OBJECT EXTRACTION FROM IMAGES/VIDEOS USING A GENETIC ALGORITHM BASED APPROACH

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

BY

TURGAY YILMAZ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JANUARY 2008

Approval of the thesis

"OBJECT EXTRACTION FROM IMAGES/VIDEOS USING A GENETIC ALGORITHM BASED APPROACH"

submitted by **Turgay Yılmaz** in partial fullfillment of the requirements for the degree of **Master of Science in Computer Engineering** by,

Prof. Dr. Canan Özgen
Dean, **Graduate School of Natural and Applied Sciences**

Prof. Dr. Volkan Atalay
Head of Department, **Computer Engineering**

Prof. Dr. Adnan Yazıcı
Supervisor, **Computer Engineering, METU**

Examining Committee Members:

Prof. Dr. Faruk Polat
Computer Engineering, METU

Prof. Dr. Adnan Yazıcı
Computer Engineering, METU

Asst. Prof. Dr. Tolga Can
Computer Engineering, METU

Dr. Onur Tolga Şehitoğlu
Computer Engineering, METU

M.S. Yakup Yıldırım
Havelsan A.Ş.

Date:

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

Name, Last name	:	Turgay Yılmaz
Signature	:	

ABSTRACT

OBJECT EXTRACTION FROM IMAGES/VIDEOS USING A GENETIC ALGORITHM BASED APPROACH

Yılmaz, Turgay
M.S., Department of Computer Engineering
Supervisor: Prof. Dr. Adnan Yazıcı

January 2008, 104 pages

The increase in the use of digital video/image has showed the need for modeling and querying the semantic content in them. Using manual annotation techniques for defining the semantic content is both costly in time and have limitations on querying capabilities. So, the need for content based information retrieval in multimedia domain is to extract the semantic content in an automatic way. The semantic content is usually defined with the objects in images/videos. In this thesis, a Genetic Algorithm based object extraction and classification mechanism is proposed for extracting the content of the videos and images. The object extraction is defined as a classification problem and a Genetic Algorithm based classifier is proposed for classification. Candidate objects are extracted from videos/images by using Normalized-cut segmentation and sent to the classifier for classification. Objects are defined with the Best Representative and Discriminative Feature (BRDF) model, where features are MPEG-7 descriptors. The decisions of the classifier are calculated by using these features and BRDF model. The classifier improves itself in time, with the genetic operations of GA. In addition to these, the system supports fuzziness by making multiple categorization and giving fuzzy decisions on the objects. Externally from the base model, a statistical feature importance determination method is proposed to generate BRDF model of the categories automatically. In the thesis, a platform independent application for the proposed system is also implemented.

Keywords: Object extraction, genetic algorithms, normalized-cut image segmentation, representative feature, discriminative feature, MPEG-7 descriptors

ÖZ

İMGE VE VİDEOLARDAN GENETİK ALGORİTMA YAKLAŞIMIYLA NESNE ÇIKARILMASI

Yılmaz, Turgay
Yüksek Lisans, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Adnan Yazıcı

Ocak 2007, 104 sayfa

Sayısal video ve imgelerin kullanımındaki artış, video ve imgelerin mantıksal içeriğine göre modellenmesi ve sorgulanabilmesi ihtiyacını ortaya çıkarmıştır. Mantıksal içeriği elle notlar ekleme yöntemiyle tanımlamak, hem çok zaman gerektirmektedir hem de sorgulama kabiliyetlerini kısıtlamaktadır. Öyleyse çoklu ortam alanında kullanılan içeriğe dayalı bilgi kazanım sistemleri için ihtiyaç, mantıksal içeriğin otomatik bir şekilde çıkarılmasıdır. Mantıksal içerik genellikle imge ve videolarda gözüken nesneler yardımıyla tanımlanır. Bu tezde, video ve imgelerin mantıksal içeriğinin elde edilebilmesi için, Genetic Algoritma temelli bir nesne çıkarma ve sınıflandırma mekanizması önerilmiştir. Nesne çıkarımı ise bir sınıflandırma problemi olarak tanımlanmıştır ve sınıflandırma için Genetik Algoritma temelli bir sınıflandırıcı önerilmiştir. Düzgelenmiş kesimle imge bölütleme kullanılarak olası nesneler video ve imgelerden çıkarılmış ve sınıflandırıcıya sınıflandırılmak üzere gönderilmektedir. Nesneler, En İyi Temsili ve Ayrıştırıcı Öznitelik modeli ile tanımlanmaktadır. Burada kullanılan öznitelikler, MPEG-7 betimleyicileridir. Sınıflandırıcının kararı, bu öznitelikler ve bahsedilen model yordamıyla hesaplanmaktadır. Sınıflandırıcı, Genetik Algoritma içerisindeki genetik işlemler yardımıyla, zaman içerisinde kendisini geliştirmektedir. Bunlara ek olarak sistem, çoklu sınıflandırma yapması ve nesneler üzerinde bulanık kararlar vermesi ile bulanıklılığı desteklemektedir. Temel modelden hariç olarak, nesne sınıflarının En İyi Temsili ve Ayrıştırıcı Özniteliklerini otomatik olarak üretebilmek için istatistiksel bir öznitelik önem tespit metodu önerilmiştir. Bu tezde, ayrıca, önerilen sistem için platform bağımsız çalışabilen bir uygulama geliştirilmiştir.

Anahtar Kelimeler: Nesne çıkarımı, genetik algoritma, düzgelenmiş kesimle imge bölütleme, temsili özellik, ayrıştırıcı özellik, MPEG-7 betimleyicileri

ACKNOWLEDGMENTS

I would like to express my inmost gratitude to my supervisor Prof. Dr. Adnan Yazıcı for his guidance, advice, insight throughout the research and trust on me. It is an honour for me to share his knowledge and wisdom. Also I am grateful to my voluntary co-supervisor, fellow and manager Yakup Yıldırım. Without his assistance, encouragement, tolerance and patience; this thesis could not be accomplished.

I am indebted to my parents and younger brother for all their support and self-sacrifice on my behalf.

I am also thankful to Ertuğrul Kartal Tabak, Göker Canıtezer, Eren Şenelmiş and Emre Erdoğan for their assistance and contribution whenever I need their expertise; my company Havelsan Inc., managers and colleagues for their toleration during my research; my friend Gökhan Gürgeç for all his support during my thesis writing process.

Finally, I would like to thank to everyone that has an effort on me, during all my life.

