : Luminance Chrominance

CHAPTER 1

INTRODUCTION

Images provide valuable visible information for a wide range of disciplines such as astronomy, architecture, engineering, biomedical, or communication. Digital imaging makes it possible to hold large databases for these disciplines. Digital image databases of growing sizes spread over the world with the huge number of new images added day by day. Databases may be founded for amateur or professional intentions. Whatever the intention databases are founded for, these databases require efficient storing mechanisms to retrieve desired images since browsing gets harder and harder with the increasing sizes of databases. The need for data management increased substantially parallel to the enlargements in database sizes. Because of the difficulties in retrieving desired images from huge amounts of data, some form of indexing or cataloguing becomes a must. Content Based Image Indexing and Retrieval (CBIIR) is the automatic retrieval of desired images from image databases on the basis of features.

1.1 DIGITAL IMAGE DATABASES AND CBIIR

In a large database of images from various kinds, it is difficult to retrieve a desired image. It may be feasible to retrieve a desired image from a small database by browsing, but more effective techniques are necessary for large databases in applications such as requesting images of a particular event, a specified color or texture, or a desired object match.

By investigating the user needs, some knowledge on characteristics of queries for Image Retrieval can be obtained. It is important to know, the aim in seeking image and the usage area of the image to build a structure for query characteristics.

Being a foregoing study without certain results, it can be seen that amateurs or professionals require images for applications such as illustrating text, expressing information or emotions, displaying images for analysis, recording the data for later use or storing the photographs.

To retrieve a desired image from a database it is required to analyze the requirements of the retrieval. An image that involves a specific object or scene, emotion, texture or pattern may be desired for retrieval. Queries may be formed so that they may require a particular combination of color or shape features (e.g. orange circle in the left corner), a specific type of object (e.g. parallel Roman Columns), a particular type of event (e.g. soldering process of printed boards), or named objects or locations (e.g. the Bosphorus).

The mentioned query types represent a graded complexity of image retrieval. This leads naturally a classification of query types into three levels of increasing complexity [1, 2]. Retrieval based on primitive features such as color, texture, shape or the spatial location of image elements is the Level 1. Examples of such queries might include finding images with star like shapes at the center, images of orange disks with black background or similar images of the query image. These features are objective and can be derived from the image directly with limited use to specialist applications such as identification of designs or color and texture matching of materials.

Retrieval by derived (logical) features involves some degree of logical inference about the objects in the image. This kind of retrieval is called Level 2 and it involves retrieval of specified events (e.g. “find pictures of a sailing boat voyaging at sea”) or retrieval of individual objects or persons (“find a picture of Pisa tower”). Answering queries at this level, a logical use of knowledge on desired objects is required; such as Pisa Tower that is a sloped vertical structure with cream color and a texture of columns. Level 3 kind of retrieval corresponds to retrieval by abstract attributes. It involves understanding the event in image and commenting on it such as retrieving emotions. Level 2 and 3 retrieval are together named as semantic image retrieval [1, 2].

The common primitive features extracted from images are mathematical measures of color, texture and shape. A typical system may allow users to formulate

queries by submitting a typical example of the image that is being sought, though some offer alternatives such as selection from a palette or sketching the input. The system then retrieves the images, feature values of which match those of the query most closely. Color, texture and shape are commonly used types of primitive features.

A block diagram of classification of CBIIR based on feature level is given in Figure 1.1. The raw image is taken as input. Primitive features from the image are processed using signal processing. Primitive features such as color histogram or texture descriptor are extracted from the image so that retrieval based on this features can be made. The primitive features or the image can be used by computer vision for retrieving derived features (Level 2 Retrieval). In this kind of retrieval the images can be retrieved based on the objects in the images and the relationships between these objects such as scenes. The Intelligent modeling based on Concepts (Level III features) can be derived using artificial intelligence and philosophy on the derived features. It should be noticed that this kind of retrieval is rare and subjective to the user [1].

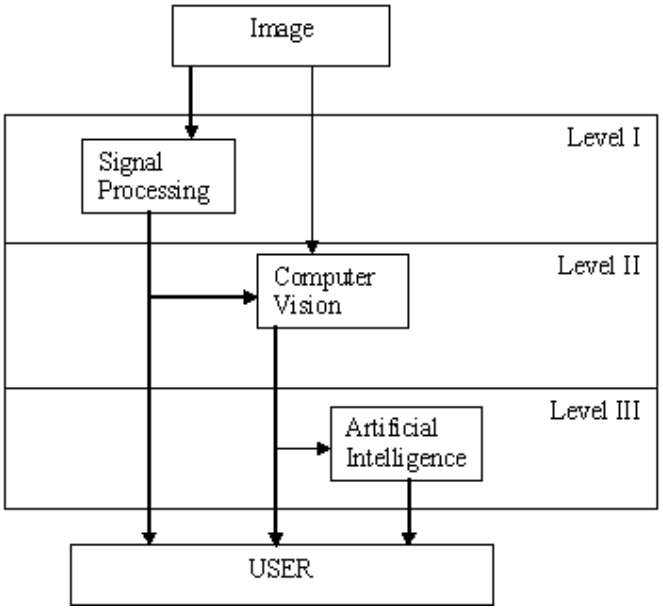

Figure 1.1 Block Diagram of Classification of Content Based Retrieval

In Content Based Image Indexing and Retrieval, retrieval is performed based on a query. The query may be the primary features or the derived features of an

image. Primary features are the features like color, texture and shape which could be retrieved from the image utilizing a low level of signal processing. Derived features are the semantic features derived from the images or from the primary features using the computer vision. Queries of primary features such as a color combination, or a desired texture need professional knowledge and queries of derived features are subjective, alternative kinds of queries are also used in content based image retrieval. Examples of such queries are query by example image or query by sketching. With such queries image retrieval is performed based on similarity or content indexing. The literature survey on CBIIR is discussed in detail in Chapter 2.

1.2 PROBLEM DEFINITION

Human vision easily classifies the images based on their semantic content. For computer vision classification of the images based on semantic features is a hard task. The problem is not only to find the indexing structure that is harmonious with the human vision but also to implement the indexing structure. Perfect classification structure to reflect the human vision is a very complicated and hard task, and there are no examples of such a complete task in the literature. Image classification, based on the user needs, is a key to this problem, so that the decision space may be divided into the classes of necessary content in a supervised fashion.

In this thesis, it is tried to obtain high level features (semantic descriptors) using the low level features derived from the image. Classification of the images is used to reach the image content. An artificial neural network (NNET) is used to classify the images. The proposed method is tested with seven different semantic classes. These classes are buildings, flowers, mountains, sea, river-waterfall, sunset and autumn-forest. To extract the low level features from the image; color histogram correlations, detail histogram correlations and directional filtering are used. The performance of the system is tested by comparing the outputs of simulations with the ground truth data formed in a supervised fashion.

Selecting the suitable features and designing a good classifier is important for system performance so that tests are held to optimize the NNET and the feature parameters for best performance and maximum stability. The stability of NNET output depends on the available data in training database as well as the NNET parameters used in design. To use maximum available data, leave one out method is

used to train the NNET and to test the performances. The performances of extracted features are compared with the performances of color and texture features/descriptors of MPEG-7 using the classification structure designed.

1.3 CONTRIBUTIONS AND THESIS ORGANIZATION

In this thesis, we developed a color descriptor based on color histogram correlations, a texture descriptor based on detail histogram correlations of multi scale decomposition and a compact color-texture mixed feature based on measuring energy over frequency channels. The features proposed are integrated to form the feature vector for the image. An artificial neural network classifier is developed to classify the images based on their contents. MPEG-7 color and texture descriptors are extracted. The experiments of classifier with the evaluation of features are presented. The developed features are compared with descriptor set of MPEG-7 using the structure proposed.

The outline of the thesis is as follows:

- In Chapter 2, a background on Content Based Image Indexing is given with the discussion of MPEG-7. The MPEG-7 color and texture descriptors are reviewed with the extraction methods,
- In Chapter 3, image features obtained from the images for CBIIR are discussed, with the emphasis on the Feature Vector formed,
- In Chapter 4, image classification using artificial neural networks is given with the discussion of the classification structure used,
- In Chapter 5, the tests and the test results are discussed,
- In Chapter 6, the final conclusions on this study are made and the further work on this area is proposed.

CHAPTER 2

BACKGROUND ON CONTENT BASED IMAGE INDEXING AND RETRIEVAL

Image database is a collection of images. With the techniques that develop the storage and the computation of the digital images, the volume of digital images increased dramatically. Although this is a good development for images to be available as sources or targets for applications, the availability of these images as sources for these purposes necessitate the use of effective data management. The source that can not be achieved is as bad as the lack of the source. Achieving/Retrieving images from digital libraries by browsing is efficient for small databases, but an automatic retrieval system is required for larger sizes of databases. Content Based Image Indexing and Retrieval is the retrieval of desired images from an image database based on automatically derived features [10].

The retrieval may be performed based on low level features or high level features (semantics). For human vision, detecting high level features directly from the images is an easy task. On the other hand, low level features are available for the computer vision and some work is necessary to retrieve high level features from low level features.

The primitive features extracted from images are color, texture and shape. The high level features are the subjective features that give out the content. The gap between low level features and high level features is called the semantic gap which can be connected by image classification based on learning.

In this chapter the image representation is discussed in Section 2.1, Image classification is discussed in Section 2.2. Several existing CBIIR systems and applications of CBIIR are reviewed in Section 2.3. The standard for Multimedia

Content Description Interface, MPEG-7, is reviewed in Section 2.4. The color and texture descriptors of MPEG-7 are discussed in Section 2.5.

2.1 IMAGE REPRESENTATION & INTERPRETATION

A digital image is a collection of binary data in pixels. A color image is represented as a two dimensional (2D) signal over color spaces such as Red Green and Blue in additive color system. Computations on the whole image require a high computational load. Representing image with the required features is enough for interpreting the image using low level features and this requires less computational load. In CBIIR images can be represented with the primitive features such as color, texture, shape and etc... The best representation is an active research area named as feature extraction. In MPEG-7 the descriptors/features are standardized. MPEG-7 Visual Descriptors are discussed in Section 2.5.

In the literature several methods for retrieving images on the basis of color similarity have been described. Red Green Blue (RGB), Luminance Chrominance (YCbCr), Hue Saturation Value (HSV) and Hue Max Min Diff (HMMD) are common examples of used color spaces. Basic color descriptors are color histogram, color moments, color lay-out and etc. In color histogram, colors in the image are mapped into a discrete color space of predefined number of colors. The population of colors is calculated, that is the number of pixels on the quantization levels that are mapped to bins. It's vulnerable to quantization. Color moments represent the dominant features of image color distribution with a small representative set of color vectors that capture the color properties of the image [36]. Due to the lack of spatial information in color histogram and color moments, sub-block histograms are proposed. Color correlograms are used to extract global distribution of local correlations with a spatial correlation of colors [11].

The texture descriptor provides measures of properties such as smoothness, coarseness, and regularity. Retrieval of images based on texture similarity is helpful in distinguishing between areas of images with similar colors such as sea and sky, or soil and farm. The texture of a region can be described using statistical, structural and spectral approaches. Statistical approaches involve the characterization of textures as smooth, coarse, grainy and so on. Structural techniques deal with the arrangement of image primitives, such as description of texture based on regularly

spaced parallel lines. Spectral techniques are based on properties of Fourier spectrum and are used primarily to detect global periodicity in an image by identifying high energy, narrow peaks in the spectrum. Texture segmentation which is a challenging and computationally intensive task is used in deriving texture features. [2, 27]

A number of features represent the characteristics of the object shape. Shape often carries semantic information. Types of shape features are circularities, consecutive boundary segments, and directional histograms of edges. Region based and contour based shape descriptors are used as shape descriptors for 2D images in MPEG-7 [24]. Segmentation requirements restrict the availability of this descriptor to objects. Queries for shape retrieval systems are either identified by an example image or as a user sketch.

Spatial location of objects in images is a used feature in CBIIR systems. It has seldom usage on its own, though it is an effective feature when used in combination with other features such as color and shape. Other types of image features proposed as a basis for CBIIR mostly depend on complex transforms. One of them is the wavelet transform which analyzes an image at several resolutions. Wang proposes Wavelet Based Image Indexing and Retrieval as a fast and accurate CBIIR system with image queries. In this work wavelet coefficients at lowest few frequency bands and their variances are stored as feature vectors. And retrieval is based on feature vector comparison using minimization of Euclidian distance [3]. With these techniques an image can be analyzed at varying levels of detail and with the use of other primitive features provide performance increase.

Content Based Image Indexing and Retrieval include examples of scene recognition and object recognition. Primitive features are used to classify images into semantic classes for content based indexing. Semantic descriptors aim at encoding interpretations of an image which may be relevant to an application. High Level descriptors are more efficient and powerful than low level features since they have content information, but they are generally subjective features. [10]

High level primitives such as objects, roles, actions and events are identified as the abstraction of visual signs that are achieved through recognition and interpretation. Recognition is commonly based on selection of a set of low-level local

features and then applying pattern recognition techniques, and interpretation can be solved as a classification problem [16].

2.2 IMAGE CLASSIFICATION FOR IMAGE RETRIEVAL

The semantic gap is the lack of relationship between primitive features that are extracted from the images and the content that is recognized by the user. Early image retrieval systems depending on primitive features have low efficiency, because of the fact that the users think in terms of semantic concepts. Identifying semantic concepts for CBIIR makes retrieval efficient and effective. This is also aimed in MPEG-7 standardization.

Images that are organized based on human perception can significantly improve the retrieval efficiency. This organization can be implemented as content based image indexing. This leads to the classification of images for content based indexing where the classes represent (imbue) the content. Humans perceive the world with classification. On the similarities and differences received/recognized objects are grouped. Knowledge that is gained by learning is used to compare the images to find their semantics [16].

In the literature examples of researches in image classification based on various feature sets are found to describe the image content especially in scenery image classification. The two research areas are the feature/descriptor extraction for CBIIR and the classification for CBIIR.

Some of the researches on classification can be summarized as: Vailaya et al. [5, 12] use Bayesian classification with the probability density functions extracted using vector quantization for semantic scene classification of vacation images. A binary hierarchical classification is realized with classes such as: indoor vs. outdoor, city vs. landscape and sunset vs. forest vs. mountain.

Torrolba and Oliva [13] separate scenery images by generating decision boundaries using a supervised learning method based on discriminant spectral templates.

Raten et al. [14] adapted the Multiple Instance Learning paradigm using the Diverse Density algorithm as a way of modeling the ambiguity in images in order to learn the visual concepts that can be used to classify new images.

Yu and Wolf [15] use Hidden Markov models to learn statistical templates from examples for indoor/outdoor scene classification.

Xiong [16] used hierarchical multi-bottleneck class descriptor method which includes multiple fuzzy prototypes and parallel fuzzy learning schemes.

Classification methods that are used in classification for content based image indexing and retrieval can be summarized as nearest neighbour classifiers, Bayesian classifiers, support vector machines and artificial neural networks.

2.3 CBIIR SYSTEMS

CBIIR technology has many application areas increasing day by day parallel to the increase in the visual media databases. Areas such as crime prevention, the military, intellectual property, architectural and engineering design, fashion and interior design, journalism and advertising, medical diagnosis, geographical information and remote sensing systems, cultural heritage, education, training, home entertainment, and web searching are examples of applications for CBIIR[2].

Inspecting these areas it is seen that research groups are developing prototype systems, and practitioners are experimenting with the technology. Few examples of fully-operational CBIIR systems can be found because of the challenges it involves.

The existing CBIIR systems can be inspected in two groups: commercial CBIIR systems and prototype research CBIIR systems. Examples of commercial systems are Excalibur, ImageFinder, IMatch, QBIC (Query by Image Content) [17, 18] and VIR Image Engine [23]. The prototype research CBIIR systems are prototype research systems at universities or research laboratories. Examples of prototype research CBIIR systems are BlobWorld [19], NeTra [20], PhotoBook [21], and VisualSEEK [22].

2.4 MPEG-7 STANDARD OVERVIEW

The goal of MPEG-7 Standard is to allow interoperable searching, indexing, filtering and access of multimedia content by enabling interoperability among devices that deal with multimedia content description. Four types of normative elements are provided with the standard; descriptors, description schemes, description definition language and coding schemes. [24].

Descriptor is a representation of a feature that defines the syntax and semantics of the feature representation. Descriptors are designed to describe low level features such as color, texture, motion, audio energy and so forth. Extraction is expected to be automatic. Human intervention is required for high level feature production. Visual descriptors are color, texture and shape descriptors that are designed to be used in similarity based retrieval. The similarity metrics to be used with these descriptors are provided in the standard as well [24, 25]. Color and texture features are discussed in detail in Section 2.5; comparisons of these descriptors with the features developed in this thesis are given in Section 5.5.

Description Schemes specify the structure and semantics of the relationships between its components which may be both descriptors and description schemes within more complex structures [25].

Description Definition Language (DDL) is a language that specifies the description schemes. It also allows for extension and modification of existing Description Schemes. XML scheme language is adopted as the MPEG-7 DDL. Coding Schemes are the textual or binary ways to decode description [25].

In Multimedia Content Description Interface, neither the extraction of descriptors/features nor the search engine is standardized. Feature Extraction involves content analysis, feature extraction and annotation. Search engine involves searching, filtering and classification, manipulation and Summarization. What are standardized are the Descriptors, Descriptor Schemes and the Description Definition Language. The non standardized subjects are so to encourage competition in market and in design [24, 25].

2.5 MPEG-7 VISUAL DESCRIPTORS

MPEG-7 Visual descriptors are color, texture and shape descriptors. Color descriptors are Color Space descriptor, Dominant Color descriptor, Scalable Color descriptor, Color Structure descriptor and Color Layout descriptor. Texture descriptors are Homogeneous Texture descriptor, Texture Browsing descriptor and Edge Histogram descriptor. Shape Descriptors are Region Based and Contour Based shape descriptors [32]. Color descriptors and texture descriptors are discussed in this section.

2.5.1 COLOR DESCRIPTORS

Color is related to the chromatic attributes of the image. Since human visual system is sensitive to colors, color is used extensively in image retrieval. Color features have provided themselves to be powerful descriptors in object identification and extraction [25].

2.5.1.1 Color Space Descriptors

The color space descriptor specifies the color space to be used with the color descriptors. Color spaces specified in MPEG-7 are RGB, YCbCr, HSV, HMMD, monochrome and linear transformation matrix with respect to RGB [24, 25]. With the color space the color quantization descriptor specifies the partitioning of the space into discrete bins.

Luminance chrominance (YCbCr) representation is derived from RGB representation using

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & 0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.1}$$

where Y denotes the luminance and others correspond to chrominance. Hue Saturation and Value (HSV) is a nonlinear transformation over RGB which is defined over a cylinder as described in Figure 2.1 [25].

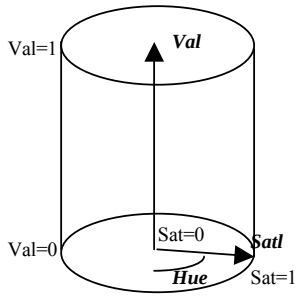

Figure 2.1 HSV Color Space

Hue takes values between 0 and 360 degrees and represents the color family. Saturation is in the range 0 to 1 and represents the pureness of the color. Value is in the range 0 to 1 and represents the brightness. For the color descriptors HSV space is

uniformly quantized into 16 to 256 bins. The partitioning of the HSV space is summarized in Table 2.1 [25]. The total number of bins is given in the heading column; the corresponding number of bins is given for the color space that is given in the heading row.

Table 2.1 Partitioning of HSV color space

Number of bins	Number of bins H	Number of bins S	Number of bins V
16	4	2	2
32	8	2	2
64	8	2	4
128	8	4	4
256	16	4	4

Hue Max Min Difference (HMMD) is a nonlinear transformation over RGB, defined as a double cone as described in Figure 2.2. Hue is derived as in HSV. Max, Min, Sum and Diff are in the range 0 to 1. Max represents the amount of black color (shade/blackness). Min represents the tint or whiteness. Diff represents the tone or colorness and sum represents the brightness.

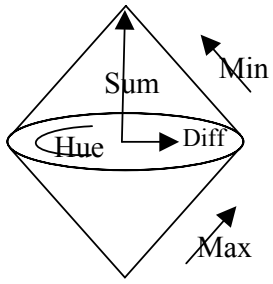

Figure 2.2 Double cone representation of HMMD color space

Partitioning of HMMD color space for 256, 128, 64 and 32 bins is given in Table 2.2. Subspaces in rows correspond to the non-uniform quantization of Diff into 5 subintervals: [0,6), [6,20), [20,60), [60,110), [110,255] which are presented in the table as subspaces 0 to 4. Hue and Sum are uniformly quantized using the Table 2.2 [25, 27]. In this table number of bins is given for the quantization levels in the

heading row. The corresponding bin number is given for the subinterval given in the heading column and the color spaces given in the second row.

Table 2.2 HMMD color space quantization

Number of cells	256		128		64		32	
Subspace	Hue	Sum	Hue	Sum	Hue	Sum	Hue	Sum
0	1	32	1	16	1	8	1	8
1	4	8	4	4	4	4	4	4
2	16	4	8	4	4	4		
3	16	4	8	4	8	2	4	1
4	16	4	8	4	8	1	4	1

2.5.1.2 Dominant Color Descriptor

Dominant color descriptor (DCD) characterizes the color information with a small number of colors. The descriptor consists of the dominant color vectors, percentages of the dominant colors and the spatial coherency [25]. The extraction of DCD as described by Manjuath et al. and Deng et al. [27, 28, 29] uses Generalized Lloyd Algorithm to cluster the pixel color values. The spatial coherency is computed as the linear combination of spatial coherency values of the dominant colors with the corresponding percentages as weights. The normalized average number of connecting pixels of each dominant color using a 3x3 masking window is used as a measure of spatial coherency for each dominant color. Main functionality of this descriptor is similarity retrieval in image databases and browsing of image databases based on single or several color values [26].

2.5.1.3 Scalable Color Descriptor

Scalable color descriptor (SCD) is the color histogram in HSV that is scaled using Haar transform [24]. Haar transform of two signals is implemented as the sum and difference of inputs. 256 bin HSV histogram is derived using Table 2.1. With the scheme described in Figure 2.3 [26], the histogram is scaled to 128, 64, 32 or 16 bins. The scaled histograms also match the partitioning in Table 2.1.

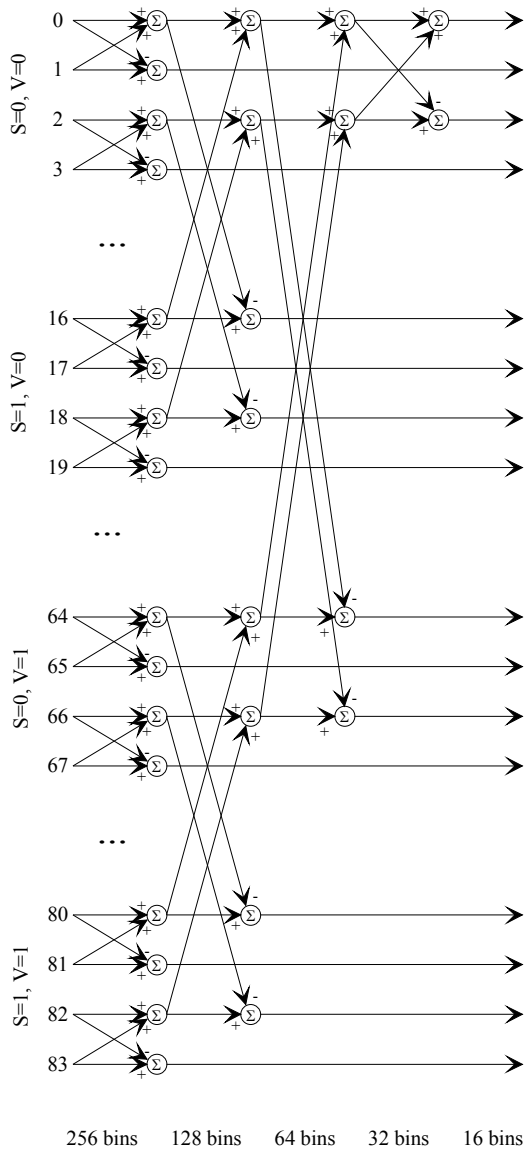

Figure 2.3 Haar Transform of 256 bin HSV Histogram

2.5.1.4 Color Structure Descriptor

Color structure descriptor (CSD) captures both the color content as in color histogram and the information about the structure of the image [24]. HMMD color space is used with CSD as the color space descriptor. CSD characterizes the relative frequency of structuring elements that contain image sample with a particular color [25]. A block schema of CSD extraction is given in Figure 2.4.

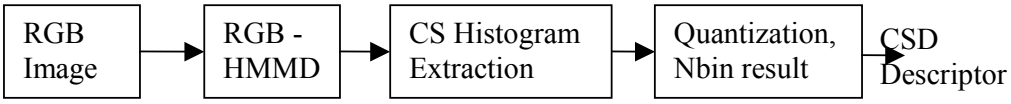

Figure 2.4 Block Schema of CSD Extraction

The image is subsampled with the subsampling factor K= 2^p where p is defined as

$$p = \max\{0, \lfloor \log_2 \sqrt{W.H} - 7.5 \rfloor\} \quad (2.2)$$

where W and H denote the width and height dimensions of the image. After the sampling, image is transformed into HMMD color space with 256 bin quantization using Table 2.2. 8x8 structure elements are used to form the color structure (CS) histogram which is populated with the existing colors in the structure. The whole image is block processed to extract the CS histogram. Bin unification and bin value quantization is used for N bin histograms for N is different from 256. Main functionality of CSD is image to image matching and image retrieval [25, 26].

2.5.1.5 Color Layout Descriptor

Color Layout Descriptor (CLD) specifies a spatial distribution of colors. The block diagram of the extraction process is given in Figure 2.5. Luminance chrominance color space is used with CLD as the color space descriptor. Using the Discrete Cosine Transform (DCT) over the tiny image which is extracted by partitioning the image into 8x8 blocks of average color values and zig-zag scanning the DCT coefficients, a compressed form of image is available for similarity matching. Main usage of CLD is high speed image retrieval and browsing, image matching and lay-out based retrieval such as sketch based image matching [24, 25, 27].

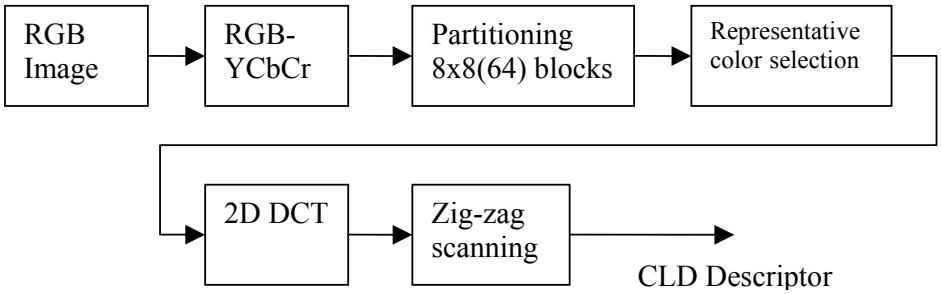

Figure 2.5 Block Schema of CLD extraction

Each color component is partitioned into 64 blocks. The icon is formed as a 8x8 image with pixel values are the average values of the blocks. 64 coefficient 2D DCT is applied. 6 coefficients for each icon are taken using zig-zag scanning the DCT output. The coefficients form the CLD. 3 coefficients for chroma icons could be taken as an alternative [26].

2.5.2 TEXTURE DESCRIPTORS

2.5.2.1 Homogeneous Texture Descriptor

Homogeneous Texture Descriptor is a quantitative representation consisting of the mean energy and the energy deviation from a set of frequency channels. 2D frequency domain is partitioned into 30 channels as shown in Figure 2.6 [25, 26].

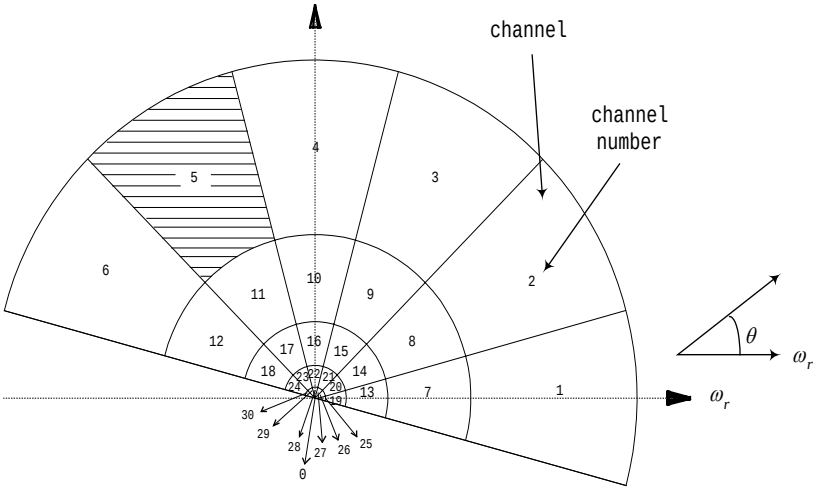

Figure 2.6 Frequency channels used in HTD

The mean energy and energy deviation are computed in each channel using

$$e_i = \log_{10}[1 + p_i] \quad i = 6s + r \tag{2.3}$$

and

$$d_i = \log_{10}[1 + q_i] \quad i = 6s + r \tag{2.4}$$

where p_i is computed using

$$p_i = \sum_{\omega=0^+}^{1} \sum_{\theta=(0^o)^+}^{360^o} \left[G_{s,r}(\omega,\theta)|\omega|P(\omega,\theta)\right]^2 \tag{2.5}$$

and q_i is computed using

$$q_i = \sum_{\omega=0^+}^{1} \sum_{\theta=(0^o)^+}^{360^o} \left[G_{s,r}(\omega,\theta)|\omega|P(\omega,\theta) - p_i\right]^2 \tag{2.6}$$

for s is in the range 0 to 4 and r is in the range 0 to 5.

$G(\omega,\theta)$ is the gabor wavelet in radial coordinates and $P(\omega,\theta)$ is the Fourier transform of the image in radial coordinates. Angular direction is divided uniformly into 6 channels and the radial direction is divided into 5 channels in octave scales. The implementation is done with the gabor wavelets using (2.7) [30]

$$G_{s,r}(\omega,\theta) = \exp\left[\frac{-(\omega-\omega_s)^2}{2\sigma_s^2}\right].\exp\left[\frac{-(\theta-\theta_r)^2}{2\tau_r^2}\right] \tag{2.7}$$

where τ_r and σ_s are calculated using

$$\tau_r = \frac{15^o}{\sqrt{2\ln 2}} \quad r \in \{0,1,2,3,4,5\} \tag{2.8}$$

$$\sigma_s = \frac{2^{-(s+1)}}{2\sqrt{2\ln 2}} \quad s \in \{0,1,2,3,4\} \tag{2.9}$$

An efficient method to compute the 2D Fourier transform of an image in radial coordinates is using the Radon transform formulated as

$$p_\theta(R) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(x\cos\theta + y\sin\theta - R)dxdy \tag{2.10}$$

and applying 1D Fourier transform. The projection slice theorem guarantees that the 1D Fourier transform of a projection of image at angle θ equals the slice at angle ω through the 2D Fourier transform of that image [4, 25, 31].

2.5.2.2 Texture Browsing Descriptor

Perceptual characterization of texture can be named as regularity, coarseness and directionality. Directionality is extracted using directional histograms computed by using the method in extracting homogeneous texture descriptor as discussed in Section 2.5.2.1 [33, 34]. For TBD computation angular direction is divided uniformly into 6 channels and the radial direction is divided into 4 in octave scales. The directional histogram is computed using

$$H(s,k)=\frac{N(s,k)}{\sum_{k=0}^{5}N(s,k)} \quad s=0,..,3 \quad \wedge \quad k=0,..,5 \tag{2.11}$$

where N(s,k) is the number of pixels in the filtered image at scale s and direction k whose magnitude is larger than a threshold t_s which is calculated as

$$t_s=\mu_s+\sigma_s \quad s=0,..,3 \tag{2.12}$$

where s denotes the scale and σ_s and μ_s are the mean and standard deviation over the filtered images at scale s [25, 26]. The dominant directions are indicated with the locations of peaks in the histogram. Sharpness is computed using

$$C(s,k)=0.5\cdot\left[H(s,k)-H(s,k-1)-H(s,k+1)\right] \tag{2.13}$$

Scales of the two peaks of highest sharpness are used to represent the dominant directions in the descriptor. On the dominant directions radon transform as given in (2.10) is applied. The normalized autocorrelation functions are computed using

$$NAC(k)=\frac{\sum_{m=k}^{N-1}P(m-k)P(m)}{\sqrt{\sum_{m=k}^{N-1}P^2(sm-k)\sum_{m=k}^{N-1}P^2(m)}} \tag{2.14}$$

where P is the projection after radon transform. The local peaks and valleys of normalized autocorrelation function are used to compute the contrast as

$$contrast=\frac{1}{M}\sum_{i=1}^{M}p\_magn(i)-\frac{1}{N}\sum_{j=1}^{N}v\_magn(j) \tag{2.15}$$

where p_magn and v_magn are the magnitudes at these peak and valley points. The scale index of maximum contrast is used to describe the coarseness for both dominant directions [26].

For the regularity estimate of the image texture the average of the distances among the successive peaks and standard deviation of distances are computed and named as dis and std. Candidate projections are determined by thresholding the std/dis computations of the projections with a typical value of 0.14. Credits are assigned using the observations: candidates are neighbors in scale orientation for regular textures and distribution of the candidates are sparse for not regular textures. The process is explained in the MPEG-7 Visual Experimentation Model document [26]. Regularity is between 1 and 4; 1 means irregular/unstructured texture and 4 means strong regular textures. Dominant directions, coarseness values and the regularity form the TBD [25].

2.5.2.3 Edge Histogram Descriptor

Edge histogram descriptor represents the spatial distribution of the edges in the image. Four directional edges (horizontal, vertical, 45º, 135º) and one non directional edge are searched in the 4x4 partitioned images. 80 bin histogram is computed to represent local edge distributions. Block scheme of the process is given in Figure 2.7. Edge Histogram (EH) is computed over the image blocks.

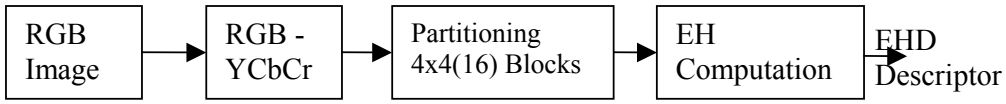

Figure 2.7 Block Schema of EHD extraction

The image is partitioned as shown in Figure 2.8.

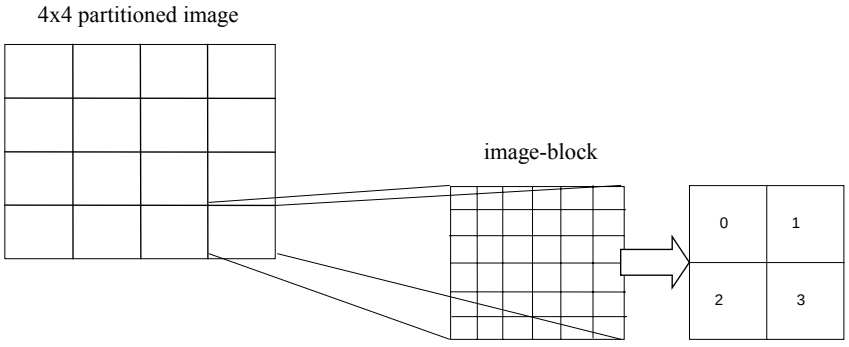

Figure 2.8 Definition of partitioning and image blocks

The image block is considered as a 2x2 super-pixel. Image blocks of 1100 seem to capture good directional edge features. The size of image block is regulated using

$$block\_size = \left\lfloor \frac{\sqrt{\frac{W \times H}{1100}}}{2} \right\rfloor \times 2 \tag{2.16}$$

where W and H are the dimensions of the image. The super pixel values are computed as the averages of corresponding pixels in the image block. Applying the filters provided in Figure 2.9, filtering results are compared with the threshold with default value 11 to populate the histogram [25, 26, 27, 35].

1	-1	1	1	$\sqrt{2}$	0	0	$\sqrt{2}$	2	-2
1	-1	-1	-1	0	$-\sqrt{2}$	$-\sqrt{2}$	0	-2	2
Vertical Edge Filter		Horizontal Edge Filter		Diagonal_45 Edge Filter		Diagonal_135 Edge Filter		Nondirectional Edge Filter	

Figure 2.9 Filter Coefficients for Directional Edge Descriptor

CHAPTER 3

OBTAINING IMAGE FEATURES FOR INDEXING AND RETRIEVAL

3.1 INTRODUCTION

In Content Based Image Indexing and Retrieval (CBIIR), indexing is performed by classifying the images. The classes are constituted to represent the semantics of the images so that images are retrieved from the appropriate class. In the classification process the image is the input to the classifier, and the output is the content of the image.

Using images as the input of the classifier leads to an excessive amount of calculations. A digital image is formed from pixels of data that involve correlations among the pixel values. This generally leads to a huge amount of data to process. Compressing images reduces the amount of data that is necessary to be processed, moreover the input can be optimized to be the feature vector from which indexing and retrieval can be made.

The feature vector derived from the image is used as the input in indexing and retrieval. It is required to form the feature vector to reflect the content of the image for a CBIIR structure with good performance. The content of an image can be thought of as a combination of color, texture and spatial properties in the image. The feature vector is derived from such primitive features of the image for content based image indexing and retrieval.

The extraction of feature vector from an image is discussed in Section 3.2. The feature vector is formed by combining color and texture features. The color and texture features are designed using color histogram correlations, detail histogram

correlations and directional filtering. These features are discussed in Sections 3.2.1, 3.2.2 and 3.2.3 consecutively.

3.2 FEATURE VECTOR

The block diagram of feature vector extraction is given in Figure 3.1

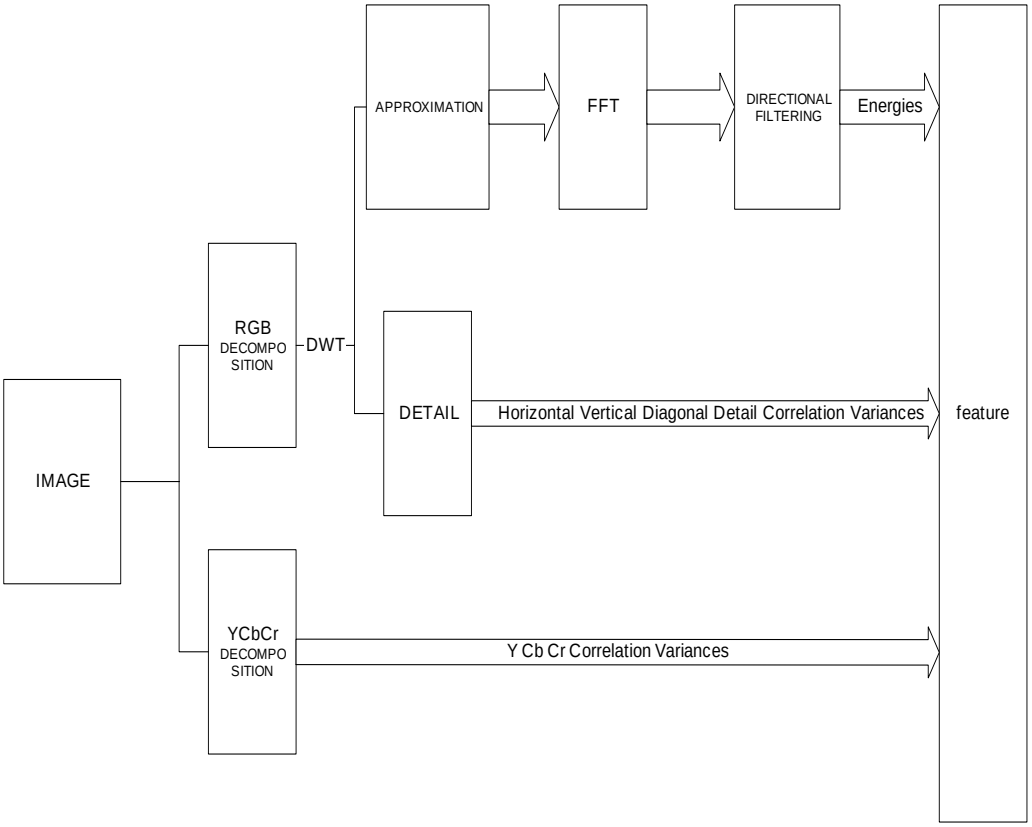

Figure 3.1 Block Diagram of Feature Vector Extraction

Color images of JPEG format at any sizes are input to the system. These images are decomposed into Red Green Blue (RGB) and Luminance Chrominance (YCbCr) images. To form the feature vector we tried to extract color, texture and spatial information from the images. In the CBIIR structure designed in this thesis the combination of these features is used.

A derived feature can be estimated from the primitive features of the image. For example in a “sunset” image red and orange are the dominant colors. In a “forest” image the dominant color is green and in a “sea” image the dominant color is blue. In an “autumn” image the dominant color is orange as well. To separate an

“autumn” image from a “sunset” image color information is not enough. Examining the texture in the images, there are periodic circular patterns in the frequency domain representation of “sunset” image and directional patterns in “autumn” image. The sun as a circle and the trees with leaves having horizontal, vertical and directional edges cause this difference. Using the monochrome “sunset” and “autumn” images it can be observed that the “sunset” image has the sun with circular pattern and the “autumn” image has the trees with directional patterns. This texture difference helps us to separate these images. Another example is the “sea” image and the “mountain” image. Both images have dominant color blue. These can be separated using the spatial location of the blue color. Using these observations it is clear that a combination of color, texture and spatial information can be used as the feature vector of the image.

Correlations of image histograms are used to extract the color information. YCbCr images are used to obtain the correlations between the image pixels and to derive features from these correlations. Luminance Chrominance representation is less sensitive to brightness whereas RGB representation is more sensitive [3]. Brightness is not a key feature in image semantics so that YCbCr representation is preferred in image correlations. The comparison of different color spaces is given in Section 5.3.

The texture information in the image is obtained using directional filtering and detail histogram correlations. 2D Discrete Wavelet Transform (DWT) is applied on RGB images. This transform outputs the down sampled low pass filtered signal named approximation, and down sampled high pass filtered signal named the detail, which is mentioned in Section 3.2.2. The detail images provide the edge information in horizontal, vertical and diagonal directions. Correlations among these detail images are used to obtain the edge information which is mentioned in Section 3.2.2. The approximation can be taken as the compressed form of the image by the down sampling process. Using the approximation decreases the computation while negligibly decreasing performance. The approximation is transformed using 2D Discrete Fourier Transform (DFT) that reveals the directions of the edges and the texture information when inspected in frequency channels. This is discussed in Section 3.2.3.

3.2.1 Color Histogram Correlations

Color images are formed from three sub images Red, Green and Blue in the additive color system. This is RGB representation, however there are alternatives to this representation such as YCbCr (luminance, chrominance), YIQ (Luminance, Intensity, and Saturation which is also used in NTSC TV systems) or HSV (Hue, Saturation, and Value) [4].

The relationship between RGB and YIQ is shown as

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} +0.299 & +0.587 & +0.114 \\ +0.596 & -0.274 & -0.322 \\ +0.211 & -0.523 & +0.312 \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{3.1}$$

In YCbCr representation, chrominance is obtained using

$$\begin{aligned} Cb &= R - Y \\ Cr &= B - Y \end{aligned} \tag{3.2}$$

In HSV representation, hue refers to the color, saturation refers to the amount of pure hue and value refers to brightness. These color spaces are reviewed in Section 2.5.1.1. Since brightness is not a key feature in CBIIR, luminance chrominance decompositions are preferable. In Section 5.3 the comparison of correlation features with different color space decompositions are given. The results are in agreement with those obtained by using luminance chrominance decomposition.

As mentioned in section 3.2, content of an image has relations with its color information. Color information from the image is obtained using the relationship between the color distribution and the content. Color distributions are studied with the color histogram of the images.

The histogram of an image is a way to represent the color level densities. It gives the number of pixels in each color level used in the color format. For example there are 256 levels for an eight bit representation monochrome image. For a 24 bit color image, in luminance chrominance decomposition there are 3 histograms of 256 bins for luminance and chrominance images Y, Cb, Cr.

Histogram of each color sub image is computed and the area under the histogram is normalized to eliminate the effects of image size. For convenience in feature extraction the normalized and scaled histogram is derived as

$$pdf(X) = (hist(X) / sum(hist(X))) * 10 \tag{3.3}$$

where pdf denotes the normalized and scaled histogram and X is the image.

The autocorrelation and cross-correlation of sub image histograms are used to investigate the relationship between the color information and the semantics. For YCbCr decomposition pdf(X) is calculated for Y, Cb and Cr sub images. Autocorrelations of the color histograms are calculated. The cross-correlations between these histograms: between Y and Cb, between Y and Cr, and between Cb and Cr, are calculated.

The true cross-correlation sequence is a statistical quantity defined as

$$R_{xy}(m) = E\{x_{n+m} y^*_n\} = E\{x_n y^*_{n-m}\} \tag{3.4}$$

where x and y are stationary random processes, and $E\{\}$ is the expected value operator. In practice, these sequences should be estimated. A common estimate based on N samples of x and y is the deterministic cross-correlation sequence which is defined as

$$\hat{R}_{xy}(m) = \begin{cases} \sum_{n=0}^{N-m-1} x_{n+m} y_n^* & 0 \le m \le (N-1) \\ \hat{R}_{xy}^*(-m) & (1-N) \le m < 0 \end{cases} \tag{3.5}$$

assuming x_n and y_n are indexed from 0 to N-1 and N is a positive integer greater than two.

In Figure 3.2, an example image of a building with blue, green and light brown color composition is given. Calculations using equations (3.3), (3.4) and (3.5) are performed using MATLAB® on the image in RGB and YCbCr decomposition. The outputs of RGB decomposition are given in Figure 3.3 and the outputs of YCbCr decomposition are given in Figure 3.4. As seen from the figures, the RGB histograms show a wider spread over the range of color levels whereas YCbCr histograms show less spread through the plot. The characteristics of YCbCr histogram correlations may be used to form a descriptor which is more compact in size.

RGB decomposition does not represent the best perception of color. HSV and luminance chrominance decompositions are used in retrieval processes since they represent the color perception better [1, 3]. The tests yielded good results with YCbCr space as expected. The test results are given in Section 5.3

Figure 3.2 Example Image

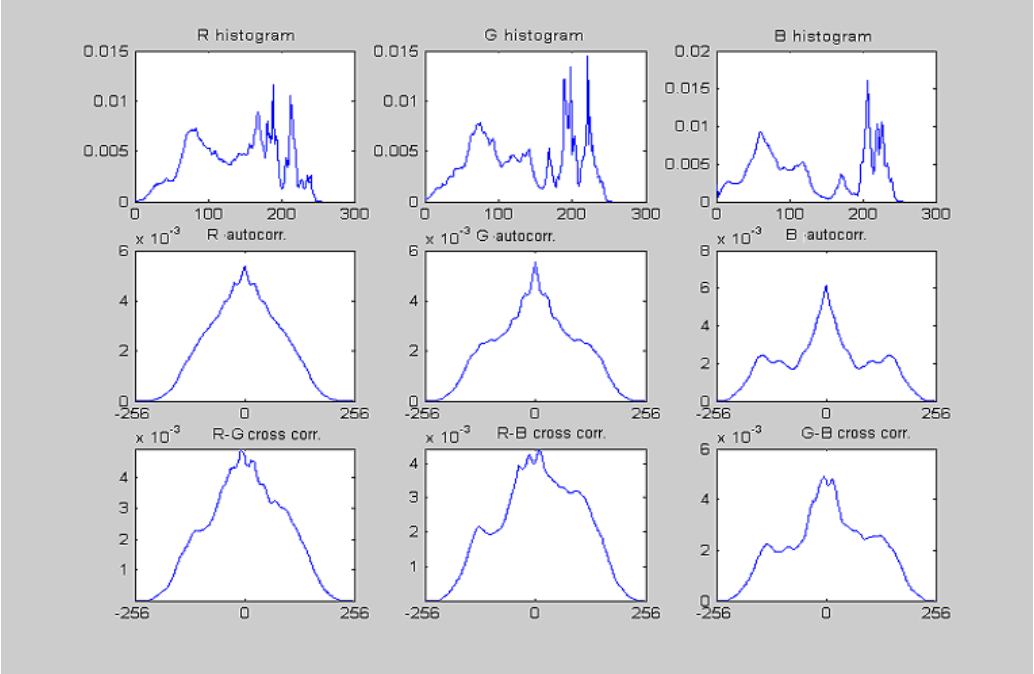

Figure 3.3 RGB Decomposition Histograms and Correlation Plots of Example Image given in Figure 3.2

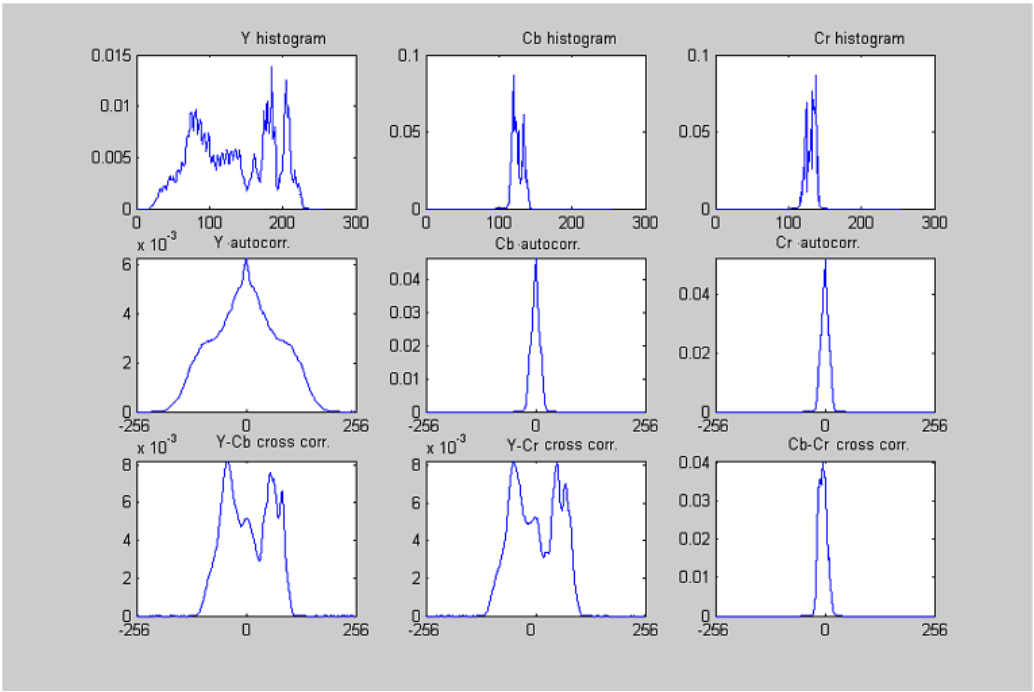

Figure 3.4 YCbCr Decomposition Histograms and Correlation Plots of Example Image given in Figure 3.2

The correlation plots in Figure 3.3 and Figure 3.4 indicate that the YCbCr representation has specific correlation properties such as the distribution over color level axis that can be used in obtaining features for CBIIR.

In Figure 3.5, Figure 3.7, Figure 3.9, Figure 3.11, Figure 3.13, and Figure 3.15 example images from the used database is given to investigate the color histogram correlation properties. Different color properties in images can be seen in the histograms. The color histogram and color histogram correlation curves that are derived from these images using (3.3) and (3.5) are given in Figure 3.6, Figure 3.8, Figure 3.10, Figure 3.12, Figure 3.14, and Figure 3.16 respectively. Differences can be observed using correlation curves as it is observed from the figures.

Correlation graphs provide information about color distribution of the images, so that they could be used as color features for content based indexing. Examining these images and their correlation curves, it's seen that instead of the whole curves, features from the curves such as moments and moment positions on the color level axis can be used. The moment positions and moments used are discussed in this section and calculated values are given as a conclusive comparison.