To my family

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURES

LIST OF TABLES

TABLES

LIST OF ABBREVIATIONS

ANMRR	Average Normalized Modified Retrieval Rank
BRDF	Best Representative and Discriminative Feature
BRF	Best Representative Feature
CBIR	Content Based Information Retrieval
CSS	Curvature Scale Shape
DDL	Description Definition Language
Ds	Descriptors
DSs	Description Schemes
GA	Genetic Algorithm
GUI	Graphical User Interface
HMMD	Hue, Max, Min, Diff
HSV	Hue, Saturation, Value
ISO	International Organization for Standardization
IEC	International Electrotechnical Commission
JNI	Java Native Interface
MPEG	Moving Picture Experts Group
Ncut	Normalized Cut (Image Segmentation)
OVDAM	Ontology-Supported Video Database Model
PNG	Portable Network Graphics
XM	eXperimentation Model (MPEG-7 Reference Software)

CHAPTER 1

INTRODUCTION

Digital video gets more importance in education, entertainment, security or any other multimedia applications. The increase in the use of digital video has showed the need for modeling and querying the digital video. Free-browsing (browsing through a collection of video files until finding the desired information) and text-based retrieval of previously annotated video data (adding textual information metadata manually to the multimedia files during a cataloguing phase and making text-based queries) are not enough due to the limitations for querying. Therefore, developing techniques for efficiently querying videos on their content has attracted many researchers [12] [57]. The content of a video can be basically defined as the objects and the interaction of the objects. So, extracting the objects in videos and finding their categories in videos gives a big support to content based retrieval job.

There are many studies in the literature about object identification in videos. In [8], Cavallaro et al. classify object extraction strategies in three classes: Manual, fully automatic and semi automatic extraction. Manual extraction methods define the objects in each video manually. Fully automatic extraction methods use special characteristics of scenes (background removing) or specific information with defined algorithms (template matching, face recognition, moving object segmentation). In these methods, the low level features of the images (color, texture, etc.) are directly used in proposed algorithms. Semi automatic extraction methods contain both manual and automatic parts. These methods can be supervised and interactive. In some part of those systems, the users define some information about the objects (selecting a pixel, region, texture, etc.) and then the system tries to find objects according to previously learned data. In fact, semi automatic and fully automatic systems do not differ so much; only difference between them is that semi automatic systems require the rules of extraction to be defined by the user while the fully automatic systems have the rules defined in the system by itself.

If we consider large number of videos, manual extraction of objects is not an efficient mechanism. So, the need of CBIR systems is to extract objects in an automatic way and find their categories. These two needs can be handled one by one or together. The researches usually focus on the two requeirments together; and try to find the answer with pattern recognition and pattern classification approaches [12] [57]. These researches propose ready-to-use (the classification rules are defined by itself) classifiers for specific problem domains. Although these methods have good performances at their domain specific problems and small datasets of their own, they can not repeat these performances for real life problems. To increase the flexibility, generalizability and configurability for any domain without losing the level of automaticity, the problem can be attacked as two separate sub-problems; feature extraction and classification. Separating classification from the feature extraction decreases the dependency on specific features while making decisions on objects(images).

This thesis proposes a mechanism that separates feature extraction from classification and attacks the problem as a categorization problem. Possible objects are extracted from videos/images by using Normalized-cut segmentation and categorization process is applied on them. To handle the categorization, a classifier which is based on Genetic Algorithms (GA) is used. Categories are defined with the Best Representative and Discriminative Feature (BRDF) model, where features are MPEG-7 descriptors [38]. BRDF model defines multiple different features of the categories that are specific to the categories. To generate BRDF model of the categories automatically, a statistical feature importance determination method is proposed. During categorization, the decisions of the classifier are calculated by using these features and the BRDF model. With the nature of GA, the classifier improves itself in time by using genetic operations of GA. Also, the system supports fuzziness by making multiple categorization and giving fuzzy decisions on the objects.

To give more details on the solution, it can told that all possible object candidates are found in the frames of the video and then each candidate is tried to be classified. This solution contains several sub-tasks as shown in Figure 1.1.

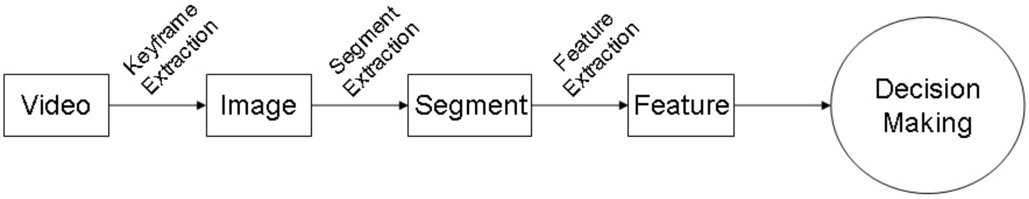


Figure 1.1: Sub-tasks in Object Extraction from Video

According to the flow in Figure 1.1, when a video is encountered, firstly the frames/ keyframes of the video are extracted by using a popular keyframe extraction tool. Then to find possible objects in each frame/ keyframe, segmentation is performed on it. For segmentation, Ncut segmentation algorithm [56] of Shi and Malik is used. Ncut segmentation algorithm performs over-segmentation on frames as a result of graph partitioning, which mostly yields objects or parts of the objects. Thus all possible objects in the video can be obtained by treating each segment and each neighboring groups as candidate objects. Next operation to be performed is the classification of the candidate objects. Among all these steps, the thesis focuses on the classification problem, which is named as 'Decision Making' on Figure 1.1. For classification of a candidate object into defined categories, a GA based classifier is used. The classifier gives decisions according to the features (MPEG-7 descriptors) of images. Each category is defined with the importance values of features that represent and discriminate that category. While making decisions, only the important features for each category are used according to the importance value. The classifier makes multiple categorization and makes fuzzy decisions. To perform feature extraction and feature comparison, the Experimentation Model (XM) software [42] of MPEG-7 research group is used.

The proposed system is designed as a component oriented approach and mostly dealt with the classification problem. The scope of the thesis does not contain segmentation and image processing details. The aim is to construct a model that unifies the MPEG-7 descriptor decisions in different ratios for different categories with a Best Representative and Discriminative Feature (BRDF) methodology. Other tasks are adapted from different studies to the system as components.

1.1 Motivation Behind The Proposed System

The main motivation for this thesis is the need of CBIR systems for automatic content indexing. In fact, this thesis study is started as a part of a CBIR system called Ontology-Supported Video Database Model (OVDAM) [67] [68]. OVDAM needs objects and low-level features of objects in order to extract events and concepts in videos. The system developed here supports these needs of OVDAM. But, not only the needs of OVDAM is considered; the system can be used with any CBIR system as an automatic content indexer. Besides, the system can be used for object extraction from videos and images, as a standalone system.

Considering that the proposed system here deals with both images and videos, the thesis is named as “Object Extraction From Images/Videos Using a Genetic Algorithm Based Approach”.

1.2 Thesis Outline

Organization of this thesis is as follows; in Chapter 2, a literature survey on the topics dealed in the thesis is presented. In Chapter 3, background knowledge that is necessary to overlook the thesis is provided. Chapter 4 explains the proposed system in details. In Chapter 5, the implementation of the system, the tests performed and the evaluations are given. Lastly Chapter 6 concludes and gives future work.

CHAPTER 2

REVIEW OF THE LITERATURE

In this chapter, a review of survey on the topics that are related with this study is presented. Since the topics are in a broad range, they are given under different sections separately. Also, a comparison/evaluation with the proposed system is given at the end of the sections, when necessary.

As mentioned before, the object extraction process mainly contains two parts; performing feature extraction on the image and then using the extracted features to retrieve a result. So which features are selected, how they are selected and how the extraction results can be used are the main areas for research. The chapter summaries the approaches for feature selection and object/image retrieval studies in the literature in first two sections. In the third section, the object extraction mechanisms using Genetic Algorithms approach are given due to the importance of Genetic Algorithms in this study. Fourth section lists the studies on implementations of descriptors in MPEG-7 Reference Software, XM Software. Last section is for introducing the studies that is used for performance comparison at the evaluation chapter.

The other salient ideas used in this thesis study; normalized cut segmentation, best representative feature selection are not discussed in this chapter since they are given in Chapter 3 in detail.

2.1 Feature Extraction

Feature extraction is the most indispensable part of an object extraction mechanism from images/videos. All methods and models use feature extraction as a preliminary part, but also most of them depend on the features they selected. So feature selection is an important question on its own. Many researches are performed for comparing, combining, creating features on different domains and many different methologies are applied. [12] and [57] present the studies on the topic done in last ten years. Below, some of the recent studies are discussed briefly.

[13], [15], [24] deal with the color descriptors. In [13], a compact color descriptor and an efficient indexing method with region clustering is proposed. Colors in a given region are clustered into a small number of representative colors and the feature descriptor consists of the representative colors and their percentages in the region. The representative colors are indexed in the three-dimensional color space. Results show that, computationally, it is more efficient than traditional color histograms. In [15], forming a multiresolution histogram by using color histograms of multiple resolutions of an image instead of a single image histogram is proposed and obtains better performance against five widely used image features. In [24], again color histograms are used, but Gauss mixture vector quantization (GMVQ) is used as a quantization method for color histogram generation and results give better retrieval than uniform quantization for color images. In addition to these, in [7], a set color and texture descriptors including dominant color, spatial color, texture and histogram based descriptors are presented and evaluated as MPEG-7 descriptors.

Shape is an another important descriptor. In [2], a Fourier-based approach that uses Fourier coefficients and a distance named Dynamic Time Warping to compare shape descriptors is proposed. In [4], a new descriptor, 'shape context', which captures the distribution of the remaining points relative to a reference point is presented. In [31], a cognitively motivated similarity measure is presented. The method presented simplifies the shapes with a novel process of digital curve evolution, establishes the best possible correspondence of visual parts and computes the similarity between corresponding parts. [50] proposes an approach for matching distorted and possibly occluded shapes using Dynamic Programming and bases the shape matching and retrieval on Fourier descriptors and moments.

[12] and [57] survey totally more than 100 studies on features and feature extraction. Considering the above reviews and the surveys in [12] and [57], it can be said that the studies tend to make improvements on the current features by combining more than one feature, making changes on the calculations or making corrections for distortions and noises rather than proposing brand new features. As most of these studies do, it is possible to deal with low level definitions and mathematical calculations on features and make improvements or combinations on these. Furthermore it is possible to create a model that does not deal with low level definitions but the results of the features which are currently declared or will be declared from now on. This study prefers to create such a model considering that a high-level

model that can combine any future in the literature is more effective and proposes a model containing a Best Representative and Discriminative Feature selection mechanism.

2.2 Object/Image Retrieval

There is a large number of different image retrievel systems proposed. In [12], it is shown that the number of publications about image retrieval increased 20 times in last ten years; it has increased from 50 per year to 1000 per year (Figure 2.1).

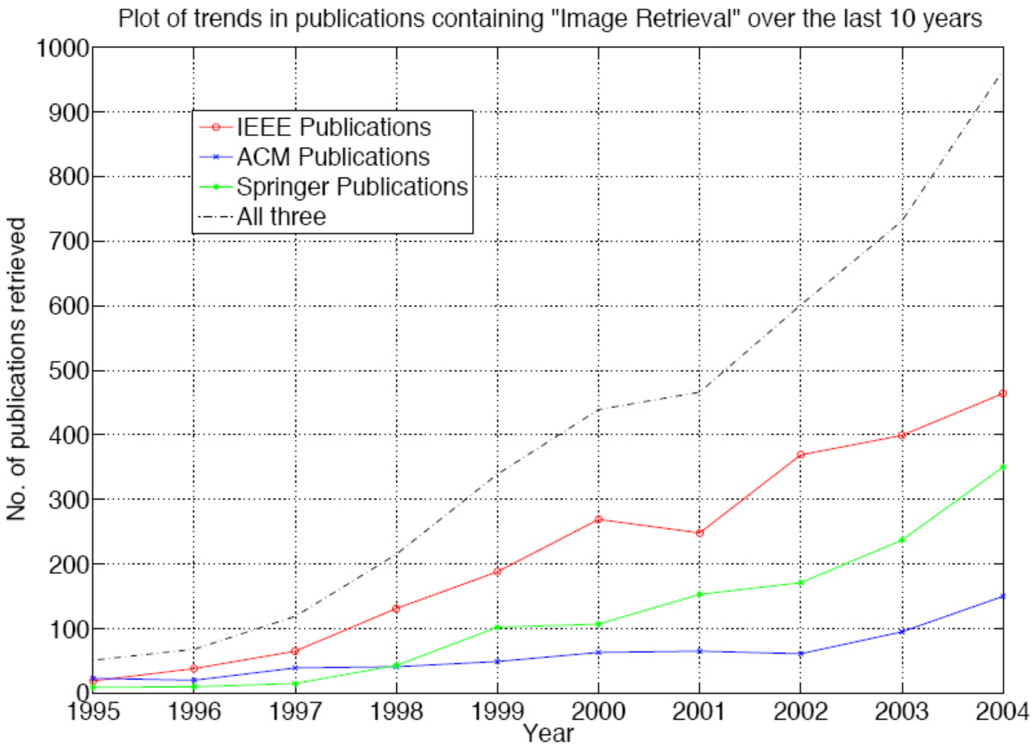


Figure 2.1: Publisher wise break-up of publication count on papers having “image retrieval” [12]

The surveys [12] and [57], successfully examines the studies in last ten years. In [12], it is seen that region-based image retrieval methodology takes an important place among the studies, many studies use region-based retrieval as the core idea and make improvements, although it is hard to group the studies due to the broad range of ideas used. Besides, [57] groups the studies according to how they use the features extracted from the image; the studies using a semantic interpretation on the features and the ones bases on the similarity functions of the features. [57] examines 200 studies and gives brief information on how they use features of images to perform image retrieval.

Since this study mostly deals with GA based retrieval systems, giving only the trends in the image retrieval area regarded enough and details on other systems is not given.

2.3 Object Extraction with Genetic Algorithms

Genetic Algorithms methodology is a powerful search technique for solving problems in many different research areas. GAs are mostly used for improving the performance; instead of testing all combinations, GAs ensure the most fitting-ones to be obtained in fewer tries. There has been an extensive research in this field. GAs are tried to be used in different levels/phases of object/image retrieval and below some remarkable ones are given briefly as introducing how they use GA.

[27], [22], [26], [28] and [49] propose a model such that a Genetic Algorithm is used during the spatial segmentation of videos. Chromosomes are modelled as containing a label and a feature vector, fitness function is as the energy function which is the energy in a particular pixel calculated using features of the pixel with Markov Random Field Model. The model contains three levels as the frame, the segment and the pixel, which of all have energy function. Using fitness function of GA, the pixels with minimum energy is tried to be obtained. The evolution of chromosomes are performed between consecutive frames.