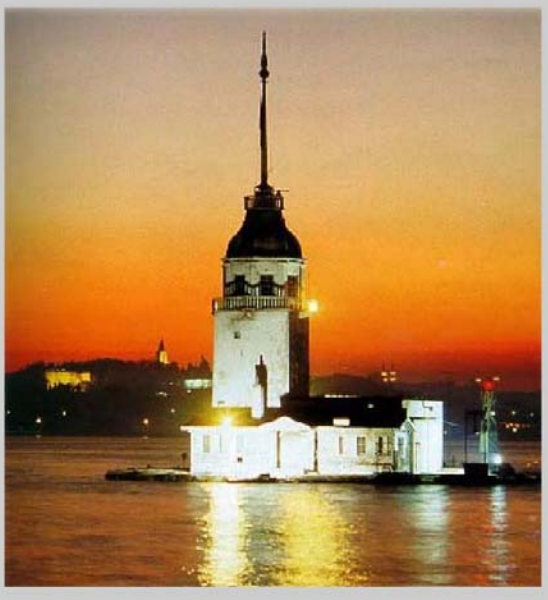

Figure 3.5 Example Image 1

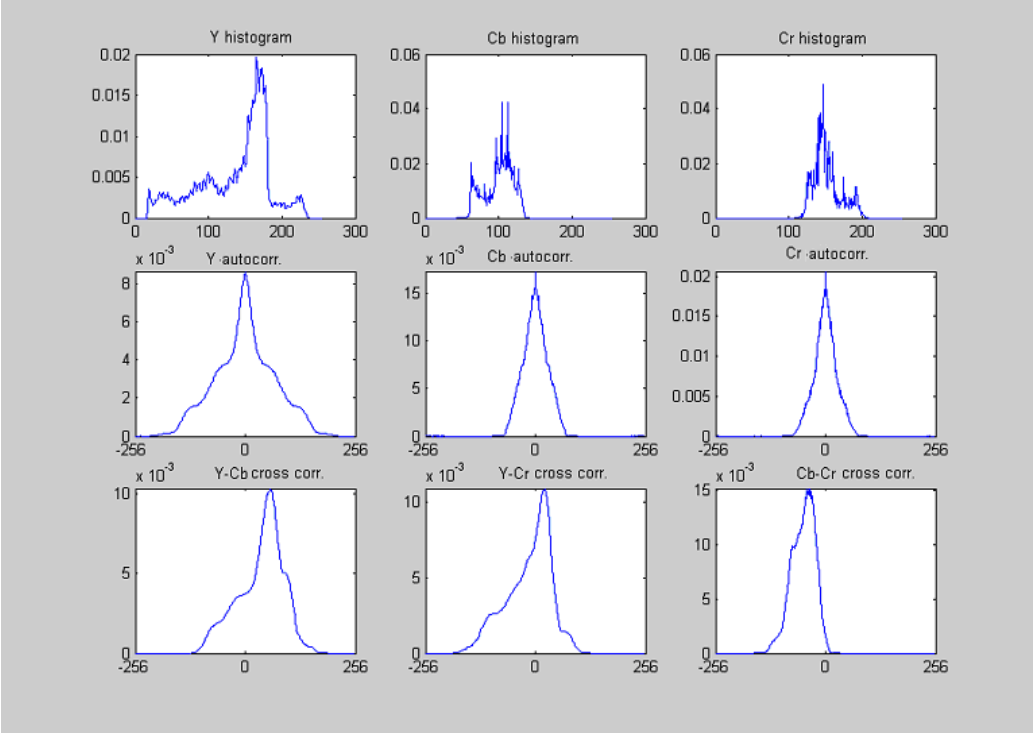

Figure 3.6 Histograms and Correlation Outputs of the Image at Figure 3.5

Figure 3.7 Example Image 2

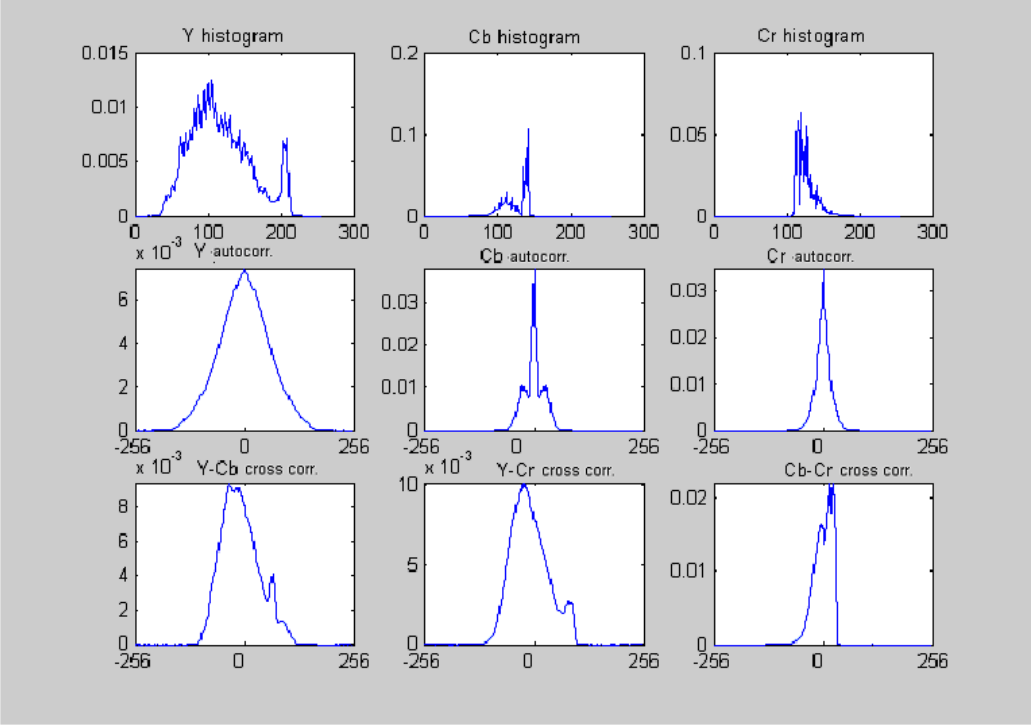

Figure 3.8 Histograms and Correlation Outputs of the Image at Figure 3.7

Figure 3.9 Example Image 3

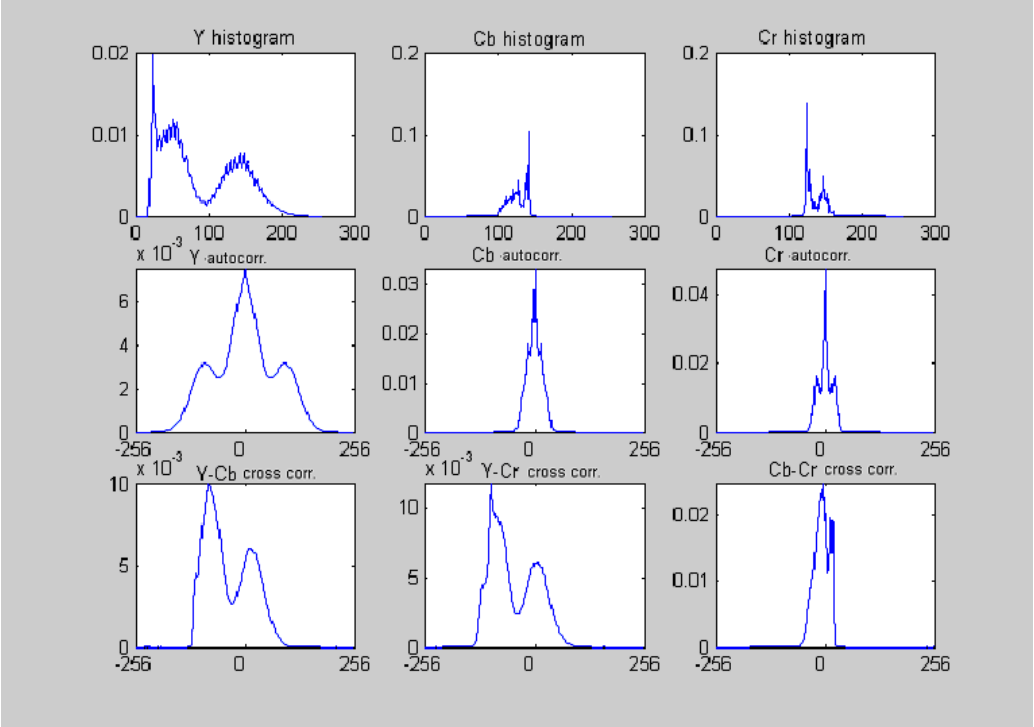

Figure 3.10 Histograms and Correlation Outputs of the Image at Figure 3.9

Figure 3.11 Example Image 4

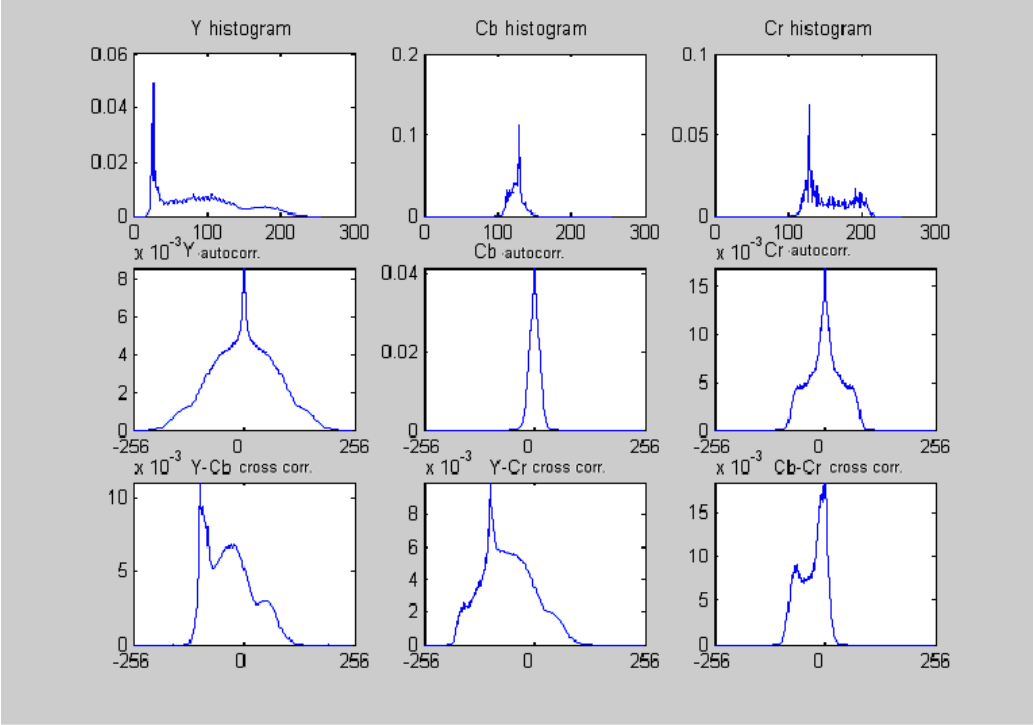

Figure 3.12 Histograms and Correlation Outputs of the Image at Figure 3.11

Figure 3.13 Example Image 5

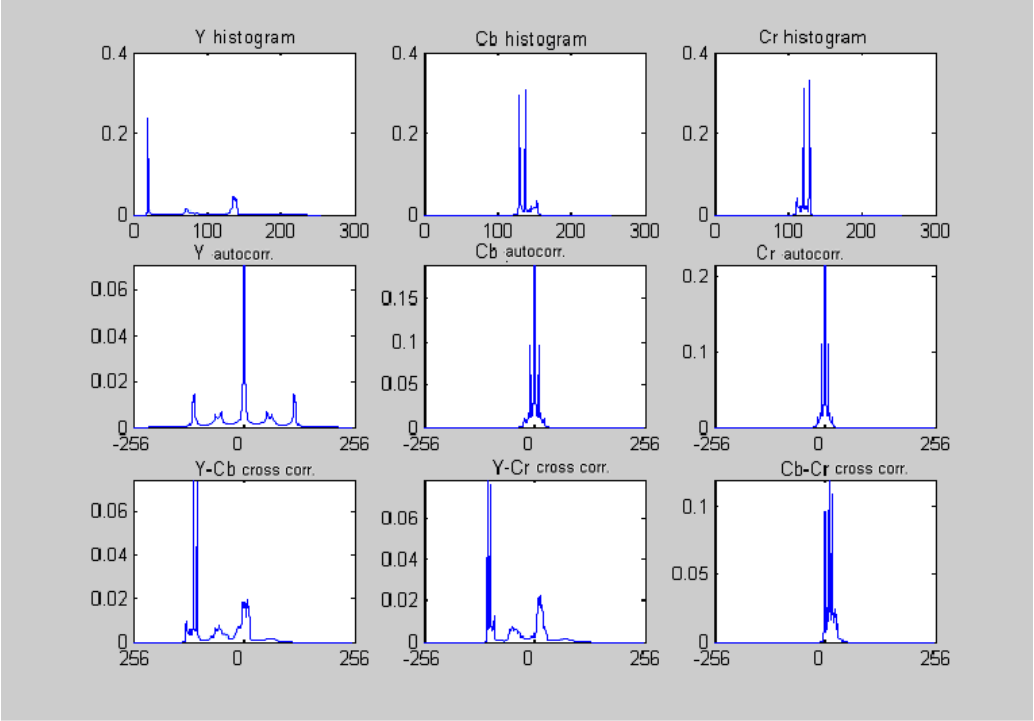

Figure 3.14 Histograms and Correlation Outputs of the Image at Figure 3.13

Figure 3.15 Example Image 6

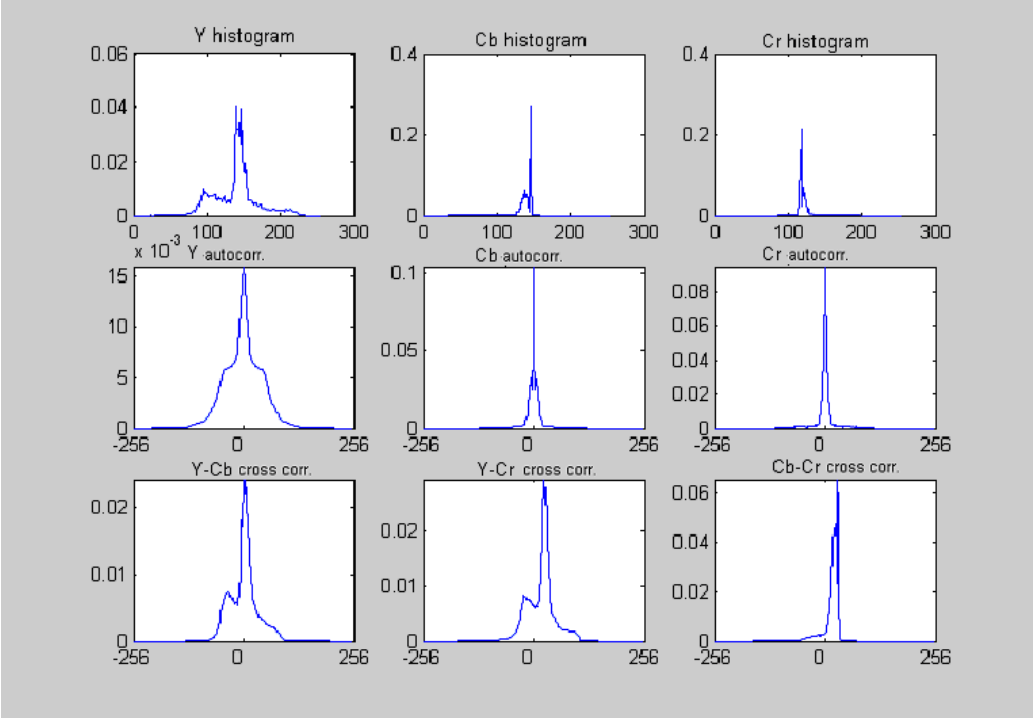

Figure 3.16 Histograms and Correlation Outputs of the Image at Figure 3.15

Variance and energy distribution are the characteristics derived from the correlation curves of the images. The maximum and the mean value positions are the characteristics derived from the cross correlation curves for each image.

The variance of the correlation vector is computed as

$$\sigma_x^2 = \sum_x ((x-\bar{x})^* \bullet (x-\bar{x})^T)/(m-1) \tag{3.6}$$

where σ^2_x is the variance, x is the correlation vector, and m is the length of x. The variance and mean of the curve are important features for the curve. The mean and maximum value positions of the autocorrelation curves are at the origin and their values do not yield a feature for CBIIR.

The variances for the autocorrelation and cross correlation is used to obtain a set of 6 features considering the color sub images. The variance features for example images given in Figure 3.5, Figure 3.7, Figure 3.9, Figure 3.11, Figure 3.13 and Figure 3.15 are given in Table 3.1 for the color spaces referenced in the heading column.

Table 3.1 Correlation variance features

Image	**Figure 3.5**	**Figure 3.7**	**Figure 3.9**	**Figure 3.11**	**Figure 3.13**	**Figure 3.15**
Y Autocorrelation	0.0454	0.0574	0.0397	0.0350	0.1750	0.1063
Cb Autocorrelation	0.1592	0.2391	0.2738	0.4266	1.2270	0.6037
Cr Autocorrelation	0.1816	0.3181	0.3323	0.1029	1.5706	0.7933
Y-Cb Cross-correlation	0.0761	0.0898	0.0813	0.0732	0.3175	0.1882
Y-Cr Cross-correlation	0.0802	0.0950	0.0841	0.0542	0.3653	0.2051
Cb-Cr Cross-correlation	0.1687	0.2540	0.2930	0.1683	1.0850	0.6303

The 90% of the area under the correlation curves reflects the energy distribution of the curve. This distribution is found by integrating the curve and taking the interval between 5% and 95% points in the integration output. The ratio of this interval to the total number of points in the curve is used as the energy distribution feature of the curve. The results formed a set of 6 features from autocorrelations and cross correlations of the color sub images. In Table 3.2 the set of these energy distribution features are given for the example images referenced in the heading row and the color spaces referenced in heading column.

Table 3.2 Correlation characteristics from 90% energy distribution

Image	**Figure 3.5**	**Figure 3.7**	**Figure 3.9**	**Figure 3.11**	**Figure 3.13**	**Figure 3.15**
Y Autocorrelation	0.4736	0.3796	0.4736	0.4932	0.4736	0.2661
Cb Autocorrelation	0.1840	0.1487	0.1135	0.0783	0.0744	0.0822
Cr Autocorrelation	0.1761	0.1096	0.1018	0.2661	0.0509	0.1018
Y-Cb Cross-correlation	0.3464	0.2838	0.3112	0.3327	0.3366	0.2153
Y-Cr Cross-correlation	0.3346	0.2877	0.3092	0.3992	0.3327	0.2153
Cb-Cr Cross-correlation	0.1683	0.1233	0.1096	0.1761	0.0587	0.1037

The mean and the maximum of the correlation curves yield the same values that provide no indexing feature. The mean and maximum points of autocorrelation curve are obtained at the origin of the correlation curves. The points where the mean and maximum values of cross correlation curves are obtained provides to be suitable to be used as features in content based indexing. Color-mean is used to represent the color level where the mean occurs. Color-maximum is used to represent the color level where the maximum occurs. These points are used to calculate the measure of similarity of cross correlation curves that could provide features for classification,

since cross correlation of color histograms reflects the correlation between the perceived colors.

The results are taken as a set of 6 features from cross correlations of the color sub images. In Table 3.3 the set of color-mean features is given for the example images referenced in the heading row and the color spaces referenced in heading column. The color-maximum features are given in Table 3.4 for the example images referenced in the heading row and color spaces referenced in heading column.

Table 3.3 Color-mean features

Image	**Figure 3.5**	**Figure 3.7**	**Figure 3.9**	**Figure 3.11**	**Figure 3.13**	**Figure 3.15**
Y-Cb Cross-correlation	0.5930	0.4755	0.3973	0.4286	0.4149	0.5049
Y-Cr Cross-correlation	0.4932	0.4736	0.3777	0.3699	0.4384	0.5440
Cb-Cr Cross-correlation	0.4070	0.5049	0.4834	0.4540	0.5225	0.5421

Table 3.4 Color-maximum features

Image	**Figure 3.5**	**Figure 3.7**	**Figure 3.9**	**Figure 3.11**	**Figure 3.13**	**Figure 3.15**
Y-Cb Cross-correlation	0.6164	0.4305	0.3346	0.2994	0.2681	0.5049
Y-Cr Cross-correlation	0.5362	0.4599	0.3033	0.3014	0.2857	0.5479
Cb-Cr Cross-correlation	0.4325	0.5303	0.4932	0.5029	0.5186	0.5558

From the plots and tables provided, it is seen that moment distributions of the correlation curves and second moment of cross-correlation curves can be used as color features for content based image indexing. A total of 18 features are obtained from the Correlations: 6 from variances of all correlations, 6 from mean and peak positions of cross correlations and 6 from the distribution of 90% energy of the correlation curves.

3.2.2 DWT and Detail Histogram Correlations

Wavelet Transform (WT) is a signal analysis tool that has very attractive properties and applications. WT makes it possible to analyze non stationary time varying signals in time and frequency domains by using scaled and shifted wavelets. A Wavelet is a finite duration signal of zero average. For example: a sinus of finite

duration, ‘Haar Wavelet’ or ‘Symlet Wavelet’. ’Symlet Wavelet’ and ‘Haar Wavelet’ illustrations are given in Figure 3.17.

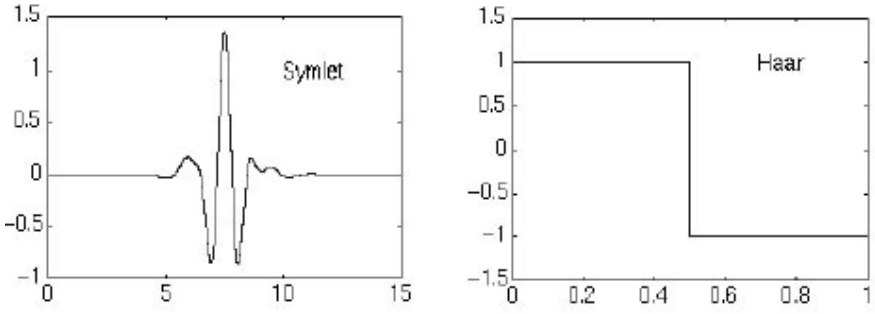

Figure 3.17 Illustrations of ‘Symlet’ and ‘Haar’ Wavelets

The projections of the input signal on scaled and shifted wavelets are used in analyzing signal in time-frequency domain. This is called Continuous Wavelet Transform, because operations are performed on continuous signals with continuous wavelets. Redundant information in the time-frequency plane that is placed by the Continuous Wavelet Transform is overcome by discretizing and using algorithms equivalent to two-channel filter banks [6]. Discrete Wavelet Transform is such an algorithm. Filter Banks that are formed from low pass and high pass filters are used. There is a relationship between the wavelets and the filters in DWT such that the wavelet is determined with the shape of the reconstruction filters.

The wavelet function is determined by the high pass filter, which produces the details of the wavelet decomposition. The low pass filter is determined with the scaling function. Iteratively up sampling and convolving the high pass filter produces a shape approximating the wavelet function while iteratively up sampling and convolving the low pass filter produces a shape approximating the scaling function. In Figure 3.18 ‘Daubechies 4’ wavelet, its scaling function and its filter equivalents are given [7].

In one dimensional Discrete Wavelet Transform (1D DWT), low pass filter output is the approximation, high pass filter output is the detail. Images are two dimensional signals; therefore two dimensional Discrete Wavelet Transform is used. The filter bank implementation of 2D DWT is given in Figure 3.19 [7].

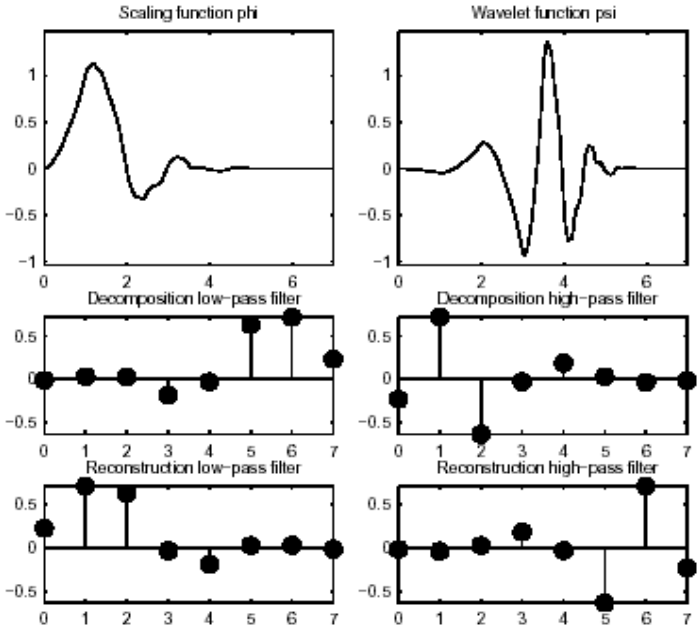

Figure 3.18 'Db4' Wavelet, Corresponding Scaling Function and Filters

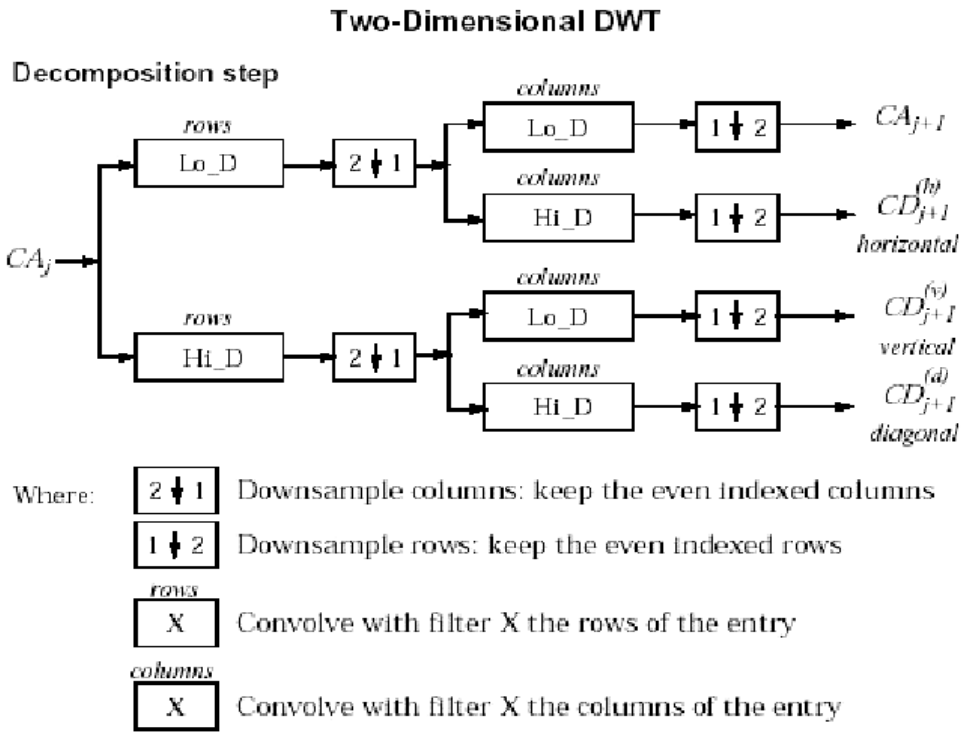

Figure 3.19 Block Diagram of Two Dimensional DWT

Down sampling process involved in DWT results in reduction of the size, that is at appropriate levels the signal is compressed. Level is the number of DWT applied to the signal. After the first transformation, the approximation can be transformed further. The size reduction in the form of approximation appears to be useful in our study as explained in Section 3.2.3, by decreasing the amount of data to be processed.

2D DWT is taken using a filter bank that involves low pass (LP) and high pass (HP) filters on two dimensions. Approximation is the output of LP filter on rows and columns, horizontal detail is the output of LP filter on rows and HP filter on columns, vertical detail is the output of HP filter on rows and LP filter on columns as well as diagonal detail is the output of HP filter on both rows and columns. An illustration is given in Figure 3.20.

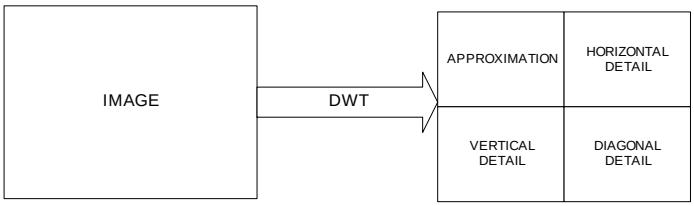

Figure 3.20 2D DWT Illustration

2D DWT is applied on the R, G and B decompositions of the image in Figure 3.2 by transforming each color sub image. The result is given in Figure 3.22 using the 'Daubechies 3' ('Db3') wavelet. An illustration of 'DB3' wavelet is given is in Figure 3.21.

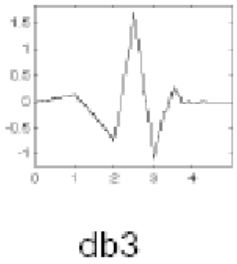

Figure 3.21 Illustration of 'Daubechies3' Wavelet

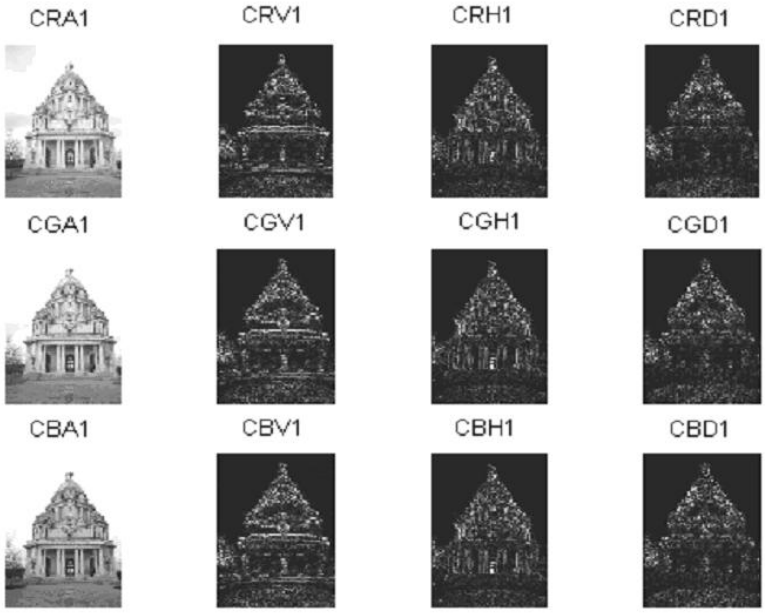

Figure 3.22 Example of the One Level DWT on R G B Images

The details carry high frequency content resulting from the HP filtering involved. The high frequency content in images corresponds to the edge information. The detail information can be used to fetch the edge information in the images, which is used as features for content based indexing.

'Db 3' wavelet from the Daubechies Wavelet family is used because of its sharp frequency properties in the filter implementations. Therefore it is possible to make comparisons on low pass and high pass filtered outputs of images [3]. That aids in estimating edge densities from the detail since edges are well defined.

In Figure 3.23, Figure 3.24, Figure 3.25, Figure 3.26, Figure 3.27, and Figure 3.28 the calculation of histograms and correlation curves of details of DWT of luminance decomposition of images in Figure 3.5, Figure 3.7, Figure 3.9, Figure 3.11, Figure 3.13, and Figure 3.15 respectively, is given. The detail histograms are computed from the coefficient matrices of DWT output of the luminance images. The histogram contains the information of densities of coefficients. Replacing color spaces R, G, B to CH, CV and CD the detail histogram correlations are calculated using (3.3) and (3.5), where CH means horizontal detail coefficients, CV means vertical detail coefficients and CD means diagonal detail coefficients.

The variance and energy distribution of cross correlation curves of detail images are calculated as the detail correlation features. The coefficient distribution has similar properties such as the high density at low coefficient levels, which can seen in the Figure 3.22 as the black areas in the detail images. Because of this and the normalization process, only the variances of the correlation curves and the energy distribution of the correlation curves are found to be features for edge information. Those components are given in Table 3.5 and Table 3.6**,** from which it can be seen that these can be used in classifying images for content based indexing.

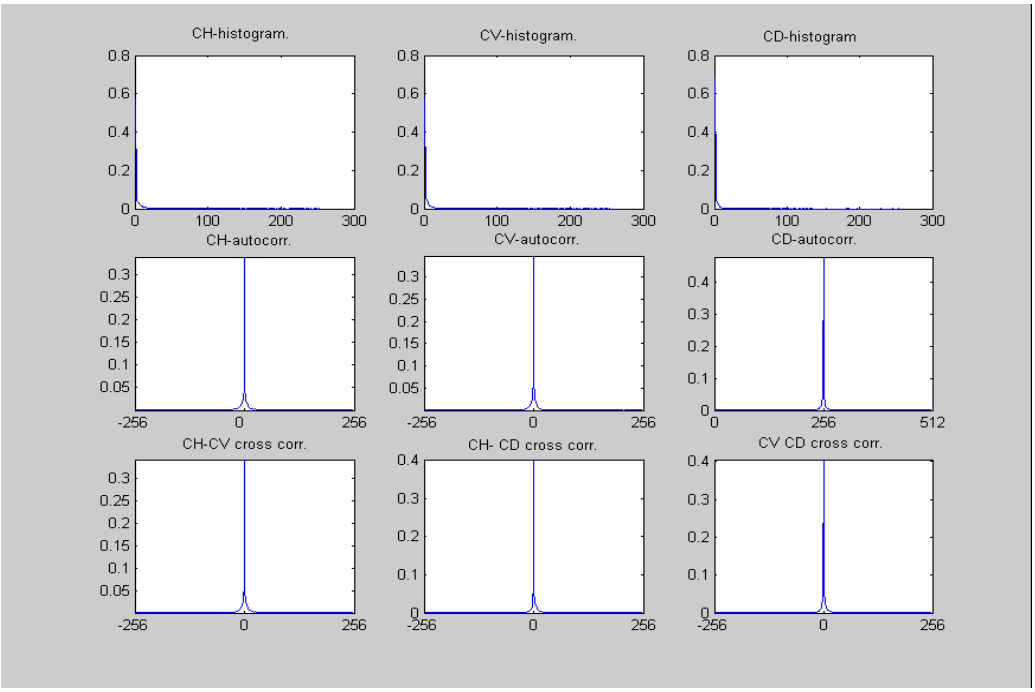

Figure 3.23 Histograms and Correlation Plots of Details of Level 1 DWT of the Image of Figure 3.5

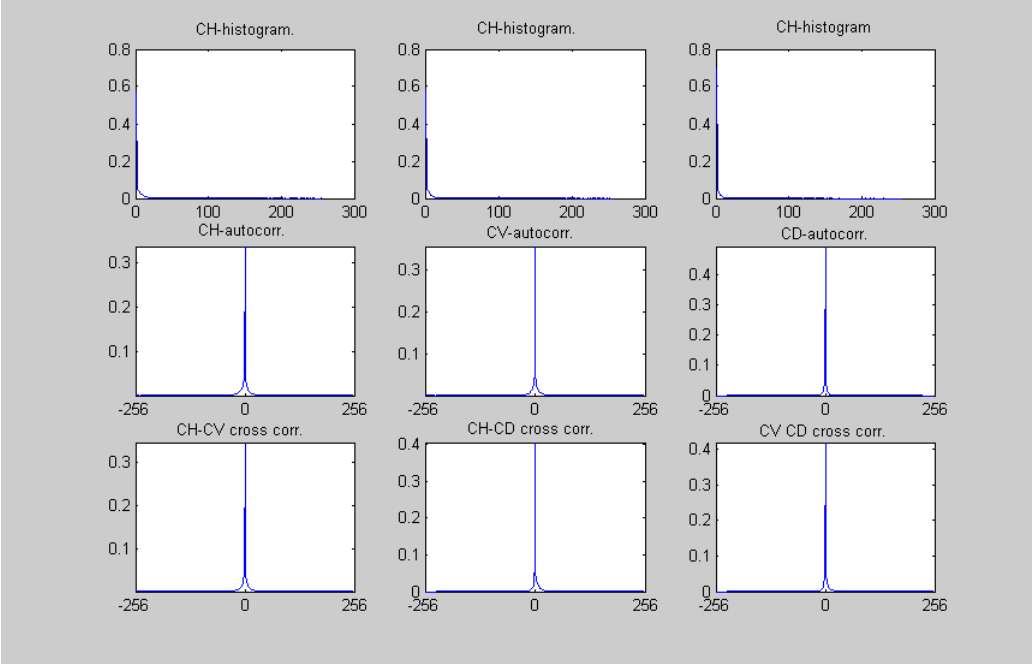

Figure 3.24 Histograms and Correlation Plots of Details of Level 1 DWT of the Image of Figure 3.7

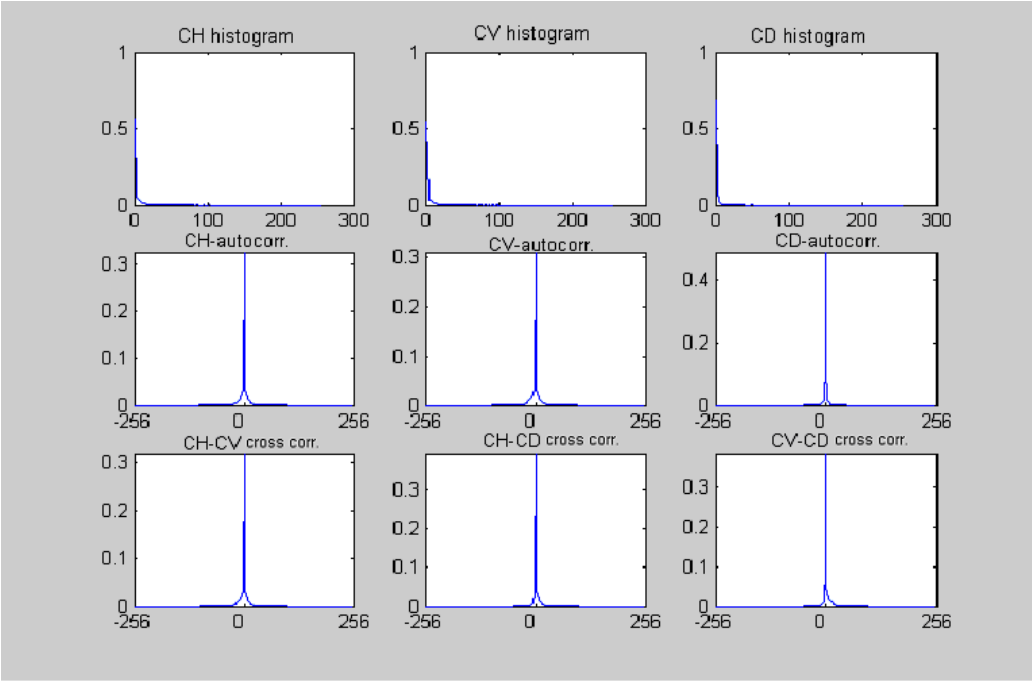

Figure 3.25 Histograms and Correlation Plots of Details of Level 1 DWT of the Image of Figure 3.9

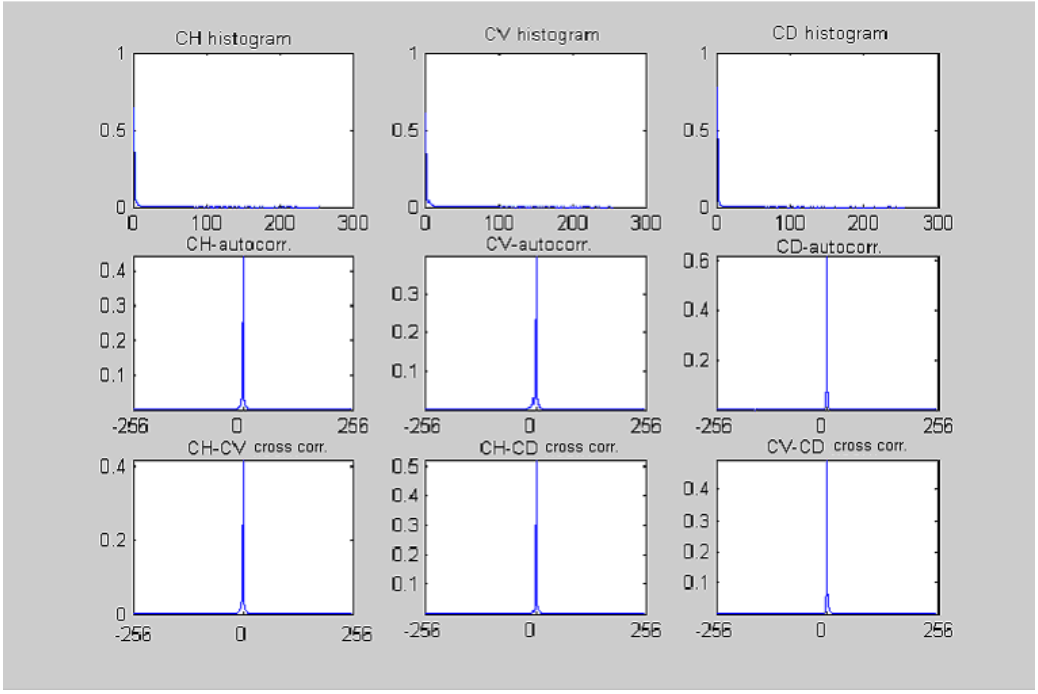

Figure 3.26 Histograms and Correlation Plots of Details of Level 1 DWT of the Image of Figure 3.11

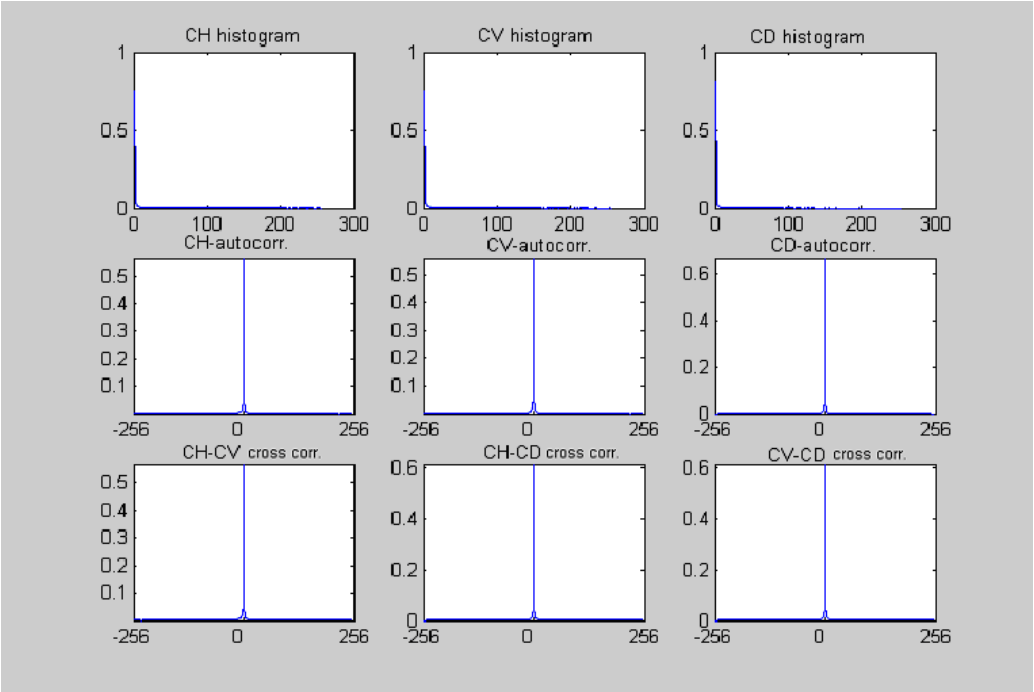

Figure 3.27 Histograms and Correlation Plots of Details of Level 1 DWT of the Image of Figure 3.13

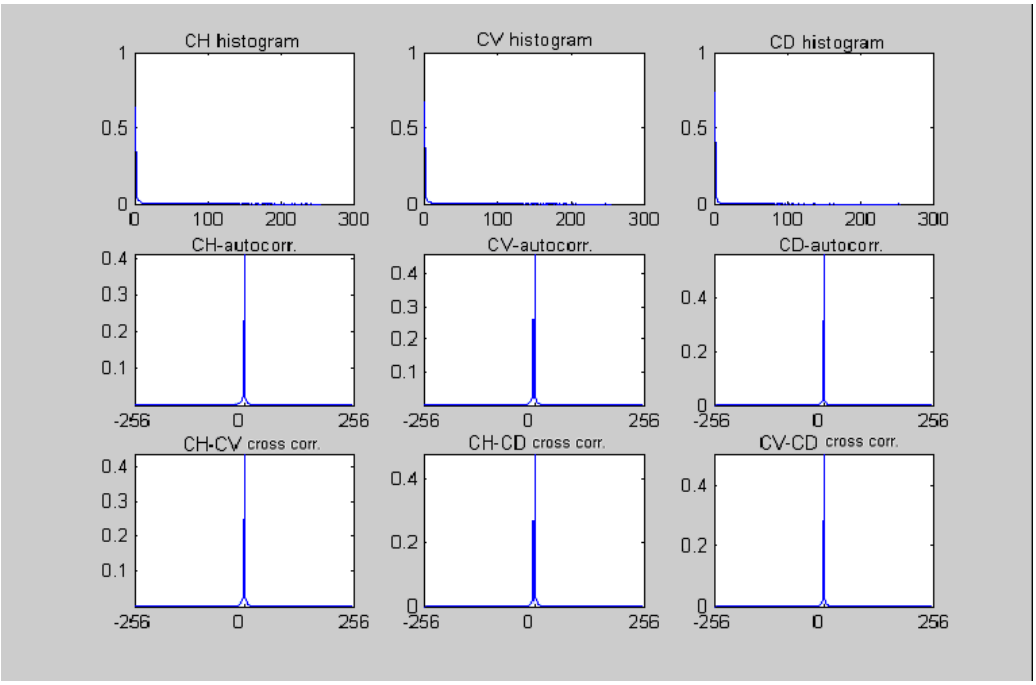

Figure 3.28 Histograms and Correlation Plots of Details of Level 1 DWT of the Image of Figure 3.15

Table 3.5 DWT detail correlation characteristics from variances

Image	**Figure 3.5**	**Figure 3.7**	**Figure 3.9**	**Figure 3.11**	**Figure 3.13**	**Figure 3.15**
CH Autocorrelation	0.2444	0.2408	0.2323	0.4142	0.6418	0.3533
CV Autocorrelation	0.2609	0.2694	0.2107	0.3470	0.6346	0.4418
CD Autocorrelation	0.4799	0.5109	0.5132	0.7914	0.8867	0.6451
CH-CV Cross-correlation	0.2520	0.2544	0.2212	0.3788	0.6379	0.3948
CH-CD Cross-correlation	0.3405	0.3473	0.3394	0.5681	0.7541	0.4760
CV-CD Cross-correlation	0.3518	0.3691	0.3223	0.5176	0.7495	0.5331

Table 3.6 DWT detail correlation characteristics from 90% energy distribution

Image	**Figure 3.5**	**Figure 3.7**	**Figure 3.9**	**Figure 3.11**	**Figure 3.13**	**Figure 3.15**
CH Autocorrelation	0.1409	0.0861	0.0626	0.0352	0.0900	0.0900
CV Autocorrelation	0.0939	0.1096	0.0665	0.0391	0.0587	0.0626
CD Autocorrelation	0.0391	0.0352	0.0196	0.0117	0.0313	0.0313
CH-CV Cross-correlation	0.1174	0.0978	0.0646	0.0372	0.0724	0.0783
CH-CD Cross-correlation	0.0939	0.0626	0.0411	0.0235	0.0607	0.0626
CV-CD Cross-correlation	0.0665	0.0744	0.0450	0.0254	0.0431	0.0489

The results that are given in Table 3.5 and Table 3.6 are for one level DWT where the used wavelet is ‘db3’. When 3 level DWT is used; for each level, a set of 12 features are formed. In Section 5.3, the discussed test results give that 3 level DWT results better in agreement with the resolution of the detail proposed. As the multiresolution level is increased, more edge information is supplied. DWT detail histogram correlations are efficient in content based indexing because of the edge information supplied.

3.2.3 Directional Filtering on Approximation

Color histogram correlation features and detail histogram correlation features used in the feature vector were discussed in Sections 3.2.1 and 3.2.2. Section 3.2 claims that the directions of the edges such as circular edges, or sloped curves are important in human vision for perception of objects. Fourier Transform (DFT) provides such information. An edge in the image results in rapid changes in the color values, therefore the values corresponding to an edge can be localized in the frequency domain as high energy parts. This property of DFT is used to get the edge information from the image and to form features to be used in CBIIR.

In Figure 3.29 the approximation output of an image’s Red color decomposition and its DFT output are given. The high energy parts in DFT output resulting from the edges of mountains can be seen as the sloped whiter curves in the frequency domain representation.

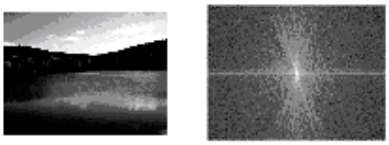

Figure 3.29 Example Approximation Image before and after DFT

To observe this property clearly detail image of the same image and DFT output of the detail image are given in Figure 3.30. The edges can be localized to at some frequency channels.

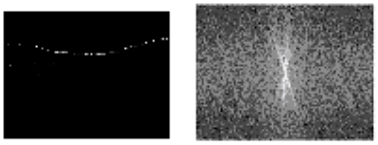

Figure 3.30 Example Detail Image before and after DFT

In Figure 3.31, the approximation image of DWT output of the blue color decomposition of the image at Figure 3.2 and its DFT output is given to show the high energy parts in DFT resulting from the edges of roofs. Those are the 45^0C lines resulting from the edges of the building's roof.

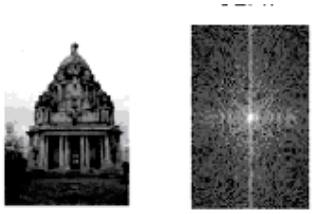

Figure 3.31 Example Image 1 before and after FFT

In Figure 3.32 DFT of approximation images of DWT output of an image's Red, Green and Blue color decompositions and their DFT outputs are given to show the effects from circular objects. In the frequency domain circular patterns are observed. In different color decompositions the densities of circular patterns differ because of the color composition of the image.

These observations show that the characteristics of edges or circular objects in the image can be sampled in the DFT output of the image. To characterize and extract features from these properties, the energy densities at frequency channels can be used on color decompositions of the image. The frequency domain is uniformly divided into frequency channels of equal angles. The frequency channels are implemented as DC stop filters in the frequency domain. The normalized energy densities at filtering outputs are used as features to represent the edge information.

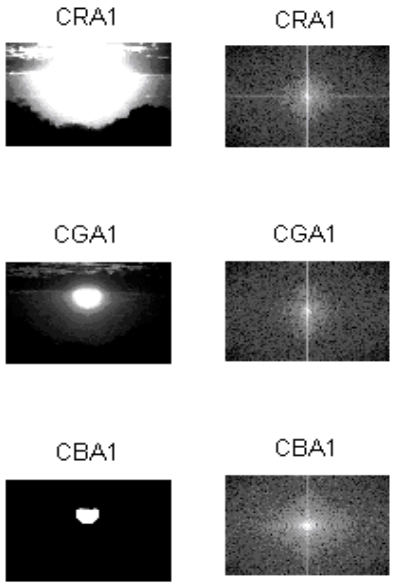

Figure 3.32 CA before and after DWT

Using the fact that DFT of real images is symmetric the filter outputs in quadrant three and four are sufficient for computations. Examples of used filters implemented in Quadrant 3 and Quadrant 4 with 16 frequency channels are given in Figure 3.33.

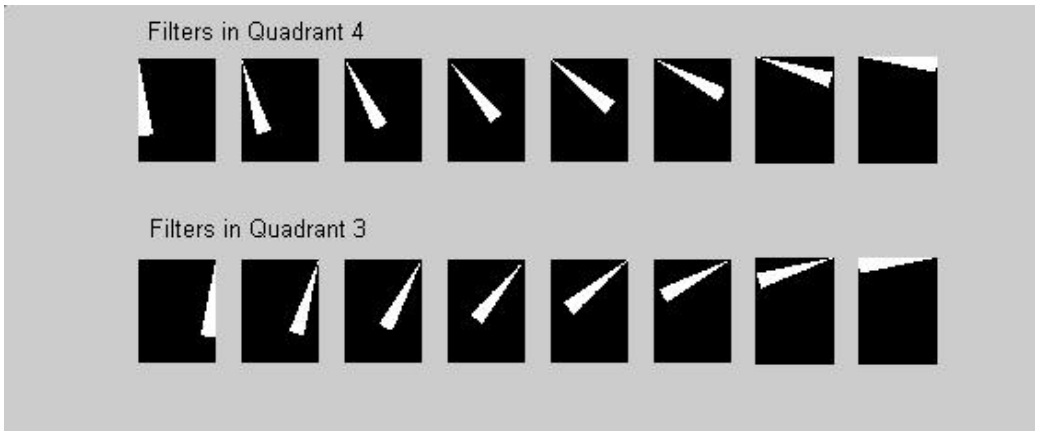

Figure 3.33 Used Filters in Frequency Domain

Different numbers of frequency channels are tried and the test results give best performance with 7 channels for quadrant. The results are given in Section 5.2. The feature extraction is discussed next.

For ith channel the filter output o_i is obtained using

$$o_i = I \cdot fc_i \tag{3.7}$$

where I is the DFT output of the image and fc_i is the filter for ith channel. The energy output of ith channel is computed using

$$E_i = \sum_x \sum_y o_i^2[x, y] \tag{3.8}$$

where x and y denotes the dimensions one quadrant of the image in the frequency domain. The energy output for ith channel is normalized with k which is obtained using

$$k_i = \sum_x \sum_y fc_i^2[x, y] \tag{3.9}$$

to eliminate the effects of the image size. The energy feature is obtained by scaling the normalized energy using

$$e_i = \log\left[1 + \left|\frac{E_i}{k_i}\right|\right] \tag{3.10}$$

This process is computed for each color space decomposition. Directional Filtering feature of length 42 (14x3) is derived. To decrease the computational load features are implemented on the approximation outputs of DWT of the image. The test results on this feature are given in Section 5.3. The feature vectors for the example images given in Figure 3.7, Figure 3.9, Figure 3.11, Figure 3.13, and Figure 3.15 are given in Appendix.

CHAPTER 4

CONTENT BASED IMAGE INDEXING AND RETRIEVAL USING ARTIFICIAL NEURAL NETWORKS

4.1 INTRODUCTION

Image Retrieval in digital databases is performed based on a query. The images matching with the query are retrieved. The queries vary in the sense of the retrieval system. For Content Based Image Indexing and Retrieval query is the content of the image. This requires the indexing of the images. There are also retrieval systems based on the query of primary features such as a desired color, texture, shape or spatial location in the image. A common type of primitive feature retrieval is retrieving images that are similar to the query image.

In Content Based Image Indexing and Retrieval, the index reflects the content of the image. Indexing is obtained by image classification. The classes represent the semantics of the images. The database is searched to index the images. Hence the indexed image database can be used for CBIIR. Indexing also enhances primitive feature retrieval by searching only the valid classes that carry the properties of the query.

In human vision, interpretation of images is based on feature comparisons with the known images in memory. The knowledge of image content is acquired in a supervised learning fashion. In computer vision image indexing can be acquired using artificial neural networks (NNET) that are trained in a supervised fashion. The NNET forms a classifier for images, to be used in image classification for content

based indexing. Feature vectors are used instead of the images in the classification. Obtaining the feature vector is discussed in Section 3.2.

The training of the NNET makes it possible to classify images for content based image indexing. Training database is composed of sets that represent the contents so that the NNET can estimate the content based index of the images. For this purpose training database is divided into sets that reflect content indexes; classes. The indexes and the training database are mentioned in Section 4.2.

In Section 4.3 a background on artificial neural networks is given and the implementation of NNET Based CBIIR is discussed.

4.2 INDEXING, THE CLASSES AND THE FEATURE VECTOR

The NNET based image classifier is formed using the training process which is discussed in Section 4.3. The training data makes it possible to classify the images for content based indexing. A training database is formed from sets of images, where each set represents a specific index for content based indexing.

The used database is traced and divided into 7 classes. These classes are: buildings, flowers, mountains, sea, river-waterfall, sunset and autumn-forest which are the contents that can be picked from the database. The aim was to test the content based indexing and retrieval system formed so a basic structure of image classification is formed and used as a starting point, better and more general classification structures can be built for more general databases as a future work.

In Figure 4.1, images from the Building Class are given. The common characteristics of this class are the buildings and the horizontal and vertical edges involved. In Figure 4.2, images from the Flower Class are given. The common characteristics of this class are the flowers, numerous color distributions and numbered edges involved. In Figure 4.3, images from the Mountain Class are given. The common characteristics of this class are the mountains and the blue sky above the brown or white mountains with curved edges involved. In Figure 4.4, images from the Sea Class are given. The common characteristics of this class are the blue sea, the sky and the horizontal edge from the horizon. In Figure 4.5, images from the Autumn-Forest Class are given. The common characteristics of this class are the trees, green or orange color distribution and vertical edges from trunks of the trees.

In Figure 4.6, images from the Sunset Class are given. The common characteristic of this class is the orange red circle on a dark background. In Figure 4.7, images from the River-Waterfall Class are given. The common characteristic of this class is the white stream with bent edges as well as the blue rivers.

Examining these examples it is observed that the image indexes are not exclusive. To index images for all the available contents multiple outputs are enabled. Examples such as buildings near sea have building and sea outputs. A building in a forest has building and sea outputs. A mountain, sea and sunset combination outputs mountain, sea and sunset indexes. The training database is adjusted to train the NNET with multiple outputs.