[14] uses GA method for obtaining the representative frames of the selected scenes from the video. In the model, GA is not used during scene selection or feature selection, but is used to obtain representative frames. The selection of representative frames is achieved by selecting the frames with the minimum correlation among them. In GA model, the correlation between frames is used for fitness function and the chromosomes are defined by using the frames and features of the frames.

Besides, [59] do not use GA for frame selection, but for object localization in the frame. The method uses the idea of examining all possible subimage positions and sizes in oder to find the objects in the frame. Due to the enormous computational complexity for a brute-force solution, it uses GA based model instead with the same idea. The GA model uses coordinates of random shapes for representation in the chromosomes, and evolution is performed. The method requires a training phase in which a set of training images are used and objects in these images are given by hand. The Most Expressive Features (MEF) and Most Descriptive Features (MDF) of these objects are stored in database to be used as the fitness function during the tests.

[45] proposes a novel hybrid genetic algorithm for feature selection. It simply encodes chromosomes as the available features, assigns the values of features as the fitness function, uses the most basic GA, but supports the GA by providing a local improvement on offsprings obtained during crossover of GA.

[60] uses a thresholding based segmentation method to obtain the objects. For thresholding, it divides the images to dark, gray and white parts, defines fuzzy parameters to obtain these parts and proposes a procedure for finding the optimal combination of all the fuzzy parameters by implementing a genetic algorithm with appropriate coding method so as to avoid useless chromosomes. In the model, chromosomes are encoded by three levels of threshold values (both min and max). Fitness function is defined as the entropy function which reflects the amount of dispersion in the image.

[18] considers high-level shape and textural information due to its domain. The proposed model consists of training and test parts. In training part, the objects are marked in the training images, shape and texture information of objects are stored in the database. In the test part, the shapes are retrieved from database and applied on the test images. This comparison work applied via a GA based model. The shapes and locations on the image defined as the chromosomes and the comparison result on the texture information is used as the fitness function. By using GA methodology, more fitting shapes can be obtained.

[23] defines a GA model by using the chromosomes as the selected pixels from the image and the fitness function as the edge relations in the image. The model uses uniformly distributed random number generator to generate initial population of chromosomes and evolves these to obtain the object in the image. The model defines the object as the set of selected pixels in the image.

[51] and [52] deal with separating object from background and uses thresholding technique for segmentation. They use GAs for determining the thresholding value. So the chromosomes are modelled as gray level values of 0 to 256 which are the thresholding values. Fitness function is the histogram function. They try to find the best thresholding value by evolving the the chromosomes defined.

Considering above given studies, it can be stated that GAs are used for many different purposes in object extraction; for selecting representative frame of video, for generating coordinated for objects, for feature selection, for pixel selection, etc. To the best of our knowledge, this study is the first study that uses GA for selecting the representative features of representative images for categories. Furthermore the GA is used only for object categorization, it does not affected from the features used in the system. Using GA for categorization problem is also studied by Şahin in [53], but for Text Categorization. In his study, Şahin uses words to represent categories and uses GA to obtain for improvement of the categories.

2.4 XM Reference Software Descriptor Implementations

This thesis study does not deal with low level features and feature extraction methods on images. But since XM Reference Software is used for extracting features (MPEG-7 descriptors) of the images, it is necessary to give the references to the studies that implements the descriptors used in the implementation of this study.

As mentioned before, in this study, 8 descriptors are used. Color Layout descriptor implementation is described in [25], Color Structure in [39], Dominant Color in [9], Scalable Color in [46] and [47]. Shape descriptors Contour Shape and Region Shape implementations are declared in [6] and [29], texture descriptors Edge Histogram and Homogeneous Texture are defined in [48] and [66], respectively. Also [7] presents an overview of all color and texture descriptors and implementations.

2.5 Studies Using CalTech 101 Dataset

In the evaluation chapter, CalTech 101 Dataset [17] is used for performance measurement. Below, the studies that uses CalTech 101 Dataset in the literature are given as a list. These studies are used for comparison.

- Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories of Fei-Fei et al. [16]

- Shape Matching and Object Recognition Using Low Distortion Correspondences of Berg et al. [5]

- The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features of Grauman et al. [20]

- Combining Generative Models and Fisher Kernels for Object Recognition of Holub et al. [21]

- Object Recognition with Features Inspired by Visual Cortex of Serre et al. [55]

- SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition of Zhang et al. [70]

- Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories of Lazebnik et al. [32]

- Empirical Study of Multi-scale Filter Banks for Object Categorization of Jimenez et al. [37]

- Multiclass Object Recognition with Sparse, Localized Features of Mutch et al. [43]

- Using Dependent Regions for Object Categorization in a Generative Framework of Wang et al. [62]

CHAPTER 3

BACKGROUND KNOWLEDGE

In this chapter, the fundemantal concepts about the ideas that the study is built on are presented. For this purpose; MPEG-7 standards, normalized cut segmentation, best representative feature selection and genetic algorithms are presented. The major goal of this chapter is to give readers some brief information about the building stones of the proposed system.

3.1 MPEG-7 Standard and Features of Multimedia Data

MPEG-7 is an ISO/IEC standard developed by MPEG (Moving Picture Experts Group) [38]. Formally called "Multimedia Content Description Interface" specifies the standard set of descriptors that can be used to describe various types of multimedia information. MPEG-7 also standardizes ways to define other descriptors as well as structures (Description Schemes) for the descriptors and their relationships. Furthermore, MPEG-7 defines a Description Definition Language (DDL) to standardise the language to specify description schemes. The material that MPEG-7 standarts can be used may include: still pictures, graphics, 3D models, audio, speech, video, and information about how these elements are combined in a multimedia presentation ('scenarios', composition information). [40]

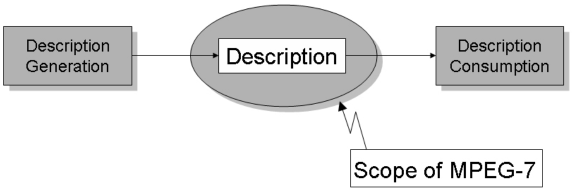


Figure 3.1: Scope of MPEG-7

Briefly, MPEG-7 focuses on the standardization of a common interface for describing multimedia materials (representing information about the content, but not the content itself). In this context, MPEG-7 addresses aspects such as facilitating interoperability and globalization of data resources and flexibility of data management. The purpose of MPEG-7 standarts is to make audio-visual material as searchable as text.