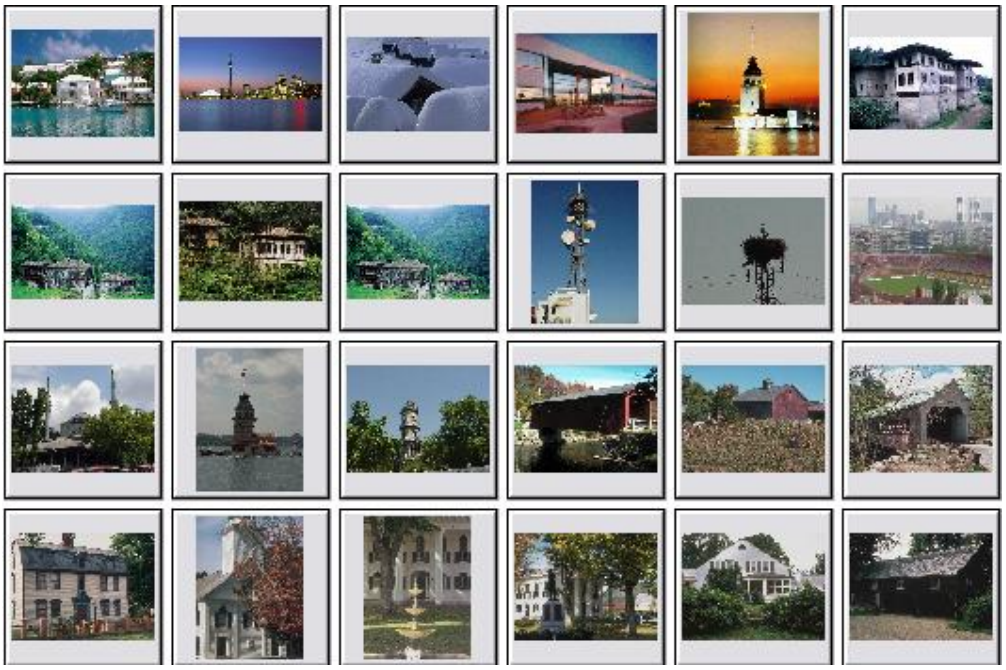

Figure 4.1 Sample Images from the Building Class

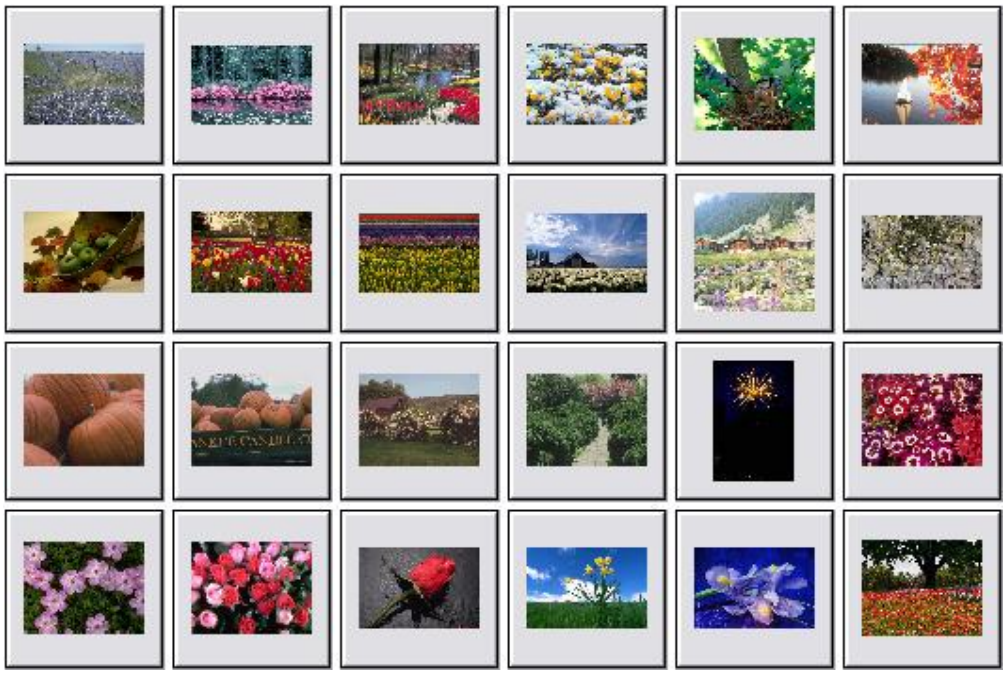

Figure 4.2 Sample Images from the Flower Class

Figure 4.3 Sample Images from the Mountain Class

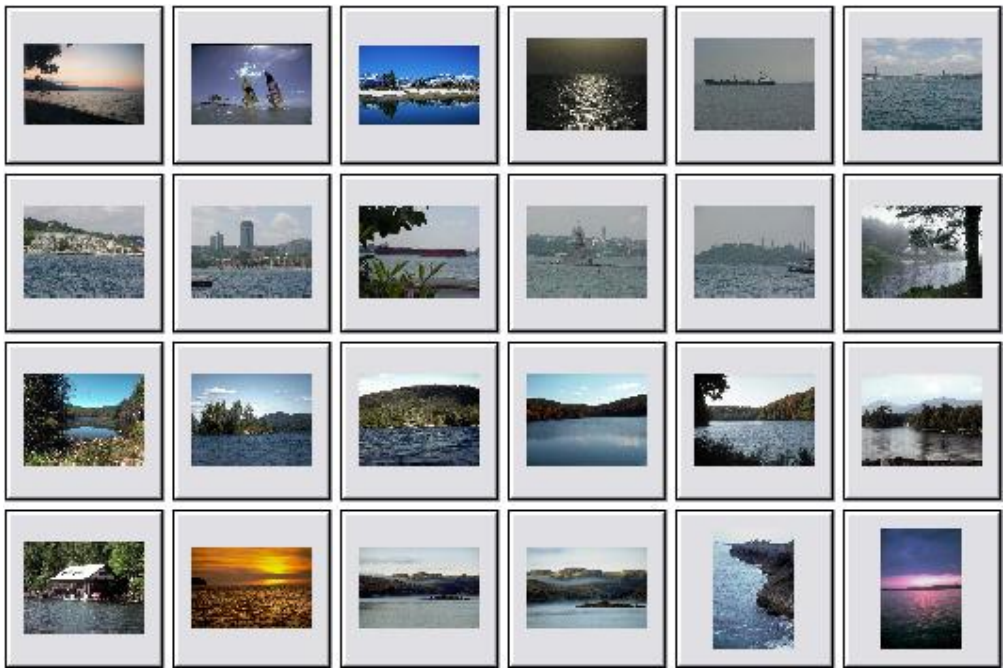

Figure 4.4 Sample Images from the Sea Class

Figure 4.5 Sample Images from the Forest-Autumn Class

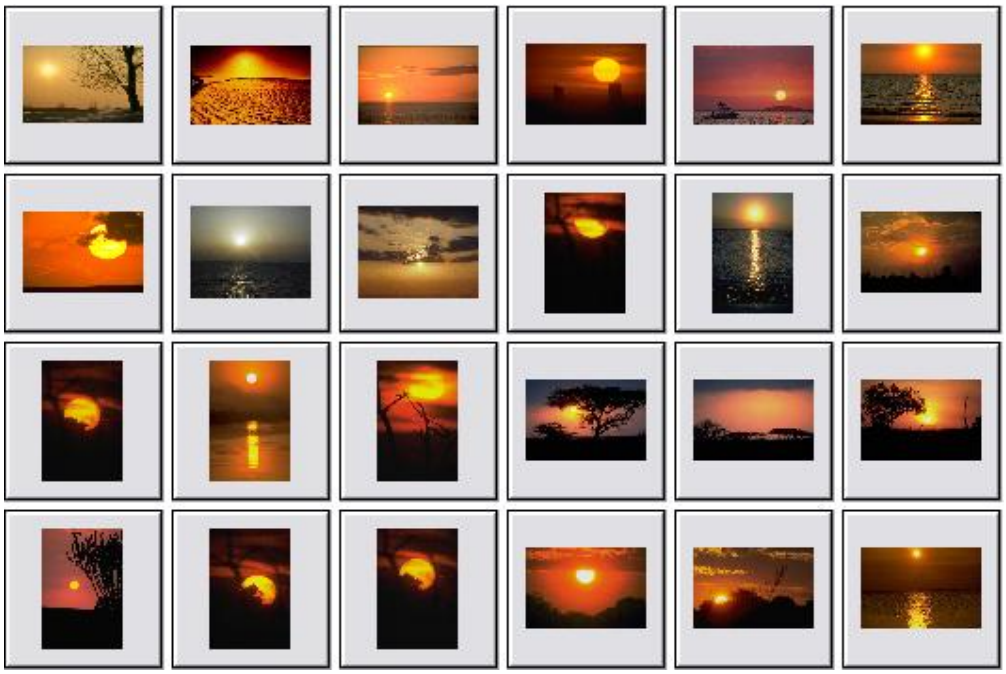

Figure 4.6 Sample Images from the Sunset Class

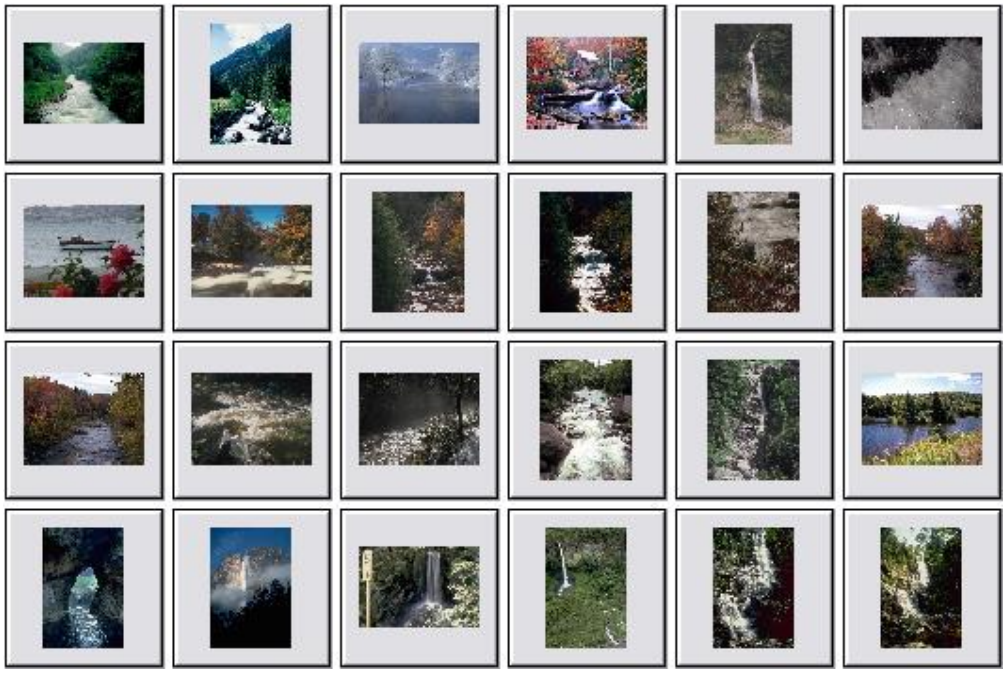

Figure 4.7 Example images from the River-Falls Class

4.3 IMAGE CLASSIFICATION BASED ON BACKPROPOGATION NNETS

In image classification for Content Based Image Indexing and Retrieval, images are indexed due to their content. An Artificial Neural Network is used to classify the images. Training database is formed in a supervised fashion to train the NNET. Training database is formed from indexed images as mentioned in Section 4.2 Block diagram of neural network based classifier is given in Figure 4.8

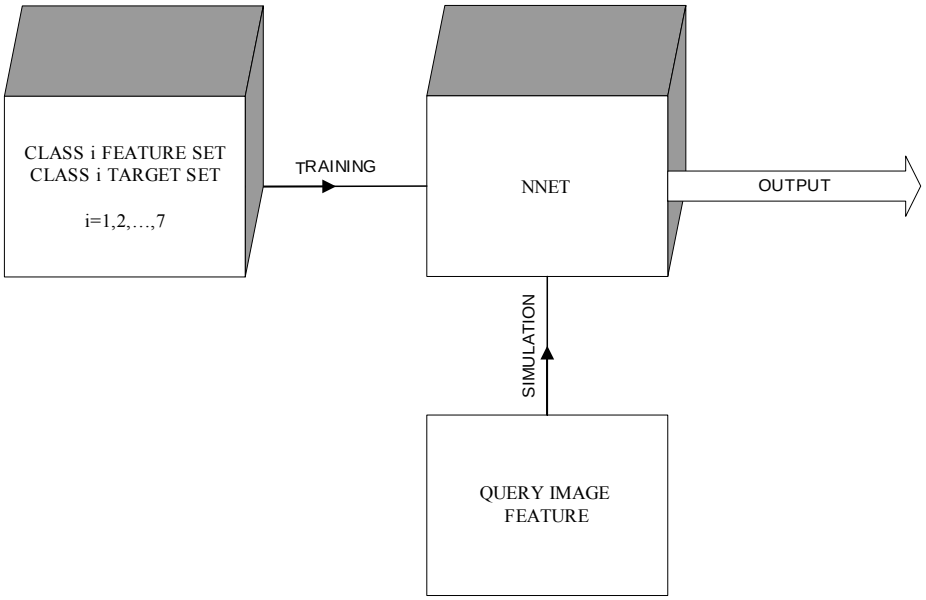

Figure 4.8 Block Diagram of CBIIR using NNET

The NNET is trained with the feature sets from the training database to give the class as the output. In the training session the NNET is adjusted to give desired targets for the training set with minimum predefined error (goal). An optimization and feedback mechanism is used to achieve the goal. Simulating the trained NNET with an input image, the NNET estimates the class(es) as the output. A background on neural networks is given in Section 4.3.1, and the used neural network based classifier structure is discussed in detail in Section 4.3.2.

4.3.1 Background on Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. The network function is determined largely by the connections between elements as the neural networks in the nature. We can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. [9]

Training the neural networks, the weights are adjusted so that a particular input leads to a specific target output based on a comparison of the output and the target, until the network output matches the target. Many such input/target pairs are used as the training set to train a network, which can be called as a supervised learning. A block scheme of the procedure is given in Figure 4.9.

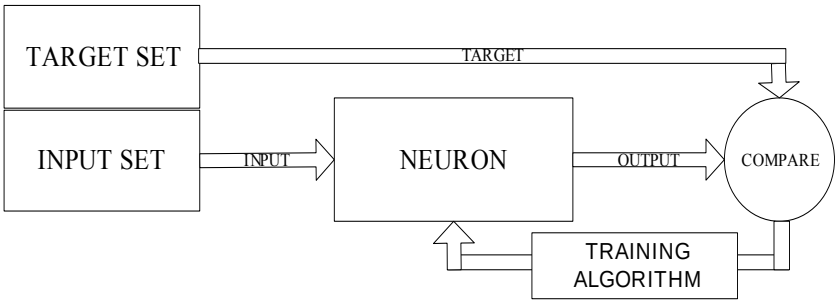

Figure 4.9 Block Scheme of NNET Training

By supervised learning the network is trained to give desired outputs to predefined inputs using mean-squared error optimal learning [8]. After learning NNET estimates outputs for new inputs by simulating. Neuron is the basic element of the NNET, model of which is given in Figure 4.10.

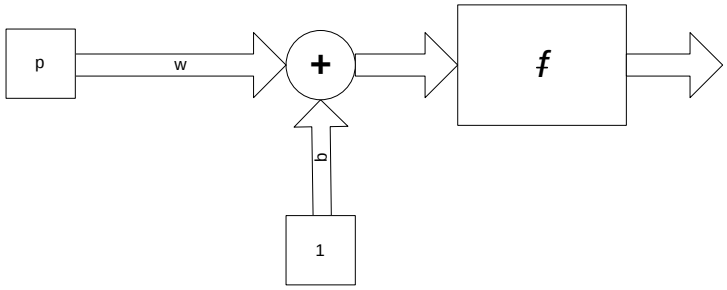

Figure 4.10 Neuron Model

The mathematical model of a neuron is formulated as

$$a = f(p * w + b) \tag{4.1}$$

where a is the output, w is the weight, p is the scalar input and b is the bias. Basically the neuron model is a weighted sum of input elements with the biases to give a desired number of outputs. The outputs are processed with functions to add functional properties to the weighted sum.

The weights and biases are determined to their values in the training process. In the train process the weights and biases are computed for a performance goal of predefined error which is calculated as the difference of output and the desired output. The function involved changes the properties of the NNET. A hard-limit (signum) function results in the perceptron which is useful in binary classification. A linear function results in linear filters which is useful in linear applications to estimate functions greater than unity in absolute value, and a nonlinear function is useful in estimating nonlinear functions.

Neurons may be used parallel to form a one layer neural network. A layer of S neurons is given in Figure 4.11.

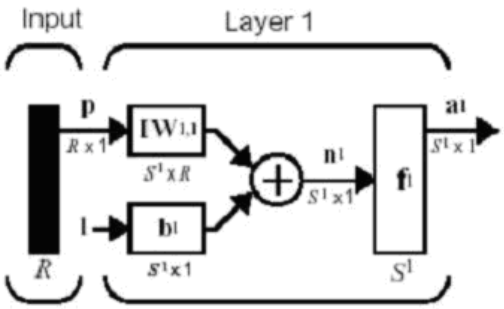

Figure 4.11 Layer of neurons

In this model the input vector is **p** where **p** is of R length, W is the weight matrix of dimensions S and R, **b** is the bias vector of length S, and S is the length of output vector **a**. The function denoted by **f** defines the neuron's property where **f** may be unity, hard-limit, linear or nonlinear functions [9]**.**

The NNET may be designed as single layer network or multiple layer network. In a single layer NNET Widrow-Hoff learning rule is used to implement Least Mean Square Error to find the minimum error solution in the training process. In multiple layer NNETs, the layer(s) before the output layer are called the hidden

layer(s). Feed Forward NNET is an example of multi layer neural networks that may have nonlinear and linear functions to estimate a general function.

Back propagation is the algorithm which is a generalization of Widrow-Hoff method to non-linear differentiable functions and multiple layers. There are many ways that back propagation NNETS are implemented. Steepest gradient descent is used as a general method. It can be formulated as

$$x_{k+1} = x_k - \alpha_k g_k \tag{4.2}$$

where x_k is the kth weight and bias vector with g gradient and α is the learning rate. Learning rate is useful to reach the goal and train the network in a narrower time, but a high learning rate may cause instability resulting in non convergence to the goal. The bias and weight vector is changed to employ the most rapid difference in gradient. The NNET is trained until the goal or the minimum gradient is reached.

In training processes preprocessing is useful to train the networks efficiently. In preprocessing the input vectors are normalized in mean and standard variation and than principal component analysis is employed. By this way the network is trained with less data since principal component analysis uncorrelates the input vectors and decrease the input vector size by eliminating the inputs that contribute less than a predefined variance. When the NNET is preprocessed before training, post processing is required with the simulation. In post processing the input is normalized and rescaled using the preprocessing data.

The artificial neural network structure that is used to form an image classifier is discussed in Section 4.3.2.

4.3.2 NNET Classifier for Image Indexing

Feed forward back propagation neural network with variable learning rate is used as the classifier for images. The model of the used NNET structure is given in Figure 4.12, where p is the input feature vector and a is index of the image for CBIIR, W demotes bias matrices and b denotes bias vectors. The weights and biases are determined in the training step. This model is given with the number of outputs in the hidden layer 15 and the length of the input vector is 102.

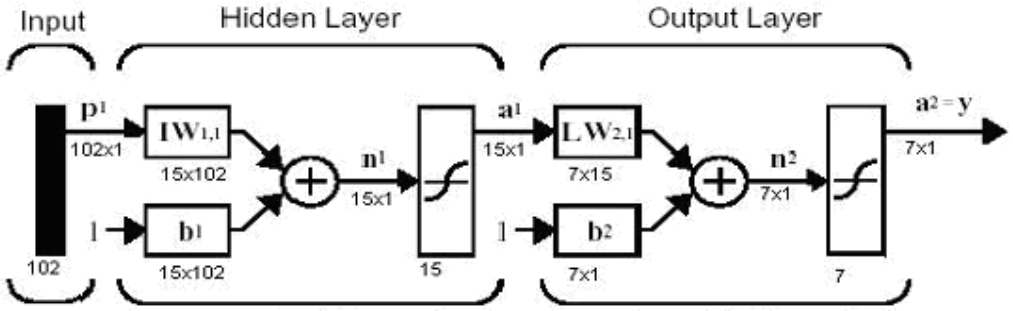

Figure 4.12 Model of the used NNET structure

Logarithmic sigmoid function is used as the hidden and output layer neuron function. It is aimed to handle the nonlinearities and to have an output range of 0 to 1. An illustration of logarithmic sigmoid (log-sig) is given in Figure 4.13. This function is chosen to be nonlinear to fit as a general purpose class estimator.

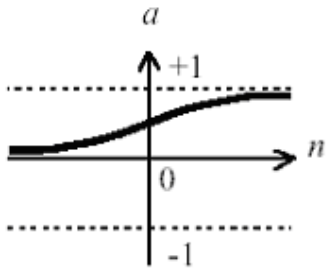

Figure 4.13 Illustration of Logarithmic Sigmoid Function

The output of the NNET for an input is in the range 0 to 1. To index an image to a class the corresponding output is compared with a threshold. The threshold is obtained to be 0.2 with the tests. This is discussed in Section 5.2. The outputs are desired to index available classes of the image so that multiple outputs are enabled. This is discussed in detail in Section 5.1.

The used performance goal is minimum mean square error where the error is calculated from the difference of NNET output with the target output. A goal of 0.4 is found to be the optimum. This is discussed in Section 5.2.

Mean and Standard Value Normalization as well as Principal Component Analysis (PCA) is used on the input feature vectors to enhance the NNET outputs while fastening the training periods. In Figure 4.14 training curve of NNET without preprocessing is given whereas in Figure 4.15 training curve of NNET with

preprocessing given. As seen from the Figures, the preprocessing clearly fastens the process as well as enhancing the performance goal.

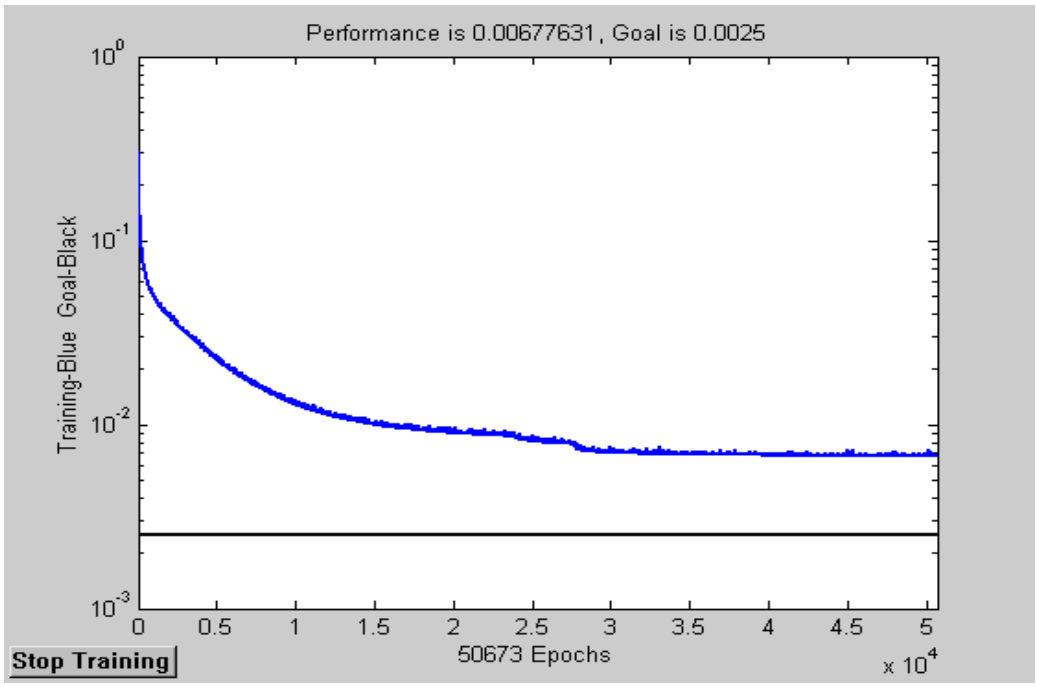

Figure 4.14 Training Performance Curve without Preprocessing

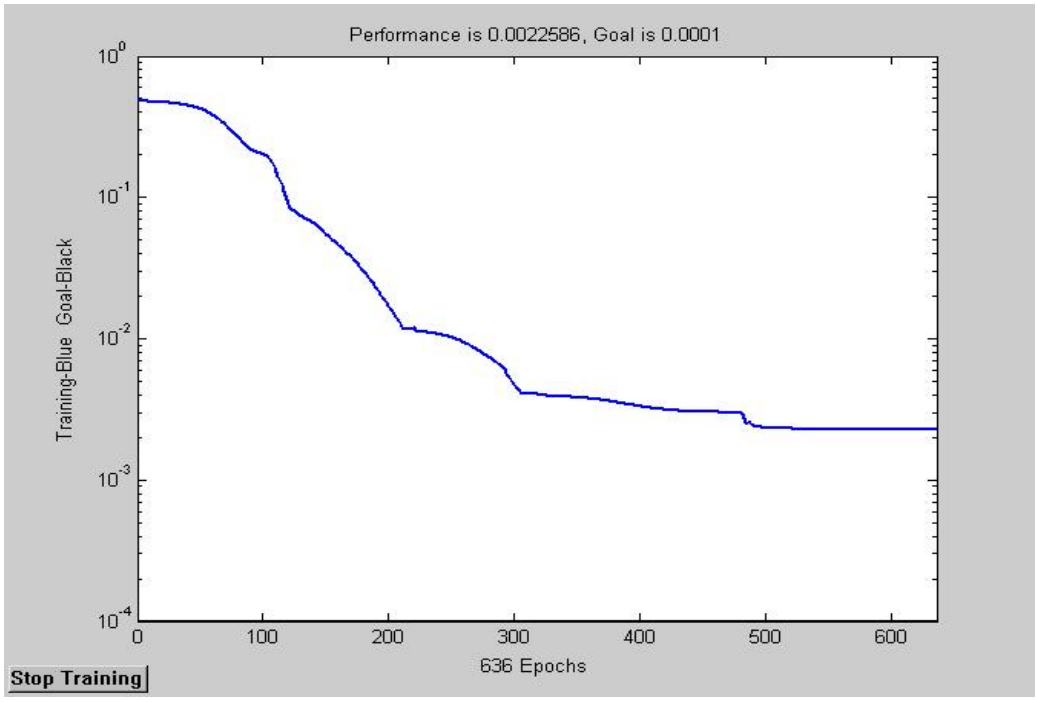

Figure 4.15 Performance Curve with Preprocessing

In PCA, principal components are computed, so that the output vector components are uncorrelated and ordered according to the magnitude of their variance. Those components contributing small variance are eliminated. In the designed NNET those principal components that contribute less than 0.1% to the total variance in the data set are eliminated. The tests on this value is discussed in Section 5.2.

The number of outputs of the hidden layer (S1) is an important issue in designing the NNET. It affects the number of neurons used. If too many neurons are used the NNET will memorize the training data and the generalization will fail. Another point is that the necessary number of neurons increases with the increasing complexity in relationships between input and output data. This is required to handle the complexity. So that an optimum should be found. There is an other issue that affect the performance of the NNET: the number of available data in the training database. With the increasing number of neurons, it is required to supply an increasing number of training data for the stability in the output. In Figure 4.16. The S1 vs. class recall for waterfall class is given. A train database of 25 images for each class and an independent test database of 20 images for each class is used. For each S1, 20 NNET is trained and the outputs are observed not to be stable with a peak to peak difference of 35%. In the figure the trials are plotted on the graph. To optimize these issues leave one out method is used in training and testing the NNET structure designed. With this method peak to peak difference in trials is reduced to 17% for the worst case. This is discussed in Section 5.2 with the test results.

In this Chapter, Content Based Image Indexing and Retrieval using Image Classification with artificial neural networks is discussed. The images in the database are indexed using the classifier designed. With queries as the content of the images to be retrieved, the indexed images in the database are retrieved. The indexes represent the available contents in the image so that they can be retrieved upon their content. The indexing is done to give the available contents in the image. The classification results are discussed in Chapter 5.

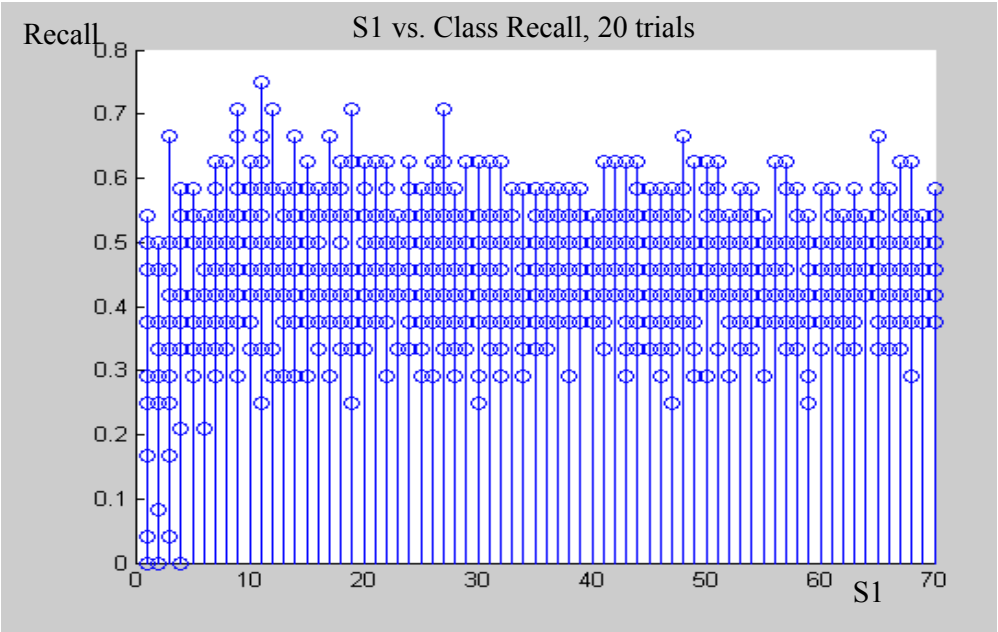

Figure 4.16 S1 vs. Waterfall Class Recall, 20 trials

CHAPTER 5

SIMULATION RESULTS

In the previous sections a classification method for content based image indexing is given, feature extraction for image retrieval is discussed and MPEG-7 visual descriptors are reviewed. In this thesis, the classifier and the features are implemented and tested using MATLAB® environment. In this chapter, the tests are presented and the test results are discussed.

The performances of the classification structure and the features are tested using leave one out method. A database of 357 color images of various sizes with scenery contents is used. The ground truth for the database is formed to compare the classification results with the desired results. Success, recall in classes and mean square error is used to observe the performance of the system with multiple outputs. The test method, the database structure and the performance evaluation are discussed in Section 5.1.

The NNET classifier design involves selecting parameters that are important in the system performance. These parameters are; the number of outputs in the hidden layer, the PCA parameter, goal in training and the threshold for capturing the multiple outputs. The system is tested to obtain the optimum values for these parameters. These tests and results are discussed in Section 5.2.

Features presented in Chapter 3 have parameters that can affect the retrieval performance. These parameters are color space in color histogram correlations, the level of multi resolution in detail histogram correlations and the number of frequency channels in directional filtering. The features are tested to obtain optimum values for these parameters. These tests and their results are discussed in Section 5.3. The overall performance of the CBIIR system with the extracted feature vector is

discussed in Section 5.4. The features are compared with the color and texture features standardized in MPEG-7, and the results are given and discussed in Section 5.5. In Section 5.6, example images that are indexed using the algorithm proposed are given.

5.1 TRAINING & TESTING METHODOLOGY

The number of images in the training database is important in NNET design as mentioned in Section 4.3. To test the design using all the data available leave one out method is used. This method is implemented as a loop: for all files in the database, each time one file is excluded from the database, NNET is trained with the remaining images and tested with the excluded image.

The image database used in training and testing is composed of 357 color images of various sizes. The content of the images are indexed as sunset, mountains, river-waterfall, autumn-forest, buildings, sea and flowers. The classes are labeled as Class 1, Class 2 … Class7 respectively through out this chapter for convenience. The ground truth data is formed in a supervised fashion: for each content available in an image corresponding container in the image's tag is marked. These tags are held in the content file of the database. The feature vectors for the images in the database are held in the signature file. The scheme of image database is given in Table 5.1 with an example. The first image belongs to flowers class, second image belongs to mountain and autumn-fall classes and the third image belongs to sunset and sea classes.

Table 5.1 Database format

Image ID	Image content	Feature vector
Key	C1 C2 C3 C4 C5 C6 C7	f1 f2 … f 96
1	0 0 0 0 0 0 1	x1x2 … x96
2	0 1 0 1 0 0 0	y1y2 … y96
3	1 0 0 0 0 1 0	z1z2 … z96
…	…	…

Image content in the tag corresponds to the desired indexing output (target) for the NNET training and tests. Key corresponds to the unique image identification number, defined for each image. Feature vector corresponds to the feature vector of the image.

The performance is calculated using the ground-truth data (image content in tags) and the indexing output. Success is the ratio of the number of indexed images with at least one correct classification output. Mean square error (MSE) is the mean square error between the classification output and the ground-truth data. The recall for each class is used to observe the performance of indexing over the classes. In the literature, for measuring the performance of retrieval systems recall and precision are commonly used. Recall is the ratio of the number of relevant retrieved data to the number of relevant data available in the database. Precision is the ratio of the number of relevant retrieved data to the number of retrieved data [37]. There is a tradeoff between these two and generally an optimum point is used. In this work success and MSE is used to measure the performance of indexing with the multiple content. Recall over classes is also given to measure the performance of retrieval. Since MSE is usable here and gives more information it is preferred to precision.

The tests are tried 20 times, each time with a new NNET to observe the stability of test results. The average values of these trials are given with the difference of maximum and minimum values (called as success_err and MSE_err in the result graphs) as and indication of deviation. Standard deviation is not used since the peak to peak value is preferred to observe the stability. For the CBIIR system performance, standard deviation is given as well.

The test results are given in graphs and tables. MSE and Success curves are given on the same graphs to observe the performance easily, so that the y axis is not labeled in the graphs. It is the ratio for success and success_err and the MSE value for MSE and MSE_err. The test results used in these graphs and class recall rates are provided in the tables.

5.2 TESTS OF NNET PARAMETERS

A feed forward back propagation neural network is designed to be the classifier in this thesis. The performance of the neural network is affected by the parameters used in the design. These parameters are the number of outputs in the hidden layer (S1), goal error and PCA parameter.

S1 is important since it defines the number of neurons in the NNET. For training the NNET for complex functions a high number of neurons are necessary.