The MPEG-7 Standard consists of the following parts [38]:

- MPEG-7 Systems: The tools needed to prepare MPEG-7 descriptions for efficient transport and storage and the terminal architecture.
- MPEG-7 Description Definition Language: The language for defining the syntax of the MPEG-7 Description Tools and for defining new Description Schemes.
- MPEG-7 Visual: The Description Tools dealing with (only) Visual descriptions.
- MPEG-7 Audio: The Description Tools dealing with (only) Audio descriptions.
- MPEG-7 Multimedia Description Schemes: The Description Tools dealing with generic features and multimedia descriptions.
- MPEG-7 Reference Software: A software implementation of relevant parts of the MPEG-7 Standard with normative status.
- MPEG-7 Conformance Testing: Guidelines and procedures for testing conformance of MPEG-7 implementations
- MPEG-7 Extraction and use of descriptions: Informative material (in the form of a Technical Report) about the extraction and use of some of the Description Tools.
- MPEG-7 Profiles and levels: Provides guidelines and standard profiles.
- MPEG-7 Schema Definition: Specifies the schema using the Description Definition Language

This study mostly deals with the MPEG-7 Visual descriptors, MPEG-7 Description Definition Language and MPEG-7 Reference Software. In fact, only Reference Software (Experimentation Model - XM Software) is directly used. The software provides extraction and query capabilities for some of the descriptors, and give result definitions in MPEG-7 DDL. The visual descriptors dealt with in the study are described in detail in the next subsections.

3.1.1 MPEG-7 Visual Descriptors

In order not to cause a confusion, it should be firsly said that, the word ‘descriptor’ means ‘feature’ of an image. Also, visual descriptors that are described in this subsection are the global image features like color histogram or local features like shape and texture.

MPEG-7 Standarts Overview documentation [38] gives brief definitions of MPEG-7 descriptors. Below descriptor definitions from [38] and XM Software [41] documentation is given. Although not all of the descriptors are used in the implementation of the proposed system, any of the descriptors can be adapted to the system. Below, only the handled descriptors in the implementation is given in detail.

MPEG-7 Visual Descriptors are classified under six categories: Basic Structures, Color Descriptors, Texture Descriptors, Shape Descriptors, Motion Descriptors, Localization Descriptors and Face Recognition Descriptor. Each category consists of elementary and sophisticated descriptors.

Basic Structures, Motion Descriptors, Localization Descriptors, Other Descriptors

There are five Visual related Basic structures: the Grid layout, and the Time series, Multiple view, the Spatial 2D coordinates, and Temporal interpolation. Motion descriptors are Camera Motion, Motion Trajectory, Parametric Motion, and Motion Activity. Localization descriptors are Region locator and Spatio-temporal locator. Also the Face Recognition descriptor is stated under ‘Others’ category. None of these structures and descriptors are used in the implementation of the proposed system.

Color Descriptors

There are seven Color Descriptors: Color space, Color Quantization, Dominant Colors, Scalable Color, Color Layout, Color Structure, and GoF/GoP Color. Four of them are used in the implementation of proposed system: Color Layout, Color Structure, Dominant Color and Scalable Color.

- Dominant Color: This color descriptor is most suitable for representing local (object or image region) features where a small number of colors are enough to characterize the color information in the region of interest. Whole images are also applicable, for example, flag images or color trademark images. Color quantization is used to extract a small number of representing colors in each region/image. The percentage of each quantized color in the region is calculated correspondingly. A spatial coherency on the entire descriptor is also defined, and is used in similarity retrieval.

- Scalable Color: The Scalable Color Descriptor is a Color Histogram in HSV Color Space, which is encoded by a Haar transform. Its binary representation is scalable in terms of bin numbers and bit representation accuracy over a broad range of data rates. The Scalable Color Descriptor is useful for image-to-image matching and retrieval based on color feature. Retrieval accuracy increases with the number of bits used in the representation.

- Color Layout: This descriptor effectively represents the spatial distribution of color of visual signals in a very compact form. This compactness allows visual signal matching functionality with high retrieval efficiency at very small computational costs. It provides image-to-image matching as well as ultra high-speed sequence-to-sequence matching, which requires so many repetitions of similarity calculations. It also provides very friendly user interface using hand-written sketch queries since this descriptors captures the layout information of color feature. The sketch queries are not supported in other color descriptors.

- Color Structure: The Color structure descriptor is a color feature descriptor that captures both color content (similar to a color histogram) and information about the structure of this content. Its main functionality is image-to-image matching and its intended use is for still-image retrieval, where an image may consist of either a single rectangular frame or arbitrarily shaped, possibly disconnected, regions. The extraction method embeds color structure information into the descriptor by taking into account all colors in a structuring element of 8x8 pixels that slides over the image, instead of considering each pixel separately. Unlike the color histogram, this descriptor can distinguish between two images in which a given color is present in identical amounts but where the structure of the groups of pixels having that color is different in the two images. Color values are represented in the double-coned HMMD color space, which is quantized non-uniformly into 32, 64, 128 or 256 bins. Each bin amplitude value is represented by an 8-bit code. The Color Structure descriptor provides additional functionality and improved similarity-based image retrieval performance for natural images compared to the ordinary color histogram.

Texture Descriptors

There are three texture Descriptors: Homogeneous Texture, Edge Histogram, and Texture Browsing. Two of these are used in the implementation; Homogeneous Texture and Edge Histogram

- Homogenous Texture: Homogeneous texture has emerged as an important visual primitive for searching and browsing through large collections of similar looking patterns. An image can be considered as a mosaic of homogeneous textures so that these texture features associated with the regions can be used to index the image data. The Homogeneous Texture Descriptor provides a quantitative representation using 62 numbers (quantified to 8 bits each) that is useful for similarity retrieval. The extraction is done as the following: The image is first filtered with a bank of orientation and scale tuned filters (modeled using Gabor functions) using Gabor filters. The first and the second moments of the energy in the frequency domain in the corresponding sub-bands are then used as the components of the texture descriptor. The number of filters used is 5x6 = 30 where 5 is the number of "scales" and 6 is the number of “directions” used in the multi-resolution decomposition using Gabor functions. An efficient implementation using projections and 1-D filtering operations exists for feature extraction. The Homogeneous Texture descriptor provides a precise quantitative description of a texture that can be used for accurate search and retrieval in this respect. The computation of this descriptor is based on filtering using scale and orientation selective kernels.

- Edge Histogram: The edge histogram descriptor represents the spatial distribution of five types of edges, namely four directional edges and one non-directional edge. Since edges play an important role for image perception, it can retrieve images with similar semantic meaning. Thus, it primarily targets image-to-image matching (by example or by sketch), especially for natural images with non-uniform edge distribution. In this context, the image retrieval performance can be significantly improved if the edge histogram descriptor is combined with other Descriptors such as the color histogram descriptor. Besides, the best retrieval performances considering this descriptor alone are obtained by using the semi-global and the global histograms generated directly from the edge histogram descriptor as well as the local ones for the matching process.

Shape Descriptors

There are three shape Descriptors: Region Shape, Contour Shape, and Shape 3D. Two of them are used in this study; Region Shape and Contour Shape.

- Region Shape: The shape of an object may consist of either a single region or a set of regions as well as some holes in the object as illustrated in Figure 3.2. Since the Region Shape descriptor makes use of all pixels constituting the shape within a frame, it can describe any shapes, i.e. not only a simple shape with a single connected region as in Figure 3.2 (a) and (b) but also a complex shape that consists of holes in the object or several disjoint regions as illustrated in Figure 3.2 (c), (d) and (e), respectively. The Region Shape descriptor not only can describe such diverse shapes efficiently in a single descriptor, but is also robust to minor deformation along the boundary of the object. Figure 3.2 (g), (h) and (i) are very similar shape images for a cup. The differences are at the handle. Shape (g) has a crack at the lower handle while the handle in (i) is filled. The region-based shape descriptor considers (g) and (h) similar but different from (i) because the handle is filled. Similarly, Figure 3.2 (j-l) show the part of video sequence where two disks are being separated. With the region-based descriptor, they are considered similar.

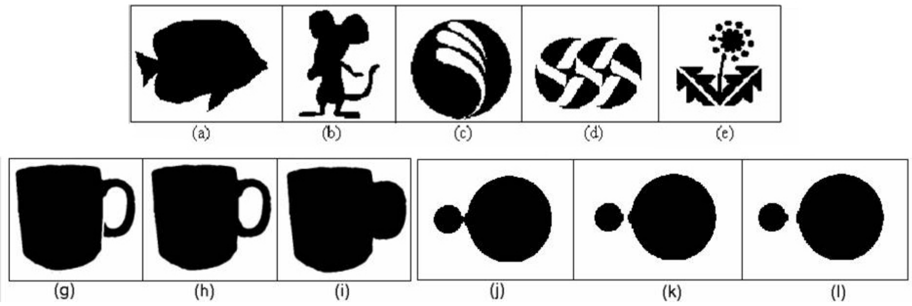


Figure 3.2: Examples of various shapes for Region Shape

- Contour Shape: The Contour Shape descriptor captures characteristic shape features of an object or region based on its contour. It uses so-called Curvature Scale-Space representation, which captures perceptually meaningful features of the shape. The object contour-based shape descriptor is based on the Curvature Scale Space representation of the contour. This representation has a number of important properties, namely:

- It captures very well characteristic features of the shape, enabling similarity-based retrieval
- It reflects properties of the perception of human visual system and offers good generalization
- It is robust to non-rigid motion
- It is robust to partial occlusion of the shape
- It is robust to perspective transformations, which result from the changes of the camera parameters and are common in images and video
- It is compact

Some of the above properties of this descriptor are illustrated in Figure 3.3, each frame containing very similar images according to CSS, based on the actual retrieval results from the MPEG-7 shape database.

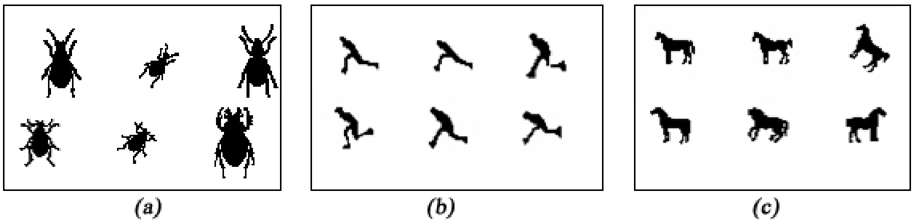


Figure 3.3: Examples of various shapes for Contour Shape: (a) shape generalization properties (perceptual similarity among different shapes), (b) robustness to non-rigid motion (man running), (c) robustness to partial occlusion (tails or legs of the horses)

3.1.2 MPEG-7 Reference Software

The MPEG-7 reference software (eXperimentation Model, XM) is the software that is created for generating conformant MPEG-7 bit streams/DDL streams on MPEG-7 descriptors from videos. The software provides specific algorithms and implementations for generating conformant streams. (There can be other software that generate conformant streams but they do not need to use the same algorithms.) The overview documentation on XM software [42] [38] gives more detailed information on the software, here an introductory information is given.

In this study, two application types in XM software is used: Extraction application and Search&Retrieval application. The software also contains Media Transcoding application and Description Filtering application.

The extraction application shown in Figure 3.4 is used for extracting descriptors from the input media data. Most of the Descriptors (Ds) and Description Schemes(DSs) defined in DDL are implemented in XM software. During extraction, first the multimedia file is loaded by the media decoder module. Next, the description is extracted by the extraction module. Then the description is passed through the encoder and the encoded data is written to a file. This process is performed for all multimedia files in the given database. The place of XM Extraction Application in the proposed system of this study is given in Figure 4.1 as "Feature Extraction".

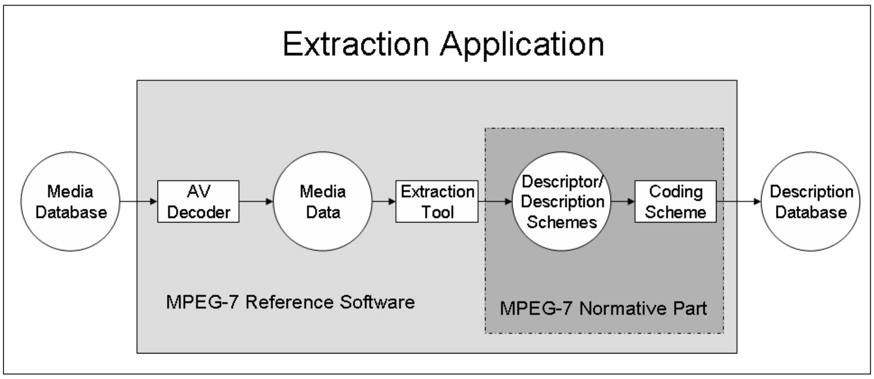


Figure 3.4: Schematic diagram of an "Extraction Application" using the XM reference software modules. In the block diagram boxes represent procedural parts, circles represent data structures. [42]

The search & retrieval application, shown in Figure 3.5 is for comparing an input multimedia file with the previously extracted multimedia database. During search & retrieval process, first all previously extracted description of the multimedia database are decoded from file and loaded. Also the multimedia file used for querying is extracted as in extraction application. Then, the query is performed for query descriptions on loaded database descriptions by matching module and all comparisons performed. Lastly, the resulting distance values obtained from matching module are sorted and given as output. The place of XM Search & Retrieval Application in the proposed system of this study is given in Figure 4.1 as "Feature Comparison".