Using too many neurons results in over fitting and memorization of the training data so that the generalization fails. The structure is tested with the feature set discussed in Chapter 3 with the expected parameters: YCbCr color space for CHC, 16 frequency channels for directional filtering and 3 level for DWT. Taking MSE goal as 0.04, PCA parameter as 0.001, and threshold as 0.2. The results are given in Figure 5.1 and Table 5.2. In the table the results are given for the S1 values that are given in the heading row and the performance measures that are given in the heading column. From the results it is observed that minimum MSE is obtained with S1=7 but the success is not enough, taking it to be around 90%, S1=11 is a good point. Since it provides good performance and not too large, 11 is taken as the optimum point for S1.

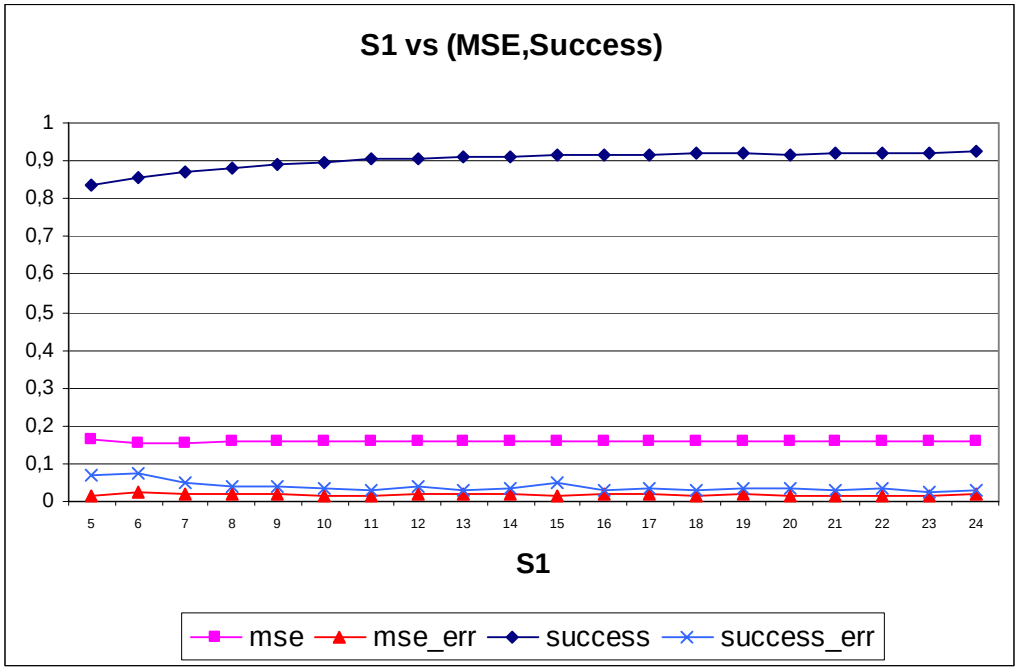

Figure 5.1 S1 vs. MSE and S1 vs. Success Graph

Table 5.2 Effects of S1 on the performance of NNET based image indexing

S1	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**
mse	2	0,156	0,154	0,157	0,159	0,158	0,158	0,16	0,159	0,161	0,16	0,16	0,159	0,158	0,16	0,159	0,159	0,159	0,159
mse_err	5	0,026	0,022	0,019	0,021	0,016	0,015	0,018	0,018	0,018	0,017	0,019	0,019	0,016	0,018	0,016	0,014	0,014	0,015
Success (%)	83,4	85,7	87,2	88,1	89,2	89,8	90,4	90,5	90,9	91,2	91,6	91,6	91,8	92,1	92,2	91,7	92,1	92,1	92,2
scs_err (%)	7	7,3	4,8	4,2	3,9	3,6	2,8	3,9	2,8	3,6	4,8	2,8	3,6	3,1	3,4	3,6	2,8	3,4	2,2
Class1 (%)	83,4	85,5	89	88,8	90	91,3	90,4	89,8	92,1	90,6	91,3	91,7	91,5	92	91,8	92,6	90,8	91,6	91,6
Class2 (%)	66,1	68	68,7	70	71,6	72,5	72,9	71,7	73,6	74	73,9	73,9	74,5	75,6	73,9	75,9	75,7	76,5	75,4
Class3 (%)	69,8	71,6	76,2	75,2	76,3	77,4	79,3	78,8	79,7	78,7	78,6	79,7	79,3	78,6	79,7	79,4	80	80,4	79,8
Class4 (%)	68,1	70,4	72,7	74,9	76	76,3	77	78	77,5	77,8	79,4	80,2	79,9	80	80,1	80,2	80,1	81,1	80,8
Class5 (%)	71,4	73,1	76,8	77,3	78,8	79,6	79,6	81,5	81,2	80,5	80,8	81,9	82,4	82,2	81,9	82,5	82,7	82,3	82,2
Class6 (%)	77,2	77,8	79,7	80,7	82,3	82,7	83,3	83,6	84,1	84,5	84,3	84,4	84,6	85,6	85,7	85,5	85,6	85,8	86,2
Class7 (%)	69,3	76,5	76,1	80,8	80,9	82,1	82	83,1	84	84,7	86,4	85,3	85,9	85,4	86,4	84,4	86,2	85,5	85,6
Class1_err (%)	14,3	10,7	12,5	8,9	12,5	12,5	10,7	10,7	10,7	12,5	10,7	8,9	7,1	8,9	8,9	7,1	10,7	8,9	7,1
Class2_err (%)	9,3	18,6	16,3	10,5	12,8	16,3	11,6	12,8	9,3	14	10,5	9,3	10,5	14	11,6	8,1	15,1	8,1	9,3
Class3_err (%)	24,5	18,9	15,1	13,2	11,3	11,3	13,2	11,3	13,2	11,3	15,1	11,3	11,3	9,4	7,6	9,4	9,4	11,3	17
Class4_err (%)	15,7	12,9	14,3	11,4	12,9	10	12,9	8,6	12,9	11,4	10	10	7,1	8,6	7,1	10	7,1	12,9	7,1
Class5_err (%)	22,6	12,9	9,7	12,9	8,1	14,5	14,5	11,3	16,1	9,7	8,1	11,3	8,1	9,7	12,9	9,7	9,7	11,3	9,7
Class6_err (%)	10,4	10,4	10,4	10,4	8,3	6,3	8,3	6,3	6,3	7,3	5,2	7,3	8,3	7,3	7,3	6,3	7,3	7,3	5,2
Class7_err (%)	18,9	18,9	17	11,3	17	15,1	18,9	13,2	7,6	9,4	11,3	7,6	13,2	11,3	17	13,2	7,6	11,3	7,6
Class_avg (%)	72,2	74,7	77	78,2	79,4	80,3	80,6	80,9	81,7	81,5	82,1	82,4	82,6	82,8	82,8	82,9	83	83,3	83,1
Cl_err_avg (%)	16,5	14,7	13,6	11,2	11,8	12,3	12,9	10,6	10,9	10,8	10,1	9,4	9,4	9,9	10,3	9,1	9,6	10,2	9

The threshold can be thought as a property affected by the neuron function. Log-sigmoid function is used so that the output is in the range 0 to 1. By thresholding the output value the image is decided to be in the content class or not. The effects of threshold is tested with the parameters S1=19, goal=0.05 and PCA_parameter=0.001 and the results are given in Figure 5.2, Figure 5.3 and Table 5.3. In the table the results are given for the threshold values that are given in the heading row and the performance measures that are given in the heading column. As an optimum point 0.2 is chosen as the threshold. 0.25 is also a good point better in MSE, worse in success. 0.2 has less variance in success so it's preferred. In the table the results are given for the S1 values that are given in the heading row and the performance measures that are given in the heading column.

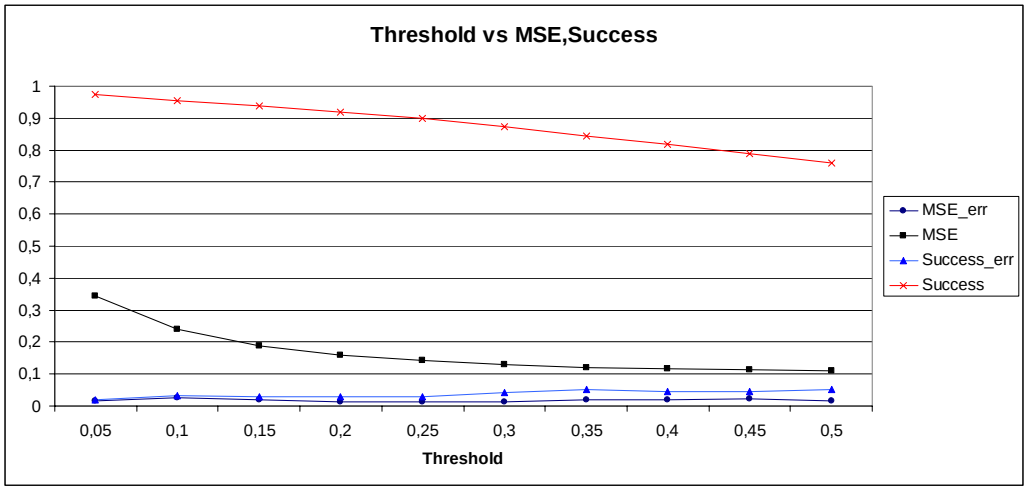

Figure 5.2 Threshold vs. MSE and Threshold vs. Success Graph

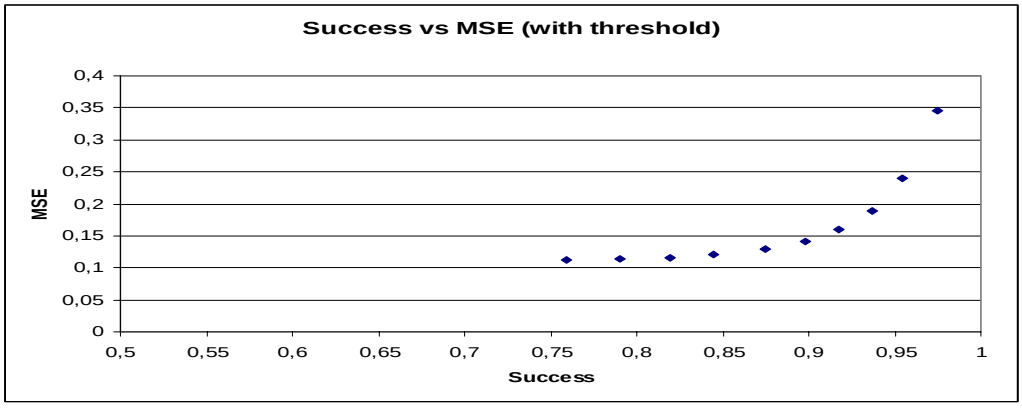

Figure 5.3 Success vs. MSE Graph with threshold changing

Table 5.3 Effects of threshold on the performance of NNET based image indexing

Threshold	**0,05**	**0,1**	**0,15**	**0,2**	**0,25**	**0,3**	**0,35**	**0,4**	**0,45**	**0,5**
mse	0,017	0,025	0,021	0,014	0,012	0,012	0,018	0,019	0,022	0,016
mse_err	0,345	0,241	0,189	0,16	0,141	0,129	0,122	0,116	0,113	0,112
Success (%)	2	3,4	2,8	2,8	3,1	4,2	5,3	4,5	4,5	5,3
Success_err(%)	97,5	95,5	93,7	91,8	89,9	87,5	84,5	81,9	79	76
Class1 (%)	97,7	95,6	93,5	91,6	90	88,8	86,8	85,9	84,3	83,4
Class2 (%)	93,1	87	80,6	74,9	69,1	64,4	59,2	55,1	50,9	46
Class3 (%)	90,9	86,3	83,8	80,5	77,2	74	70,4	66,9	64,5	60,7
Class4 (%)	91	86,7	82,4	79,9	75,9	72,4	68,2	65,1	61	58,2
Class5 (%)	93,9	88,5	85,1	80,7	77,9	75,1	70,7	68,1	64	60
Class6 (%)	92,7	89,8	87,5	84,9	82	79,7	76,9	74,1	70,9	67,6
Class7 (%)	94,2	90	87,8	86	84	81,1	78,1	75,1	71,5	68,3
Class1_err (%)	3,6	5,4	8,9	10,7	8,9	10,7	8,9	7,1	10,7	8,9
Class2_err (%)	5,8	11,6	8,1	8,1	9,3	12,8	9,3	14	12,8	14
Class3_err (%)	7,6	9,4	9,4	11,3	7,6	9,4	11,3	11,3	11,3	11,3
Class4_err (%)	7,1	5,7	8,6	10	10	11,4	10	8,6	10	8,6
Class5_err (%)	9,7	11,3	11,3	9,7	8,1	11,3	16,1	14,5	16,1	16,1
Class6_err (%)	5,2	6,3	5,2	8,3	8,3	7,3	7,3	8,3	9,4	10,4
Class7_err (%)	7,6	11,3	11,3	7,6	9,4	11,3	13,2	9,4	13,2	15,1
Class_avg (%)	93,3	89,1	85,8	82,7	79,4	76,5	72,9	70	66,7	63,5
Cl_err_avg (%)	6,6	8,7	9	9,4	8,8	10,6	10,9	10,5	11,9	12,1

The PCA is used to get rid of probable redundancy in feature vectors and increase the efficiency of the NNET. The tested PCA parameter is the percentage which the principal components that contribute less than that of the total variance in the data set that would be eliminated. The tests are held with S1=11, goal=0.001 and threshold=0.5, and the results are given in Figure 5.4 and Table 5.4. In the table the results are given for the PCA parameter values that are given in the heading row and the performance measures that are given in the heading column. The optimum point seems to be 0.001, but the difference in performance between 0.0001 and 0.001 is negligible so 0.001 is used. Examining the result it is observed that the performance decrease with the increase in parameter, that is because of the fact that the eliminated part of the feature vector is increased. With the decrease of the parameter, the performance also decreases that is because of the fact that the input vector is large for the efficient performance of the NNET designed. So that parameter 0.001 that results a good performance and gives a compact feature for the NNET input is selected.

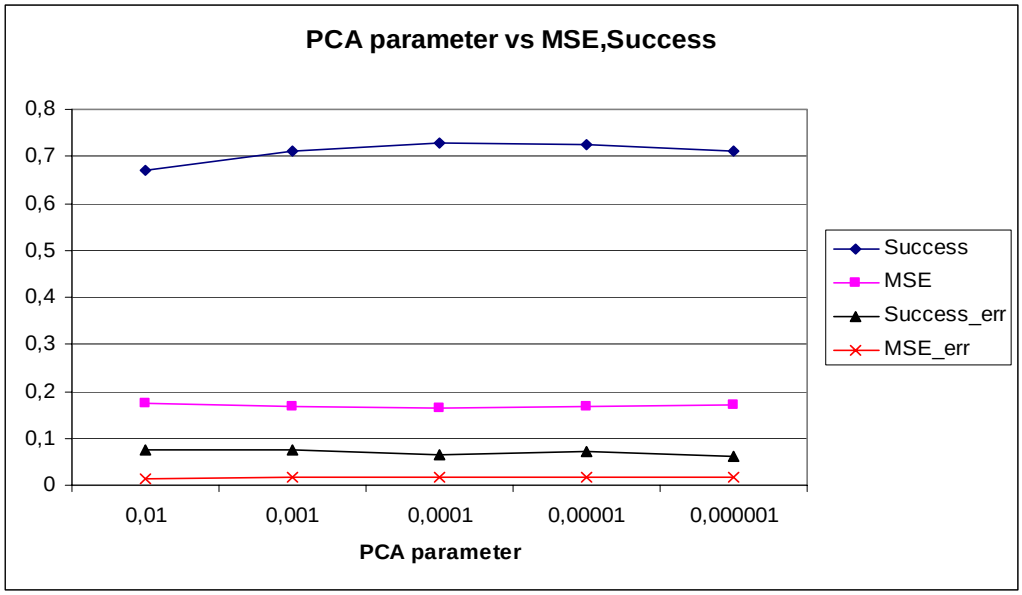

Figure 5.4 PCA parameter vs. MSE and PCA parameter vs. Success Graph

Table 5.4 Effects of PCA_parameter on the performance of NNET based image indexing

PCA parameter	**10^{-2}**	**10^{-3}**	**10^{-4}**	**10^{-5}**	**10^{-6}**
mse	0,175	0,169	0,165	0,166	0,171
mse_err	0,014	0,017	0,018	0,016	0,018
Success (%)	67,2	71	72,8	72,4	71
Success_err (%)	7,5	7,4	6,6	7,1	6,1

The MSE goal is the predefined error used in training the NNET. Using a small MSE goal usually does not result the best performance. The small MSE goal may result in over fitting and the lost of generalization. A high MSE may result in low performance since the NNET is not trained necessarily. The test results used in selecting the MSE goal is given in Figure 5.5 and Table 5.5. In the table the results are given for the goal values that are given in the heading row and the performance measures that are given in the heading column. Goal=0.04 is taken as the optimum point since it has better performance: small MSE and high stability in class recall.

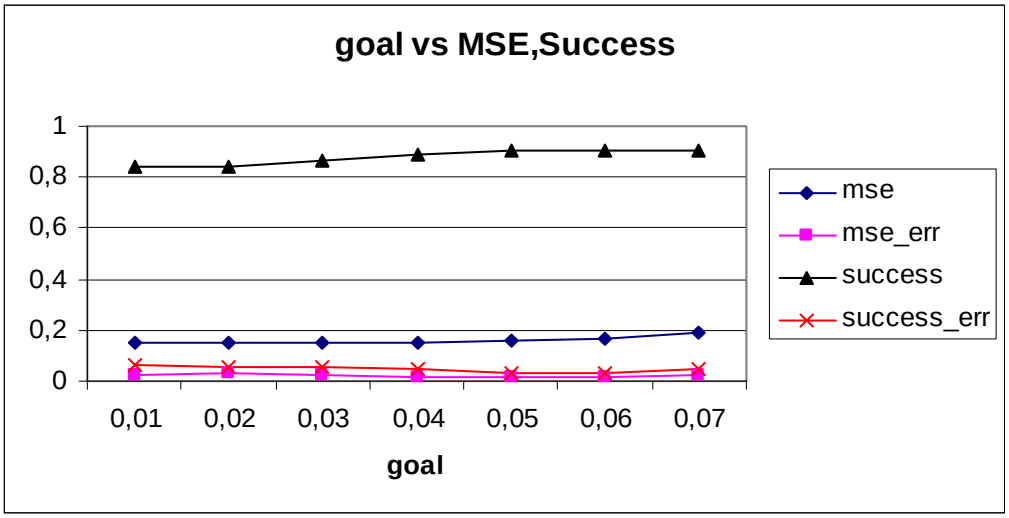

Figure 5.5 Goal vs. MSE and Goal vs. Success Graph

Table 5.5 Effects of goal on the performance of NNET based image indexing

goal	**0,01**	**0,02**	**0,03**	**0,04**	**0,05**	**0,06**	**0,07**
mse	0,148	0,148	0,149	0,152	0,158	0,17	0,187
mse_err	0,027	0,028	0,021	0,017	0,015	0,017	0,02
Success (%)	84,1	84,1	86,9	89	90,4	90,5	90,2
Success_err (%)	6,2	5,6	5,6	4,8	2,8	2,8	4,5
Class1 (%)	87,1	87,1	88,5	90,2	90,4	90,7	89,1
Class2 (%)	62,9	63,1	67	69,6	72,9	71,9	72
Class3 (%)	72,2	72,1	74,1	77,2	79,3	79,2	78,1
Class4 (%)	69,8	69,9	72,8	75,1	77	78,7	79,3
Class5 (%)	74,1	74	76,8	79,5	79,6	79,9	79,5
Class6 (%)	73,8	73,8	77,6	80,6	83,3	85,1	85,1
Class7 (%)	74,6	74,4	77,9	80,9	82	82,6	81,8
Class1_err (%)	10,7	8,9	12,5	10,7	10,7	10,7	7,1
Class2_err (%)	15,1	15,1	12,8	12,8	11,6	11,6	12,8
Class3_err (%)	22,6	22,6	15,1	13,2	13,2	11,3	11,3
Class4_err (%)	12,9	12,9	12,9	10	12,9	8,6	12,9
Class5_err (%)	12,9	12,9	9,7	11,3	14,5	11,3	11,3
Class6_err (%)	9,4	9,4	8,3	7,3	8,3	6,3	8,3
Class7_err (%)	13,2	11,3	13,2	13,2	18,9	15,1	15,1
Class_avg (%)	73,5	73,5	76,4	79	80,6	81,2	80,7
Cl_err_avg (%)	13,8	13,3	12,1	11,2	12,9	10,7	11,3

5.3 FEATURE PARAMETERS

In image classification for content based image indexing and retrieval the feature selection is as important as the classifier. The features designed and extracted

in this thesis study have some parameters that affect the performance of the indexing and retrieval system. The parameters of the features are compared using the NNET designed with the optimum parameters (S1=11, goal=0.04, PCA parameter=0.001 and threshold=0.2) as discussed in Section 5.2.

In color histogram correlations (CHC) the used color space is an important parameter. RGB, YCbCr and HSV color spaces are tried expecting that YCbCr will perform better since it is less sensitive to brightness changes. The test results are given in Figure 5.6 and Table 5.6. In the table the results are given for the color spaces that are given in the heading row and the performance measures that are given in the heading column. Examining the results it is observed that RGB and YCbCr perform better than HSV. YCbCr and RGB results are comparable: YCbCr is better in MSE and RGB is better in Success. YCbCr is preferred since the MSE outputs are more stable and recall rates over classes are more uniform.

Table 5.6 Effects of color space in color histogram correlations on the performance of NNET based image indexing

Color Space	**RGB**	**YCbCr**	**HSV**
mse	0,21	0,19	0,226
mse_err	0,014	0,009	0,016
Success (%)	80	77,3	68,2
Success_err (%)	4,5	5	4,8
Class1 (%)	86,1	79,6	71,6
Class2 (%)	78,5	67	56,3
Class3 (%)	69,3	62,5	41,7
Class4 (%)	63,6	58,7	56,4
Class5 (%)	52,6	51	52,3
Class6 (%)	68,9	68,7	64,3
Class7 (%)	69,4	64,5	57,4
Class1_err (%)	3,6	14,3	10,7
Class2_err (%)	12,8	10,5	12,8
Class3_err (%)	17	15,1	13,2
Class4_err (%)	14,3	12,9	12,9
Class5_err (%)	11,3	21	12,9
Class6_err (%)	10,4	9,4	10,4
Class7_err (%)	9,4	17	11,3
Class_avg (%)	69,7	64,6	57,1
Cl_err_avg (%)	11,3	14,3	12

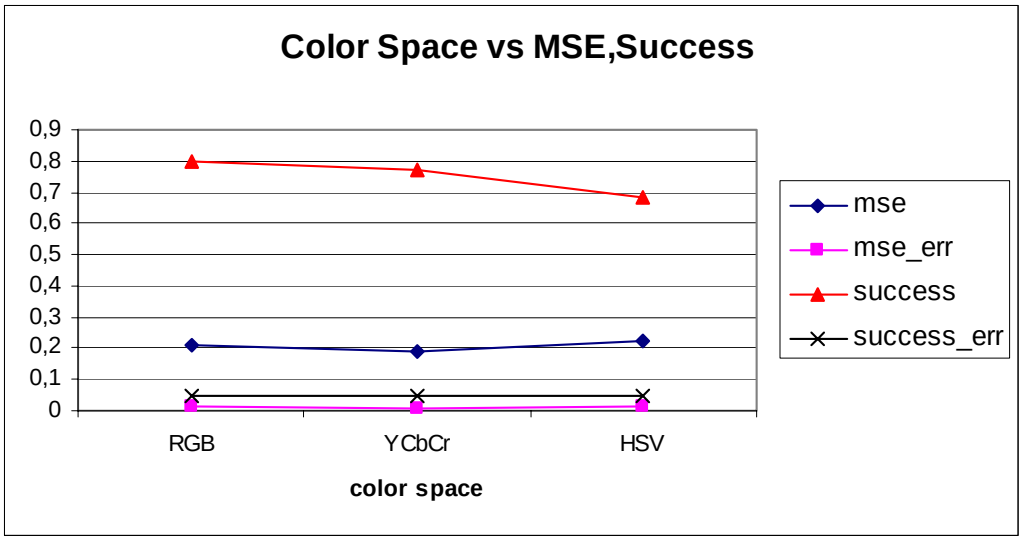

Figure 5.6 Color Space in CHC vs. MSE and Color Space in CHC vs. Success Graph

In Detail Histogram Correlations (DHC) the multi resolution level is an important parameter. One, two and three level 2D DWT is applied to the images. In each level correlations among histograms of horizontal, vertical and diagonal details are used. The histogram of DWT output indicates the distribution of coefficients over horizontal, vertical and diagonal details. The features extracted from cross correlations are used in the feature vector. The effect of increasing multi resolution level is given in Figure 5.7 and Table 5.7. In the table the results are given for the detail level values that are given in the heading row and the performance measures that are given in the heading column. As expected with the increase in multi resolution level the amount of supplied information is increased and the performance increases. There is a limit that this information increases the performance. As seen from the results Level2 and Level3 are comparable in results and outperform Level1. Level 3 is used as the feature vector.

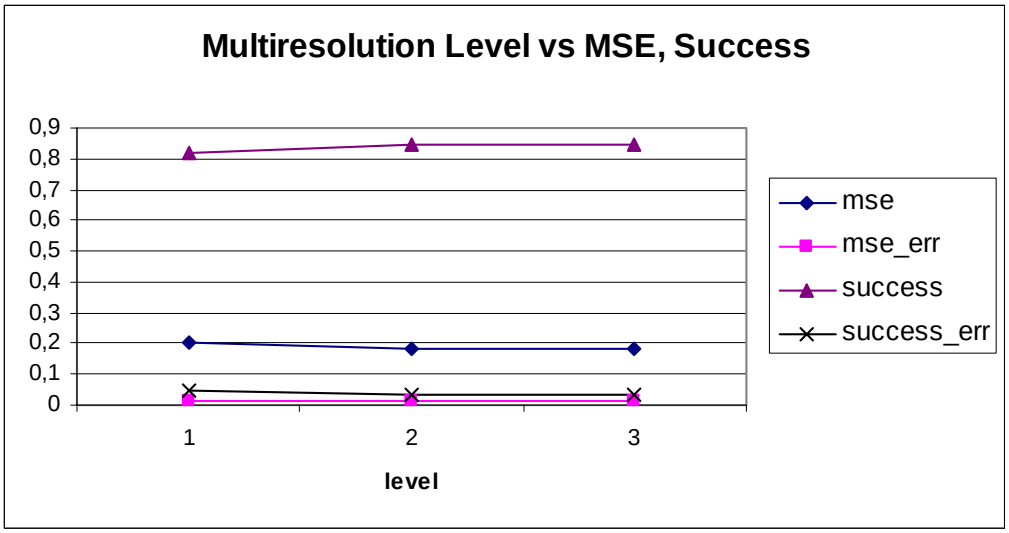

Figure 5.7 Level vs. MSE and Level vs. Success Graph

Table 5.7 Effects of detail level in detail histogram correlations on the performance of NNET based image indexing

Detail Level	**level1**	**level2**	**level3**
mse	0,203	0,181	0,184
mse_err	0,015	0,011	0,016
Success (%)	82,2	84,9	84,7
Success_err (%)	5	3,1	3,6
Class1 (%)	80,2	83,4	81,3
Class2 (%)	71,4	73,4	70,5
Class3 (%)	67,6	68,3	67
Class4 (%)	68	71,6	72,7
Class5 (%)	74,7	79,7	76,1
Class6 (%)	78,8	78,3	79,9
Class7 (%)	64,5	75,1	75,1
Class1_err (%)	7,1	7,1	12,5
Class2_err (%)	9,3	8,1	10,5
Class3_err (%)	15,1	11,3	11,3
Class4_err (%)	8,6	12,9	7,1
Class5_err (%)	4,8	11,3	16,1
Class6_err (%)	5,2	7,3	10,4
Class7_err (%)	11,3	18,9	18,9

In extracting the texture and edge information directional filters are used to extract the energies in frequency channels. This information is extracted from the color space decompositions. The number of the frequency channels affect the indexing and retrieval performance. The number of frequency channels is tested as

well as the color space to find the optimum values. The results are given in Figure 5.8, Table 5.8 and Table 5.9. In the tables the results are given for the number of frequency channels per quadrant that are given in the heading row and the performance measures that are given in the heading column. In Table 5.8, test results for RGB color space are given and in Table 5.9, test results for YCbCr color space are given. 7 frequency channels per quadrant: 14 frequency channels is the optimum number of frequency channels considering good results in MSE and Success and less computational load. RGB Color space is used since the outputs are more stable and the performances have uniform distribution over classes.