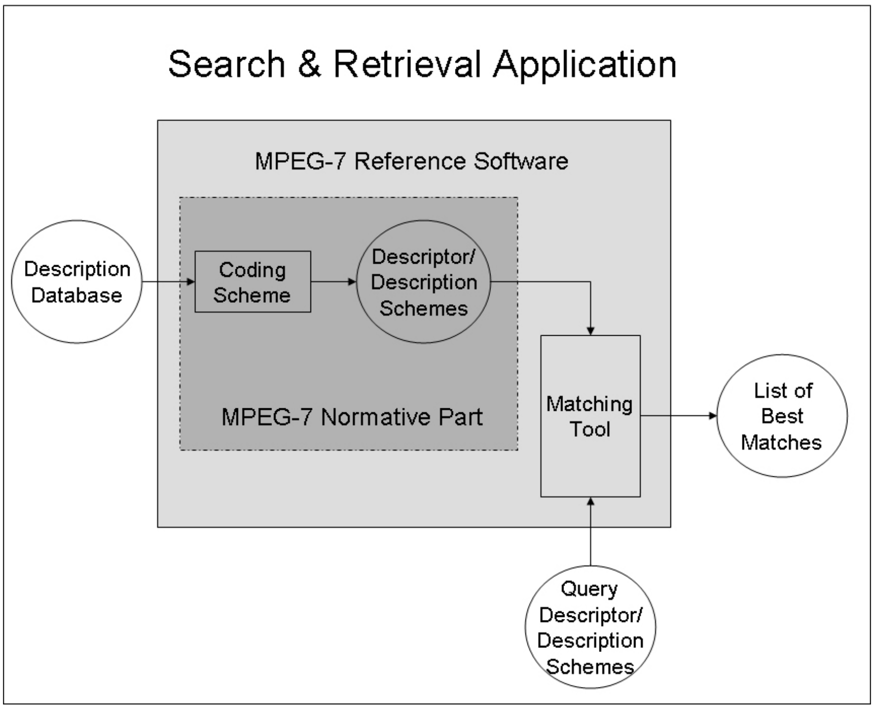

Search & Retrieval Application
MPEG-7 Reference Software
Description Database
Coding Scheme
Descriptor/ Description Schemes
MPEG-7 Normative Part
Matching Tool
List of Best Matches
Query Descriptor/ Description Schemes

Figure 3.5: Schematic diagram of a "Search and Retrieval Application" using the XM reference software modules. In the block diagram boxes represent procedural parts, circles represent data structures. [42]

The software is distributed as C++ code [41]. It provides a command line based feature extraction and search & retrieval system. For each of the MPEG-7 descriptors (it is dealt with only visual descriptos in this study), there is an application (implementation) for extraction and one application for search & retrieval.

3.2 Normalized Cut Image Segmentation

Normalized Cut (Ncut) Segmentation is a segmentation method for images that is proposed by Shi and Malik [56]. Shi and Malik proposed a novel approach by not focusing on the local features of the images but treating image segmentation problem as a graph partitioning problem.

The approach is mostly related to the graph theoretic formulation of grouping. The set of points in an arbitrary feature space are represented as a weighted undirected graph $G = (V, E)$, where the nodes are the points in the feature space, edges are the links between every pair of nodes and weight on each edge, $w(i, j)$, is a function of the similarity between nodes i and j. For example; for an image, the brightness value (this can be any feature for the feature set) of each pixel in the image are the nodes of the image and similarity between these brightness values becomes the weights. The aim of the approach is partitioning the set of vertices into disjoint sets $P = (V_1, V_2, .., V_m)$, where the similarity between any two matrices in P is low.

The partitioning is done by recursively partitioning the current graph into two disjoint sets and removing edges between them. The degree of dissimilarity between these two sets can be computed as the total weight of the edges that are removed:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \tag{3.1}$$

So the selection of partitioning is done according to the dissimilarity value and the sets with minimum dissimilarity value are selected (which is the problem minimum cut). Although minimum cut produce good segmentation on some images, it is known that this method favors cutting small sets of isolated nodes in the graph. So a new idea of normalized cut (Ncut) is proposed as:

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)} \tag{3.2}$$

where $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ is the total connection from nodes in A to all nodes in the graph. With this new idea, the small isolated points will no longer have small Ncut value although they have small cut value. Since calculation of minimizing normalized cut is an NP-complete problem, Shi and Malik offer an approximate solution for computing the optimal partition by solving a generalized eigenvalue problem.

The proposed grouping algorithm consists of the following steps:

1. Given an image or image sequence, set up a weighted graph $G = (V, E)$ and set the weight on the edge connecting two nodes to be a measure of the similarity between the two nodes.

2. Solve $(D - W)x = \lambda Dx$ for eigenvectors with the smallest eigenvalues.

3. Use the eigenvector with second smallest eigenvalue to bipartition the graph.

4. Decide if the current partition should be sub-divided, and recursively repartition the segmented parts if necessary.

Clearly, normalized cut can be defined as an unbiased measure of disassociation between sub-groups of a graph. The algorithm performs over-segmentation on images as a result of graph partitioning, which mostly yields objects or parts of the objects. The method is a successful one; the applications using this approach found good results in several researches.

In this study, a MATLAB implementation [11] by Cour et al. named as "Multiscale Normalized Cut Segmentation" [10] is used. In [10], Cour et al. present a multiscale spectral image segmentation algorithm. The algorithm works on multiple scales of the image in parallel, without iteration, to capture both coarse and fine level details. Also, the algorithm is computationally efficient so it allows to segment large images. The results they obtained show that they incorporate long-range connections with linear-time complexity, providing high-quality segmentations efficiently. Thus, this algorithm and implementation provided the facility to segment images that previous implementations could not processed because of the size of the images.

3.3 Best Representative Feature Selection

In image retrieval systems, as some examples are given in Section 2.1, usually a particular feature or a common set of features are used for comparing the query image with the database images. In these systems, the features are selected to represent the problem domain. But, if the size of the database and/or the diversity of image collection is increased, these methods fails to give satisfactory results.

The problem can be summarized as follows: Using same features for different domains and types of objects yields unsatisfactory results. Finding a solution to the problem is quite trivial; using different features for different object types. For example, shape features are more important than color features for a car whereas sea can be defined with color and texture features.

To describe the approach more formally, a classification problem with 2 classes can be considered. It is assumed that class C_1 contains n_1 number of images and C_2 contains n_2 number of images in the database. Also, it is assumed that the images of class C_1 can be defined with color features and the images of C_2 can be defined with shape features. If this database is used in an image retrieval system that compares images according to only color features or shape features, the performance of the system will be nearly 50%. If color features are used, the performance of the system will be satisfactory for C_1, but not for C_2. To obtain a satisfactory performance for the whole system, different features should be used for different classes.

In [61], Uysal et al. use a different feature for each object class and proposes a robust approach identifies Best Representative Feature (BRF) for each object class, which maximizes the correct match in a training set. Similarly, in [58] Swets et al. propose to use Most Expressive Features and Most Discriminating Features.

In this study, a method that uses both the best representative features and the best discriminative features is proposed. Representative characteristics of features are calculated according to the similarities of images with the same class and discriminative characteristics are calculated according to the ability of features to distinguish between different object classes. Using these characteristics, a Best Represantive and Discriminative Features (BRDF) index is calculated as detailed in Sub-section 4.6.2.

3.4 Genetic Algorithms

Genetic Algorithms (GAs) are adaptive methods which may be used to solve search and optimization problems. They are based on the genetic process of evolution for biological organisms. [3]

In biology, the evolution is defined as the change in a population's inherited trait from generation to generation. Traits are the expression of genes that are produced, copied and passed from ancestor to offsprings. [65]

Over many generations, natural populations evolve according to the principles of natural selection and "survival of the fittest", first clearly stated by Charles Darwin in *The Origin of Species* [3]. Also, reproduction, mutation and recombination mechanisms in a biological organism's life are important factors for evolution. These mechanisms increase the gene diversity among a population and cause more different new offsprings to be generated. Diversity in a population makes evolution cycle more successful.