Table 5.8 Effects of the number of frequency channels per quadrant in RGB color space in directional filtering on the performance of NNET based image indexing

RGB - Number of frequency channels	**5**	**6**	**7**	**8**	**9**	**10**
mse	0,206	0,206	0,201	0,2	0,2	0,198
mse_err	0,007	0,012	0,015	0,011	0,021	0,017
Success (%)	73,5	75	75,6	76,3	76,5	77,3
Success_err (%)	1,7	3,4	5,3	3,9	5	6,4
Class1 (%)	52,5	59,6	57,1	58,6	57,5	60
Class2 (%)	61,2	61,6	63,3	61,9	64,7	63,3
Class3 (%)	55,9	54,7	58,5	57	57	57
Class4 (%)	58,6	62,6	60,3	64,9	63,1	66,3
Class5 (%)	65,8	66,1	62,3	68,4	66,5	68,4
Class6 (%)	67,5	68,5	68,5	68,1	67,9	70,4
Class7 (%)	74	72,5	75,5	73,2	74	74,7
Class1_err (%)	25	8,9	14,3	16,1	7,1	12,5
Class2_err (%)	17,4	10,5	8,1	8,1	11,6	9,3
Class3_err (%)	18,9	7,6	3,8	11,3	7,6	17
Class4_err (%)	11,4	11,4	5,7	7,1	8,6	11,4
Class5_err (%)	11,3	4,8	8,1	11,3	11,3	9,7
Class6_err (%)	4,2	7,3	6,3	11,5	7,3	4,2
Class7_err (%)	3,8	11,3	9,4	9,4	7,6	9,4
Class_avg (%)	62,2	63,7	63,6	64,6	64,4	65,7
Cl_err_avg (%)	13,1	8,8	8	10,7	8,7	10,5

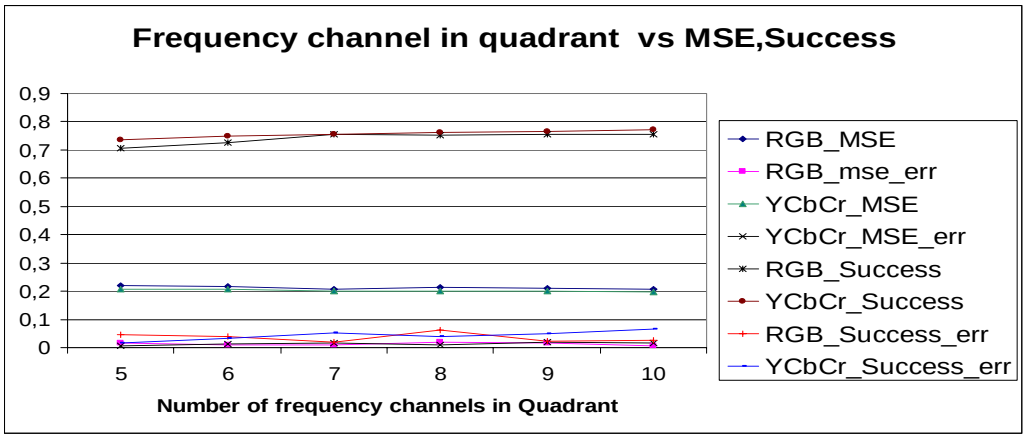

Figure 5.8 Performance Plot of Number of Frequency Channels in Quadrant3

Table 5.9 Effects of the number of frequency channels per quadrant in YCbCr color space in directional filtering on the performance of NNET based image indexing

RGB - Number of frequency channels	**5**	**6**	**7**	**8**	**9**	**10**
mse	0,219	0,217	0,206	0,213	0,209	0,208
mse_err	0,015	0,009	0,01	0,019	0,016	0,008
Success (%)	70,5	72,5	75,7	75,1	75,7	75,5
Success_err (%)	4,5	3,9	2	6,2	2,2	2,5
Class1 (%)	58,6	61,1	68,9	70	73,6	73,9
Class2 (%)	58,4	55,8	58,6	57,9	58,4	57,9
Class3 (%)	52,5	54,7	58,5	58,5	57,4	58,1
Class4 (%)	56,3	55,7	62,6	59,1	60,3	60,3
Class5 (%)	62,3	66,5	70,7	66,8	67,1	66,1
Class6 (%)	68,5	71,3	70,8	69,2	69,4	71,7
Class7 (%)	70,6	65,3	70,6	70,2	71,3	72,5
Class1_err (%)	14,3	8,9	8,9	12,5	7,1	8,9
Class2_err (%)	4,7	14	7	15,1	10,5	9,3
Class3_err (%)	9,4	11,3	7,6	15,1	7,6	5,7
Class4_err (%)	11,4	8,6	12,9	14,3	7,1	12,9
Class5_err (%)	9,7	8,1	3,2	9,7	3,2	6,5
Class6_err (%)	4,2	6,3	5,2	9,4	3,1	3,1
Class7_err (%)	11,3	5,7	7,6	9,4	11,3	5,7
Class_avg (%)	61	61,5	65,8	64,5	65,3	65,8
Cl_err_avg (%)	9,3	9	7,5	12,2	7,1	7,4

5.4 CBIIR PERFORMANCE

Using the results obtained in Section 5.2 and Section 5.3 the feature vector is extracted and the performance of the system is tested. The test results are given in

Table 5.10. The feature vector is composed from CHC features in YCbCr color space, DHC features with multiresolution level 3 and Directional Filtering with 14 frequency channels in RGB color space. The NNET is used with the parameters S1=11, goal=0.04, PCA parameter=0.001 and threshold=0.2. The feature vector is the combination of features with CHC in YCbCr, 3 level DHC and RGB directional filtering with 14 channels.

Table 5.10 CBIIR system performance

	Mean Result	**Max-Min Deviation**	**Standard deviation**
MSE	0.15	0.018	0.005
Success	89%	4%	1%
Class1 Recall Rate	90%	13%	3%
Class2 Recall Rate	70%	11%	3%
Class3 Recall Rate	76%	17%	5%
Class4 Recall Rate	76%	11%	3%
Class5 Recall Rate	79%	13%	3%
Class6 Recall Rate	80%	10%	3%
Class7 Recall Rate	82%	11%	3%

5.5 COMPARISON OF DERIVED FEATURES WITH MPEG-7 VISUAL DESCRIPTORS

In MPEG-7, color and texture descriptors are standardized as: dominant color descriptor (DCD), scalable color descriptor (SCD), color structure descriptor (CSD), color layout descriptor (CLD), homogeneous texture descriptor (HTD), texture browsing descriptor (TBD) and edge histogram descriptor (EHD). These are reviewed in Section 2.5. These descriptors are implemented and their performances are compared with the performances of features that are discussed in Chapter 3 using the NNET based classifier structure proposed in Chapter 4.

Although MPEG-7 standardizes the descriptors feature extraction is open to the developers. In this thesis the descriptors are implemented with some parameters, In deciding these, the two used points are the sizes of descriptors to be comparable with the features and easy computing. These parameters are: In DCD extraction 8 dominant colors are used where alternatives have less number of dominant colors. In SCD extraction 16 bin representation is used. In CSD extraction 128 bin representation is used where the alternative representations for SCD and CSD were

16, 32, 64, 128 and 256 bin representations. In CLD extraction 18 coefficients are used where the alternative were 12. In TBD extraction frequency domain is used where the alternative were the spatial domain. The performances are given in Table 5.11, where CHC corresponds to the color histogram correlation feature, DF corresponds to directional filtering and DHC corresponds to detail histogram correlations. In the table the results are given for the descriptors that are given in the heading row and the performance measures that are given in the heading column.

Examining the results, it is seen that features presented in this thesis are comparable and better than the color and visual descriptors of MPEG-7. With the highest success and lowest MSE, Detail Histogram Correlation feature stands as the best feature when used standalone. TBD has the worst performance since it is not enough for standalone use. Directional filtering and HTD are extracted using similar techniques but HTD is worse since it is optimized for texture for different resolutions of frequency channels while the other contains information on both color and texture. DCD, SCD and CHC features have similar performances since they are color descriptors and carry similar information.

Table 5.11 Descriptor performances

Descriptor	**DCD**	**SCD**	**CSD**	**CLD**	**HTD**	**TBD**	**EHD**	**CHC**	**DHC**	**DF**
mse	0,196	0,189	0,209	0,191	0,221	0,367	0,286	0,19	0,213	0,184
mse_err	0,016	0,018	0,018	0,021	0,03	0,026	0,029	0,009	0,019	0,016
Success (%)	77	75,5	80,4	70,6	77,5	59,6	58,3	77,3	75,1	84,7
Success_err (%)	4,5	5,9	4,5	5	4,8	5,6	7	5	6,2	3,6
Class1 (%)	87,3	77,8	81,4	80,6	78,7	55,2	43,8	79,6	70	81,3
Class2 (%)	64,9	63	74,1	61,5	59,1	56,4	46,8	67	57,9	70,5
Class3 (%)	63,2	54,9	69,6	67,3	57	47,6	52,2	62,5	58,5	67
Class4 (%)	61,2	54,7	65,3	50,3	61,8	50,6	50,3	58,7	59,1	72,7
Class5 (%)	52	60,1	61,1	45,8	68,3	33,7	41,6	51	66,8	76,1
Class6 (%)	70,2	63,2	69	66,2	74,2	69,4	54,7	68,7	69,2	79,9
Class7 (%)	62,2	71,8	75,2	44,3	71,8	43,5	64,6	64,5	70,2	75,1
Class1_err (%)	8,9	12,5	14,3	17,9	7,1	19,6	16,1	14,3	12,5	12,5
Class2_err (%)	14	17,4	10,5	12,8	16,3	20,9	19,8	10,5	15,1	10,5
Class3_err (%)	15,1	18,9	17	20,8	13,2	22,6	15,1	15,1	15,1	11,3
Class4_err (%)	14,3	8,6	11,4	14,3	15,7	12,9	15,7	12,9	14,3	7,1
Class5_err (%)	17,7	14,5	17,7	17,7	11,3	14,5	22,6	21	9,7	16,1
Class6_err (%)	10,4	14,6	8,3	16,7	12,5	9,4	11,5	9,4	9,4	10,4
Class7_err (%)	20,8	18,9	13,2	18,9	24,5	13,2	26,4	17	9,4	18,9
Class_avg (%)	65,9	63,6	70,8	59,4	67,3	50,9	50,6	64,6	64,5	74,6
Cl_err_avg (%)	14,5	15,1	13,2	17	14,4	16,2	18,2	14,3	12,2	12,4

Examining the recall rates of indexes the best features for classes can be observed. For Class 1 DCD performs best since “sunset” image can be separated from others using the dominant colors. For Class 2 and 3 CSD performs best since “mountain” and “waterfall” image can be separated from others using the color and spatial information. For the other classes DHC feature is the best since “autumn”, “flower”, “building” and “sea” images have strong edge and texture patterns. A color-texture combination feature results better with those. From the observations it is clear that a color-texture combination descriptor is necessary in CBIIR applications on color images. This can be generally obtained by combining the descriptors.

To observe the effect of combined features descriptors are combined and tests are held. Results are given in Figure 5.9 and Table 5.12. In the table, the results are given for the descriptor combinations that are given in the heading row and the performance measures that are given in the heading column. Examining the results it can be seen that the indexing performance increases with combined features. The combined features are given in the graphs. When the descriptor sets of MPEG-7 is compared with the feature vector developed in this thesis it is seen that the feature vector developed is better in Success, and MSE. Inspecting the average class recall rates the feature vector developed is better. For sunset dominant color combination is the best, for the waterfall and sea classes texture combination is important with color lay out, for the mountain class color structure is the best, for flower multiresolution is important and for the autumn-fall and building classes color structure is important.

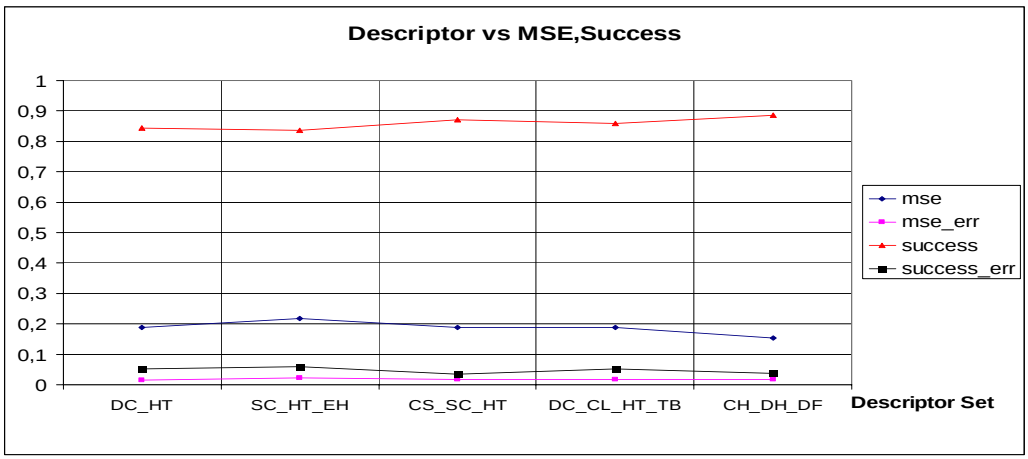

Figure 5.9 Descriptor Set vs. MSE and Descriptor Set vs. Success

Table 5.12 Performance comparison for descriptor sets

	DCD - HTD	**SCD-HTD-EHD**	**CSD-SCD - HTD**	**DCD-CLD-HTD-TBD**	**CHC-DHC-DF**
mse	0,188	0,218	0,189	0,187	0,153
mse_err	0,016	0,022	0,018	0,018	0,018
Success (%)	84,4	83,6	87,2	85,9	88,7
Success_err (%)	5,3	5,9	3,4	5,3	3,6
Class1 (%)	93,4	90,2	90	94,6	90,2
Class2 (%)	70,3	68,5	78	73	69,7
Class3 (%)	73,5	70	77,5	87,3	75,6
Class4 (%)	71,7	70,6	74,6	73,6	76,1
Class5 (%)	65,1	70,3	69,5	57,9	79,2
Class6 (%)	74,6	73,6	77,8	82,2	80,2
Class7 (%)	72,6	80,1	80,5	70,5	82,1
Class1_err (%)	10,7	14,3	8,9	8,9	12,5
Class2_err (%)	8,1	10,5	10,5	11,6	10,5
Class3_err (%)	17	11,3	13,2	9,4	17
Class4_err (%)	14,3	14,3	11,4	11,4	11,4
Class5_err (%)	12,9	14,5	16,1	19,4	12,9
Class6_err (%)	11,5	12,5	8,3	10,4	10,4
Class7_err (%)	15,1	11,3	13,2	18,9	11,3
Class_avg (%)	74,5	74,8	78,3	77	79
Cl_err_avg (%)	12,8	12,7	11,7	12,9	12,3

5.6 EXAMPLE IMAGES INDEXED USING THE ALGORITHM PROPOSED

In this section examples of indexed images using the algorithm proposed are given. In Figure 5.10, Figure 5.11, Figure 5.12, Figure 5.13, Figure 5.14, Figure 5.15 and Figure 5.16 indexed images are given for the classes sunset, mountain, waterfall-river, autumn-forest, buildings, sea and flowers correspondingly. 16 images that are indexed to the mentioned classes are given. Examples of images that indexed to multiple contents of sunset and sea, buildings and sea, buildings and autumn-forest, mountain and autumn-forest, mountain and waterfall, mountain and sea are given Figure 5.17, Figure 5.18, Figure 5.19, Figure 5.20, Figure 5.21 and Figure 5.22 consecutively.

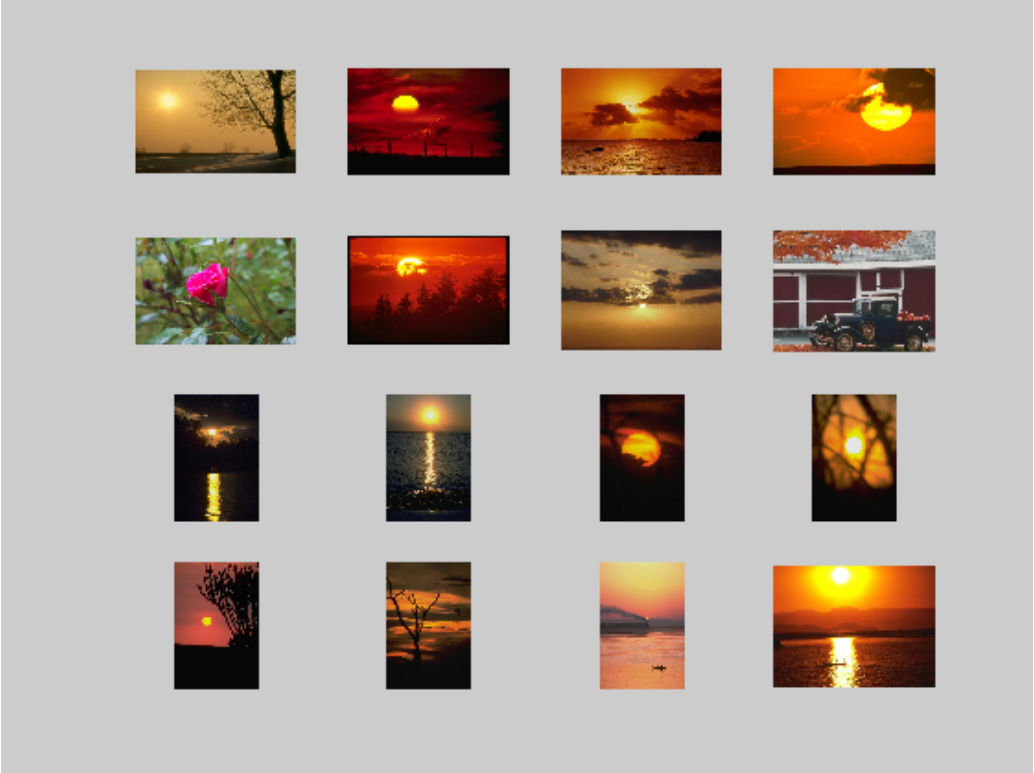

Figure 5.10 Sunset Images

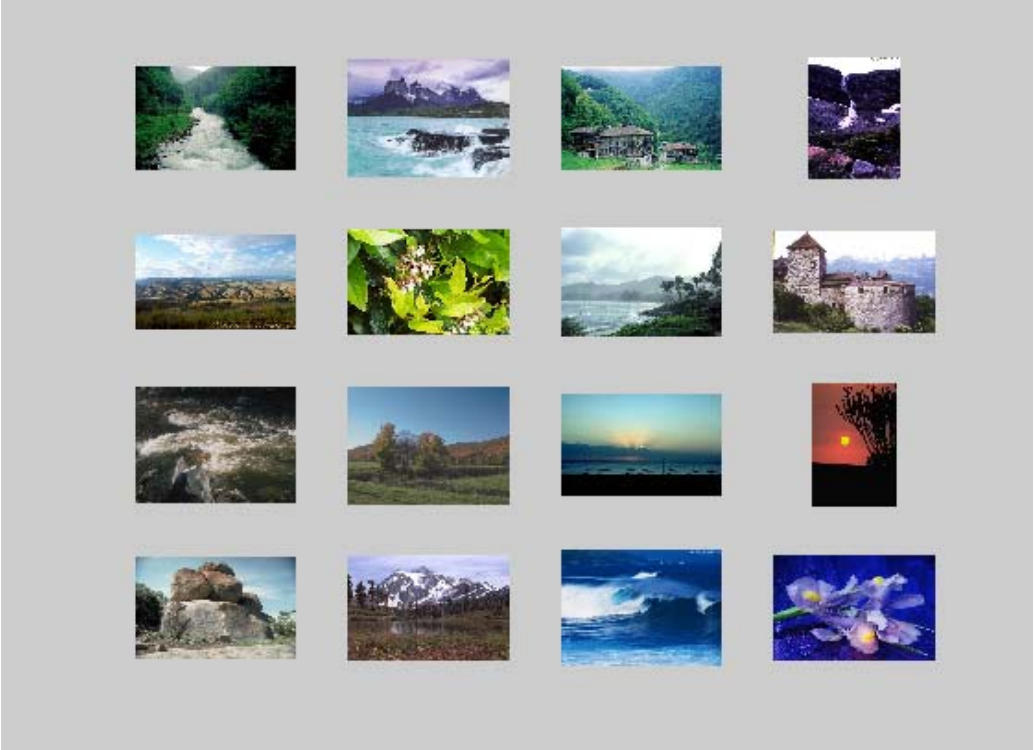

Figure 5.11 Mountain Images

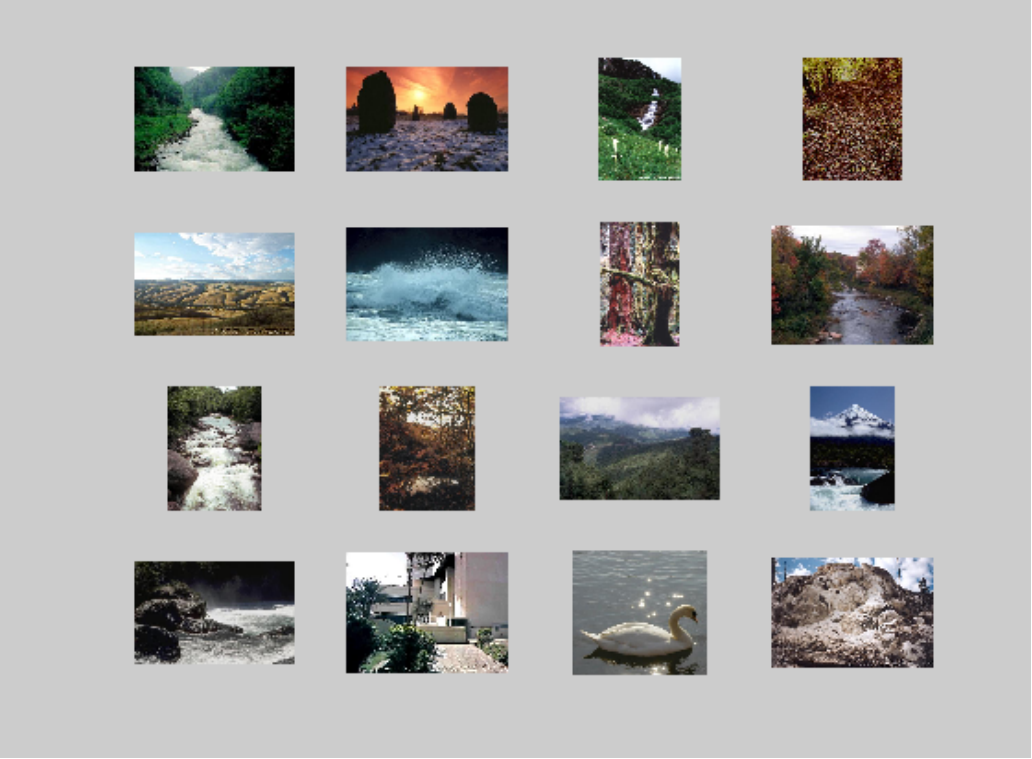

Figure 5.12 Waterfall-River Images

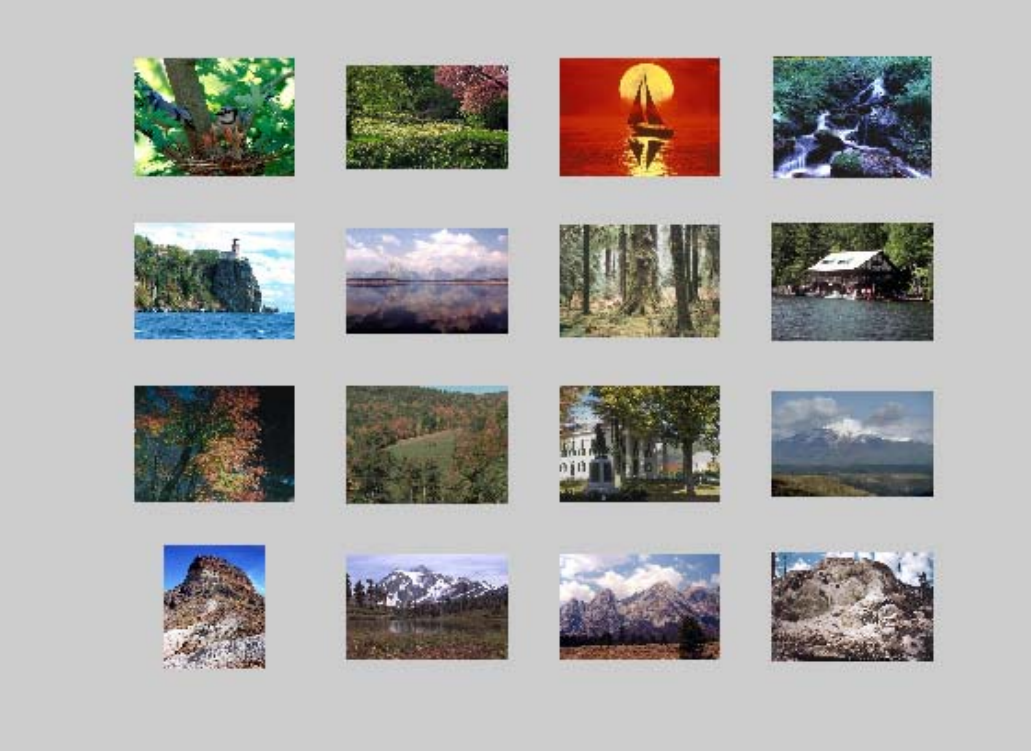

Figure 5.13 Autumn-Forest Images

Figure 5.14 Building Images

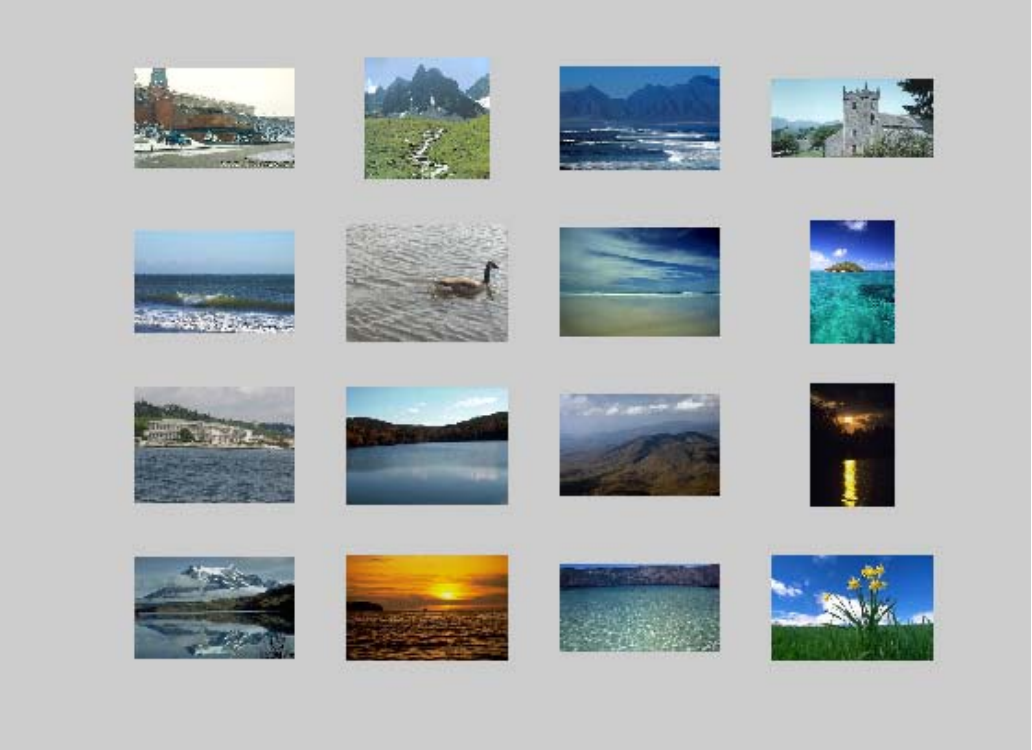

Figure 5.15 Sea Images

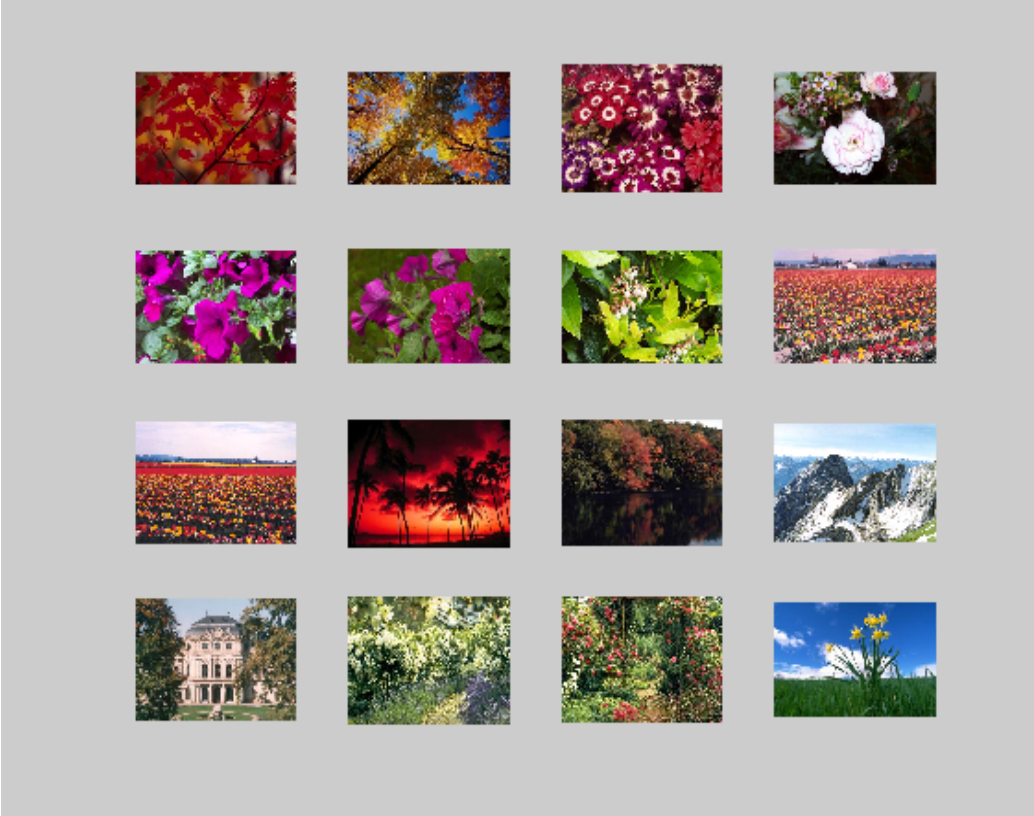

Figure 5.16 Flower Images

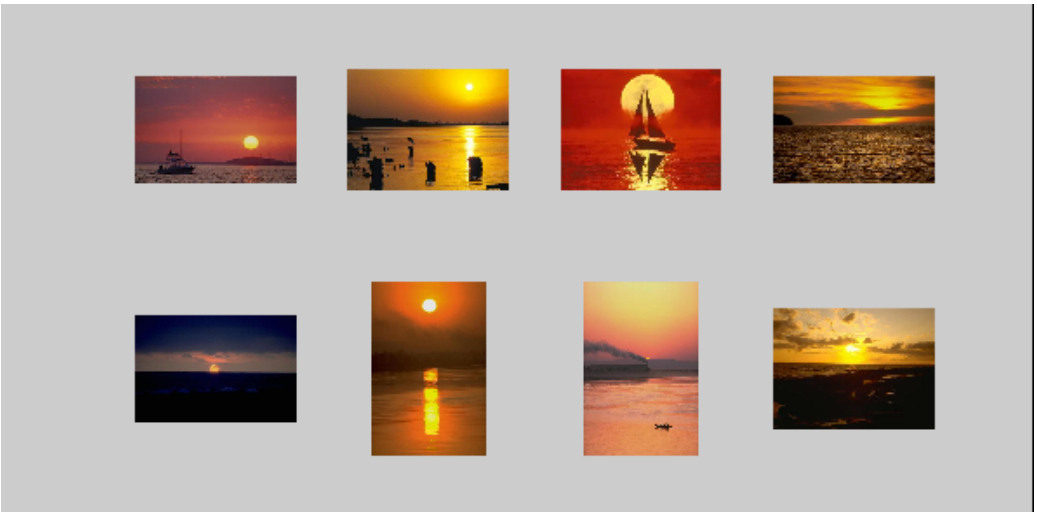

Figure 5.17 Sunset and Sea Images

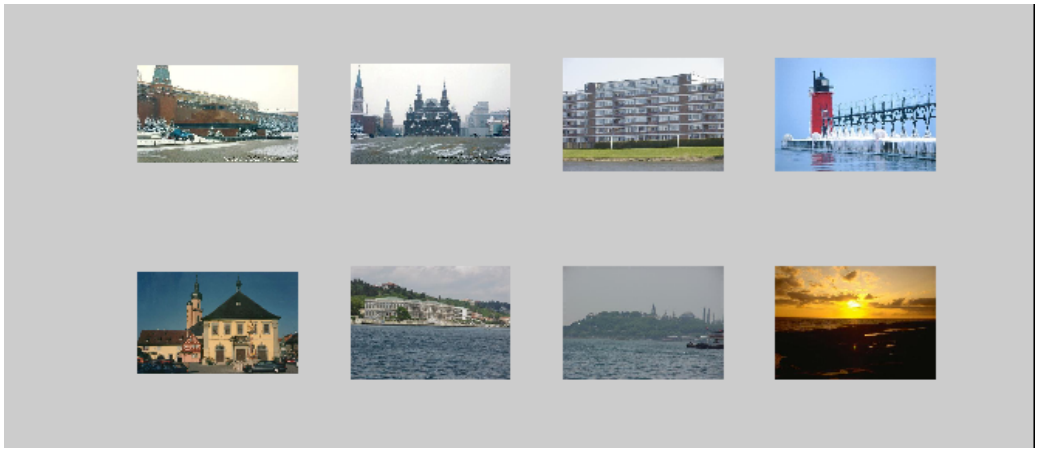

Figure 5.18 Building and Sea Images

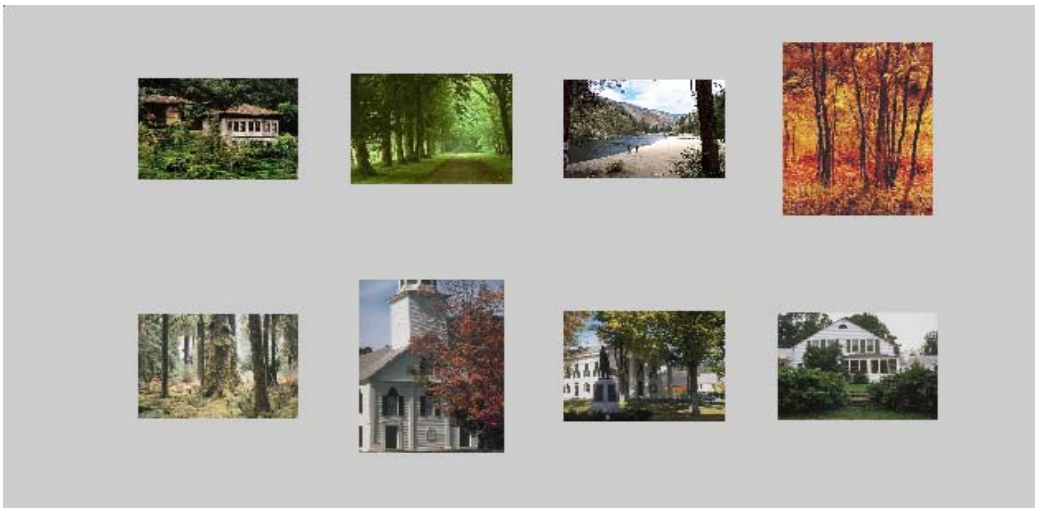

Figure 5.19 Building and Autumn-Forest Images

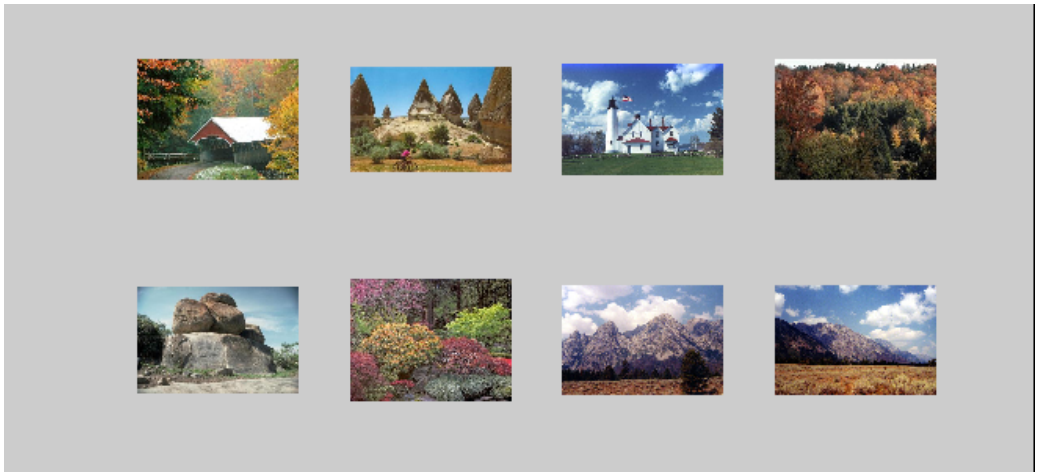

Figure 5.20 Mountain and Fall Images

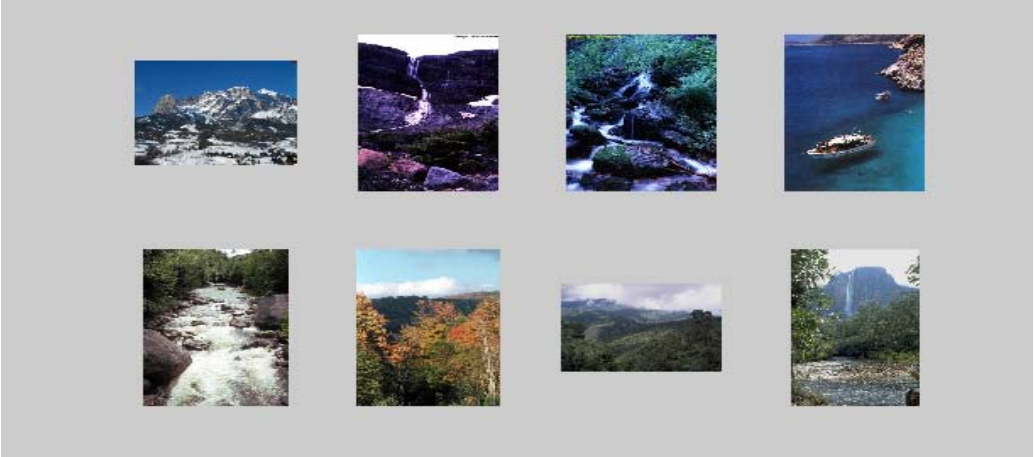

Figure 5.21 Mountain and Waterfall Images

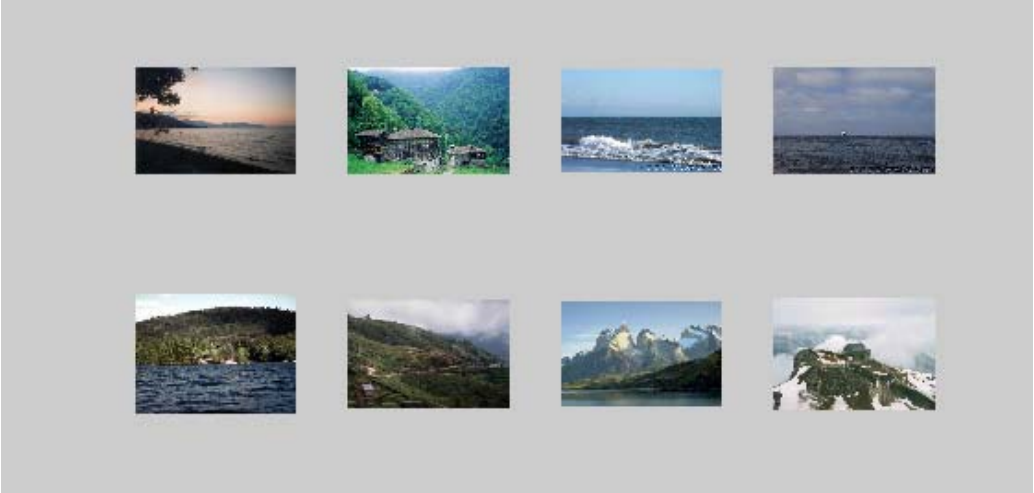

Figure 5.22 Mountain and Sea Images

CHAPTER 6

CONCLUSIONS

Image databases provide a good source of information, but its value depends on the efficient retrieval of the information, so that efficient storing and retrieval systems are required for the data management. Automatic retrieval is important in efficiently reaching the data. Content based image indexing and retrieval is the automatic retrieval of images from databases. Image classification is a key to CBIIR.

In this thesis, an algorithm for image classification is developed for content based indexing. An artificial neural network is designed to classify the images based on their content. Features are developed to represent the images for CBIIR. The database is traced and classified into seven classes. These classes are buildings, flowers, mountains, sea, river-waterfall, sunset and autumn-forest. The ground truth information is formed in a supervised fashion to train and to test the system with the multiple contents enabled. The performance of the system is tested using leave one out method. The performances of features are compared with the performance of MPEG-7 color and texture descriptors.

A feed forward back propagation neural network is used as the NNET classifier. The network is tested using leave one out method. The output of the NNET simulation for an image indicates the content classes that the image belongs to. The tests yielded a threshold to merit the output for multiple outputs.

Color histogram correlations, detail histogram correlations and directional filtering in frequency domain are used to extract features from the images. The content of the image is carried by its color, texture and spatial information. Color histogram correlations are used to represent the color information. Luminance

Chrominance color space is used in obtaining the color histograms. The correlation features extracted are the moment positions and energy distributions of the correlation curves of the image histograms. Luminance Chrominance color space is compared with RGB and HSV, and found to be better since it is less sensitive to brightness change which is not a key feature in image semantics.

To extract the content of the image, texture and spatial information in the image is important. A color texture combination feature is derived using the frequency domain representation of the image over color spaces and calculating the energy over frequency channels. The number of frequency channels is found to be 14 with the color space RGB for the optimum performance. Wavelet transform and its properties are used to extract the edge information in the image. The detail images are formed using 2D DWT. Histograms of detail images give the density of coefficients which carry the information of horizontal, vertical and diagonal edges. Using the correlations among these, energy distribution and variance of the cross correlation curves are used as features. This multiresolution analysis is tested with different levels of resolution. It's observed that the performance increases with increasing the level, with a saturation as well. As an optimum point, 3 level multiresolution is used.

The mentioned features are combined to form the feature vector. PCA analysis is used over the training database to decrease the vector length and increase NNET performance. The performance of the system is calculated using the success which is the ratio of finding at least one correct index for the image, the MSE which is the mean square error between the ground truth data and the indexing output for the image, and the class recall which is the ratio of relevant retrieved data to the relevant data in the database. The tests give the results as: the average MSE is 0.15, the success is 89%, sunset recall is 90%, mountains class recall is 70%, river-waterfall class recall is 76%, autumn-fall class recall is 76%, buildings class recall is 79%, sea class recall is 80% and flowers class recall is 82%. The tests show that the combination of color and texture levels gives a better performance than the features used alone.

The performances of the derived features are compared with the performances of the MPEG-7 color and texture descriptors. The results are found to

be comparable and better. From the tests it is observed that the proposed algorithm and the developed feature vector are successful in classification of images for content based indexing. These findings suggest that this method of classification for content based image indexing is a reliable and valid method for content based image indexing and retrieval, especially in scenery image indexing. This method can also be used in many areas a few examples are medical diagnosis in medical imaging, classification of buildings in architecture history or the indexing of photograph collections.

The future work on this topic would be the improvement of the structure. The performance of the classification structure can be improved and tested on larger image databases. Additional content classes may be added to index larger image databases with more available content such as court, airport, space and etc. Shape descriptors can be used with the color and texture features as well as the image segmentation to index the objects in the images for CBIIR. The effect of feature size and the effect of the image size on Content Based Image Indexing and Retrieval should be investigated. In this thesis one network is constructed for multiple outputs; another approach could be using one network for each class and combining the results. This could be compared with the output of one network.

REFERENCES

[1] Bashir F, et al. "Multimedia Systems: Content Based Indexing and Retrieval", University of Illinois, 2003

[2] Eakins J P and Graham M E "Content Based Image Retrieval" JTAP Report No:39, 5(2), 1999

[3] Wang J Z, et al, "Content Based Image Indexing and Searching using Daubechies' Wavelets", International Journal ON Digital Libraries, vol. 1, number 1, Springer-Verlag, 1997

[4] Lim J S, "Two Dimensional Signal and Image Processing", Prentice Hall, 1990

[5] Vailaya A, et al, "Image Classification for Content Based-Indexing", IEEE Transactions on Image Processing, Vol. 10, number 1, January 2001

[6] Goswami J C and Chan A K, "Fundamentals of Wavelets Theory, Algorithms, and Applications", Wiley-Interscience, 1999

[7] Missiti M, Missiti Y, Oppenheim G, Poggi J M, "Wavelet Toolbox For Use with MATLAB", Version 2, Mathworks, 2000

[8] Kosko B, "Neural Networks for Signal Processing", Prentice Hall, 1992

[9] Demuth H, Beale M, "Neural Network Toolbox For Use with MATLAB", Version 4, Mathworks, 2000

[10] Smeulders A.W.M, Worring M, Santini S., Gupta A., R. Jain, "Content-Based Image Retrieval at the End of the Early Years", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22, number 12, December 2000

[11] Huang J., Kumar S., Mitra M., Zhu W., Zabih R., "Spatial Color Indexing and Applications**",** International Journal of Computer Vision**,** Volume 35, Issue 3, December 1999

[12] Vailaya A., "Semantic Classification in Image Databases", Ph.D. Thesis, Michigan State University, 2000

[13] Torralba A., Oliva A., "Semantic Organization of Scenes using Discriminant Structural Templates", Proceedings of International Conference on Computer Vision, ICCV99, pp. 1253-1258, Korfu, Greece, September 1999

[14] Ratan A., Maron O., Grimson W., Lozano-Pérez T., "A Framework for Learning Query Concepts in Image Classification", Proceedings of International Conference on Computer Vision and Pattern Recognition, ICCVP99, volume 1, pp. 423-429, Fort Collins, Colorado, June 1999

[15] Yu H., Wolf W., "Scenic classification methods for image and video databases", SPIE International Conference on Digital Image Storage and Archiving Systems, SPIE 2606, pages 363-371, 1995

[16] Xiong X., "Image Classification for Image Database Organization in a Content-Based Image Retrieval System", Ph.D. Thesis, Nanyang Technological University, 2002

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

[18] Niblack W., Zhu X., Hafner J., Breuel T., Ponceleon D., Petkovic D., Flickner M, Upfal E., Nin S., Sull S., Dom B., Yeo B., Srinivasan S., Zivkovic D., and Penner M., "Updates to the QBIC system", Proceedings of SPIE, Storage and Retrieval for Image and Video Databases VI, volume 33 number 12, Jan. 1998

[19] Carson C., Thomas M., Belongie S., Hellerstein J., and Malik J., "BlobWorld: A system for region-based image indexing and retrieval,", Third International Conference on Visual Information Systems, Amsterdam, Netherlands, pp. 509-516, Springer, 1999

[20] Ma W.-Y. and Manjunath B. S., "NeTra: A toolbox for navigating large image databases,", Proceedings of ICIP'97, Vol. 1, pp. 568-571, Santa Barbara, CA, 1997

[21] Pentland A., Picard R. W., and Sclaroff S., "PhotoBook: Content-based manipulation of image databases,", Proc. SPIE Storage and Retrieval for Image and Video Databases, pp. 34-47, San Jose, CA 1994

[22] Smith J. and Chang S.-F., "VisualSEEK: a fully automated content-based image query system" ACM International Conference on Multimedia, pages 87-98, Boston, MA, November 1996

[23] http://www.virage.com

[24] Martinez J., "Overview of the MPEG-7 Standard", ISO/IEC JTC1/SC29/WG11 N4031, Version 5.0, March 2001

[25] Manjuath B. S., Salembier P., Sikora T., "Introduction to MPEG-7 Multimedia Content Description Interface", Wiley, 2002

[26] Yamada A., Pickering M, Jeannin S., Cieplinski L., Rainer Ohm J., Kim M., "MPEG-7 Visual part of experimentation Model", ISO/IEC JTC1/SC29/WG11 N4063, Version 10.0, March 2001

[27] Manjuath B.S., Ohm J-R, Vasudevan V., and Yamada A., "Color and Texture Descriptors", IEEE Transactions on Circuits and Systems for Video Technology" , Vol. 11, number 6, June 2001

[28] Deng Y., Manjuath B.S., et al., "An Efficient Color Representation for Image Retrieval", IEEE Transactions on Image Processing, Vol. 10, number 1, January 2001

[29] Deng Y., Kenney C., et al., "Peer Group Filtering and Perceptual Color Image Quantization", Proc. IEEE Int. Symposium Circuits and Systems, pp. 21-24 vol. 4, June 1999

[30] Manjuath B. S. and Ma W. Y., "Texture Features for Browsing and Retrieval of Image Data", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 8, pp. 837-842, 1996

[31] Jain A. K., "Fundamentals of Image Processing", Prentice Hall, 1989

[32] ISO/IEC 15938-3:2001, "Multimedia Content Description Interface - Part 3: Visual", Version 1

[33] Liu F. and Picard R.W., “Periodicity, Directionality and Randomness: Wold Features for Image Modeling and Retrieval”, IEEE Trans. on PAMI, Vol. 18, No. 7, pp. 722-733, July 1996

[34] Wu P., et al., “A Texture Descriptor for Browsing and Similarity Retrieval.” Journal of Signal Processing: Image Communication, 16(1), pp. 3343, September 2000

[35] Park D. W., Jeon Y-S., Won C. S., Park S.-J., “Efficient use of Local Edge Histogram Descriptor”, The Proceedings of ACM Multimedia Workshop, ACM Multimedia 2000, Nov. 2000

[36] Stricker M. and Orengo M., “Similarity of color images”, In Storage and Retrieval for Image and VideoDatabases III (SPIE), volume 2420 of SPIE Proceedings Series, pp. 381–392, San Jose, CA, USA, February 1995.

[37] Smith J., “Image retrieval evaluation”, In IEEE Workshop on Content-based Access of Image and Video Libraries (CBAIVL'98), pp. 112-113,Santa Barbara, CA, June 1998

APPENDIX A

A.1 FEATURE VECTORS OF THE EXAPLE IMAGES

The feature vectors for the Example Images in Chapter 3 are given in Table A.1. The first 18 features are derived from CHC, features 19..54 are derived from DHC and the rest is derived from directional filtering.

Table A.1 Feature Vectors of the Example Images

Feature no:	Example Image 1	Example Image 2	Example Image 3	Example Image 4	Example Image 5	Example Image 6
1	0,4736	0,3796	0,4697	0,4892	0,4736	0,2661
2	0,184	0,1487	0,1096	0,0744	0,0705	0,0783
3	0,1761	0,1096	0,1018	0,2622	0,0509	0,0978
4	0,593	0,4755	0,3953	0,4286	0,4168	0,5049
5	0,6164	0,4305	0,3425	0,2994	0,2857	0,5049
6	0,3464	0,2838	0,3072	0,3307	0,3327	0,2114
7	0,4932	0,4736	0,3757	0,3699	0,4403	0,544
8	0,5362	0,4599	0,3033	0,3014	0,2857	0,5479
9	0,3346	0,2877	0,3053	0,3953	0,3288	0,2114
10	0,407	0,5049	0,4834	0,454	0,5225	0,5401
11	0,4325	0,5303	0,4932	0,501	0,5186	0,5558
12	0,1683	0,1233	0,1037	0,1722	0,0568	0,0978
13	0,0454	0,0574	0,0411	0,0356	0,1625	0,1064
14	0,1592	0,2391	0,2884	0,4408	11.346	0,6455
15	0,1816	0,3181	0,3406	0,104	15.019	0,8525
16	0,0761	0,0898	0,0849	0,074	0,2962	0,1884
17	0,0802	0,095	0,0865	0,0549	0,3459	0,2037
18	0,1687	0,254	0,3037	0,1703	10.299	0,6732
19	0,137	0,0861	0,1331	0,1057	0,137	0,1096
20	0,247	0,2422	0,1721	0,2423	0,508	0,3657
21	0,0939	0,1018	0,1487	0,1292	0,0861	0,0665
22	0,262	0,2717	0,162	0,2074	0,6003	0,427
23	0,0391	0,0352	0,0548	0,0431	0,047	0,0313
24	0,4843	0,5172	0,2675	0,4023	0,8011	0,7322
25	0,1155	0,0939	0,1409	0,1194	0,1115	0,0861

Table A.1 (continued) Feature Vectors of the Example Images

Feature no:	Example Image 1	Example Image 2	Example Image 3	Example Image 4	Example Image 5	Example Image 6
26	0,254	0,2562	0,167	0,2241	0,5522	0,3944
27	0,092	0,0607	0,0939	0,0763	0,092	0,0705
28	0,3439	0,3503	0,2111	0,3106	0,6375	0,516
29	0,0665	0,0705	0,1018	0,0881	0,0665	0,0489
30	0,3541	0,3728	0,2048	0,2872	0,6933	0,5574
31	0,3092	0,2505	0,4697	0,4658	0,4658	0,4501
32	0,1616	0,1691	0,1336	0,1338	0,3225	0,1895
33	0,2153	0,2544	0,4853	0,5049	0,2975	0,2387
34	0,1871	0,1687	0,1259	0,1316	0,4267	0,2624
35	0,1292	0,137	0,2466	0,2348	0,2035	0,1174
36	0,2103	0,2155	0,1453	0,1741	0,5302	0,3911
37	0,2622	0,2505	0,4775	0,4873	0,3816	0,3425
38	0,1737	0,1688	0,1297	0,1326	0,3706	0,2226
39	0,2192	0,1937	0,364	0,3562	0,3386	0,2896
40	0,1832	0,1896	0,1389	0,1523	0,4124	0,2714
41	0,1722	0,1977	0,3699	0,3777	0,2524	0,182
42	0,1978	0,19	0,1348	0,1508	0,4752	0,32
43	0,7084	0,6341	0,9237	0,9863	0,9198	0,9824
44	0,1159	0,1626	0,1469	0,1831	0,3347	0,1997
45	0,6849	0,4384	0,9237	0,998	0,681	0,4501
46	0,1319	0,1556	0,1566	0,1684	0,3996	0,2364
47	0,3014	0,3288	0,6967	0,6928	0,5675	0,3053
48	0,1708	0,158	0,1249	0,1275	0,4574	0,3456
49	0,6986	0,5382	0,9256	0,9824	0,8102	0,6986
50	0,1233	0,1587	0,1515	0,1747	0,3647	0,214
51	0,5205	0,4892	0,8063	0,8258	0,7515	0,6341
52	0,1395	0,1594	0,134	0,1512	0,3891	0,2576
53	0,5108	0,3836	0,8023	0,8258	0,6262	0,3796
54	0,1491	0,1565	0,1383	0,1427	0,4268	0,285
55	0,3641	0,327	0,3259	0,2988	0,3446	0,3544
56	0,3059	0,2842	0,2881	0,2577	0,3118	0,3977
57	0,2806	0,2896	0,2915	0,2677	0,3094	0,3421
58	0,2792	0,3021	0,3	0,2865	0,32	0,3092
59	0,2922	0,3175	0,3139	0,3033	0,333	0,3273
60	0,2961	0,3333	0,3135	0,3186	0,3204	0,3593
61	0,3511	0,3322	0,3384	0,3379	0,3312	0,3457
62	0,3629	0,3014	0,3142	0,2882	0,306	0,3417
63	0,3044	0,2626	0,279	0,2536	0,3127	0,335
64	0,2805	0,26	0,2818	0,2608	0,3311	0,3222
65	0,27	0,2741	0,2924	0,2769	0,343	0,31
66	0,2881	0,2801	0,3051	0,2973	0,338	0,2982
67	0,2975	0,292	0,3131	0,3149	0,32	0,3023
68	0,3538	0,3246	0,3348	0,3367	0,327	0,336

Table A.1 (continued) Feature Vectors of the Example Images

Feature no:	Example Image 1	Example Image 2	Example Image 3	Example Image 4	Example Image 5	Example Image 6
69	0,3293	0,3988	0,4052	0,3508	0,388	0,39
70	0,2701	0,3612	0,3907	0,3148	0,3518	0,3997
71	0,2441	0,3687	0,3973	0,3275	0,35	0,3721
72	0,2409	0,3799	0,4075	0,3502	0,3612	0,3329
73	0,2509	0,3974	0,4214	0,3631	0,3721	0,3337
74	0,256	0,4098	0,4224	0,3827	0,3643	0,3627
75	0,3116	0,3994	0,4386	0,4009	0,3751	0,3679
76	0,3274	0,377	0,3991	0,3368	0,3469	0,3792
77	0,2678	0,3412	0,3854	0,3139	0,3539	0,37
78	0,2439	0,3378	0,3901	0,3233	0,3705	0,3544
79	0,2345	0,3527	0,4044	0,3439	0,3833	0,3425
80	0,2455	0,3558	0,4141	0,3609	0,3808	0,3427
81	0,2582	0,3649	0,421	0,3829	0,3617	0,3329
82	0,315	0,3949	0,4406	0,401	0,3713	0,358
83	0,3517	0,3353	0,402	0,3418	0,3544	0,2886
84	0,2858	0,293	0,371	0,3015	0,3116	0,2922
85	0,259	0,3009	0,3741	0,3123	0,3113	0,267
86	0,2529	0,3103	0,3828	0,3352	0,3235	0,2363
87	0,2589	0,3236	0,4008	0,3484	0,332	0,2379
88	0,2688	0,3302	0,4025	0,3672	0,3276	0,2472
89	0,3305	0,3262	0,4252	0,3933	0,3387	0,2529
90	0,3417	0,3137	0,3964	0,3268	0,3094	0,2759
91	0,2859	0,2787	0,3618	0,3008	0,3147	0,2671
92	0,255	0,276	0,3684	0,3073	0,3306	0,2549
93	0,2453	0,2876	0,3796	0,3252	0,3434	0,2404
94	0,2518	0,2921	0,3913	0,3467	0,3421	0,237
95	0,2725	0,2946	0,3986	0,3661	0,3257	0,2214
96	0,3311	0,3244	0,4228	0,392	0,3343	0,237