In computer science, Genetic Algorithms generally involve with implementation of evolution steps mentioned above, reproduction, mutation, recombination, natural selection and survival of fittest. The GAs are defined with two major concepts [64]:

- A genetic representation of the solution: Genetic representation encodes appearance, behavior, physical qualities of individuals. The representation can be decided according to the problem domain. It can be a list of bits, integers, strings, trees or specially defined objects. Each item in the lists represents a gene and each list represents a chromosome.

- A fitness function to evaluate performance of the solution: A "fitness function" is a particular type of objective function that quantifies the optimality of a solution. It is defined over the genetic representation and measures the quality of the represented solution. The fitness function is always problem dependent.

A simple Genetic Algorithm can be given as in Algorithm 1. The concepts “Initialization”, “Selection”, “Reproduction”, “Termination” are given in detail in next subsections.

Algorithm 1 A Simple Genetic Algorithm

```
Input :  -
Output :  Evolved Population
Choose initial population (Initialization)
Evaluate the fitness of each individual in the population
repeat
    Select best-ranking individuals to reproduce (Selection)
    Breed new generation through crossover and mutation (genetic operations) and give
birth to offspring (Reproduction)
    Evaluate the individual fitnesses of the offspring
    Replace worst ranked part of population with offspring
until Termination
```

3.4.1 Initialization

To perform genetic operations, there should be an initial population of individuals. How it is chosen depends on the problem domain and solution strategy; the population can be generated randomly or obtained from a training data. Traditionally, it is generated randomly, covering all possible solutions. Occasionally, the solutions may be seeded in areas where optimal solutions are likely to be found. [64]

3.4.2 Selection

For each successive generation, some part of the existing population is selected for mating (reproduction). There are many different techniques to select the individuals, some of these methods are listed below [36]. Some of these methods are mutually exclusive, whereas some requires using in combination.

- Elitist selection: The most fit members of each generation are guaranteed to be selected.

- Roulette-wheel selection(Fitness proportionate selection): More fit individuals are more likely to be selected. The chance of an individual's being selected is proportional to the ratio of its fitness over total fitnesses of the population.

- Scaling selection: As the average fitness of the population increases, the strength of the selective pressure also increases and the fitness function becomes more discriminating.

- Tournament selection: Subgroups of individuals are chosen from the larger population, and members of each subgroup compete against each other. Only one individual from each subgroup is chosen to reproduce.

- Rank selection: Each individual in the population is assigned a numerical rank based on fitness, and selection is based on this ranking rather than absolute differences in fitness.

- Generational selection: The offspring of the individuals selected from each generation become the entire next generation. No individuals are retained between generations.

- Steady-state selection: The offspring of the individuals selected from each generation go back into the pre-existing gene pool, replacing some of the less fit members of the previous generation. Some individuals are retained between generations.

- Hierarchical selection: Individuals go through multiple rounds of selection each generation. Lower-level evaluations are faster and less discriminating, while those that survive to higher levels are evaluated more rigorously.

3.4.3 Reproduction

After selecting fittest parents for mating, they are used for reproduction processes; crossover (also called recombination), and/or mutation. For each selected pairs of parents, new child individual(s) are reproduced. Crossover is the process that enables gene interchange between parents so that two new individuals are reproduced that are different from parents. Besides mutation is not affected from parents, it provides a random/rule based gene change on the individuals.

Common forms of crossover can be summarized as below [63]:

- Single-point crossover: A single crossover point on both parents' representation is selected. All data beyond that point in either parents is swapped between the two parents. (Figure 3.6 (a))

- Two-point crossover: Two-point crossover calls for two points to be selected on the parents. Everything between the two points is swapped between the parents. (Figure 3.6 (b))

- Cut and splice: Results in a change in length of the children genes. The reason for this difference is that each parent has a separate choice of crossover point. (Figure 3.6 (c))

- Uniform Crossover: The value at any given location in the offspring's genome is either the value of one parent's genome at that location or the value of the other parent's genome at that location, chosen with 50/50 probability. A mask can be used to define which parent's gene is used for which child. (Figure 3.6 (d))

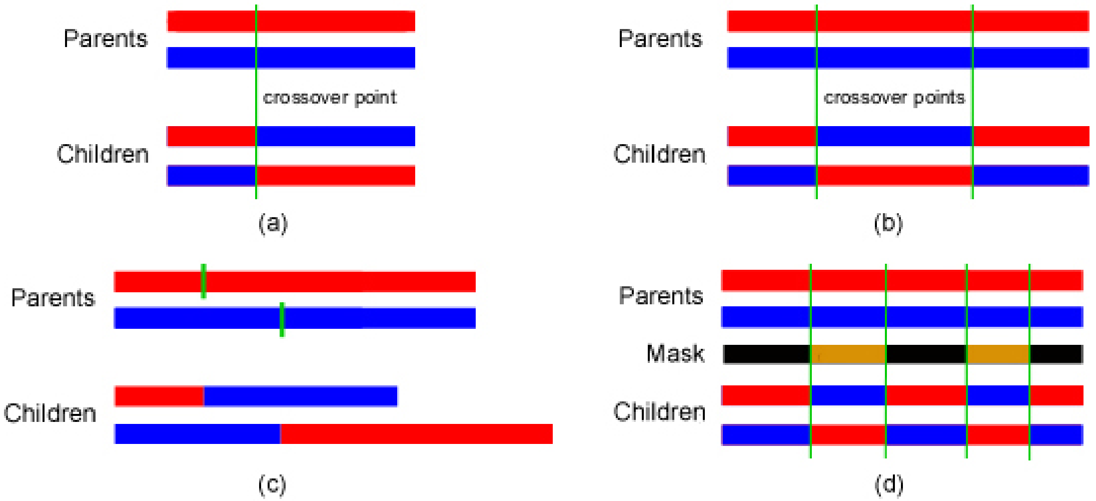


Figure 3.6: Forms of Crossover: (a) Single-point crossover, (b) Two-point crossover, (c) Cut and splice, (d) Uniform Crossover

3.4.4 Termination

The process of reproducing new generations is repeated until a termination condition. Common terminating conditions can be listed as follows [64]:

- A solution is found that satisfies a defined minimum criteria

- Fixed number of generations reached
- Allocated budget (computation time/money) reached
- The highest ranking solution's fitness is reaching or has reached a plateau such that successive iterations no longer produce better results
- Manual inspection
- Different combinations of the above.

CHAPTER 4

OBJECT EXTRACTION AND CLASSIFICATION

In Chapter 2, the main and supporting ideas that form the basis of the “Object Extraction Process” are presented in detail. This chapter describes how these ideas are used during the proposed object extraction process.

Organization of the chapter is as follows: First an overview of the system is given, then phases and contents of the phases during the object extraction process are introduced. After these general views, all details about each content are given.

4.1 Capabilities of the System

Image retrieval systems mainly provide two types of queries; querying which objects occur in an image/video and querying objects with a particular type in an image/image set/video/video database. The first type is more general, quering everything occurs in image/video enables any query possible. Second type is usually provided by storing and indexing mechanism or filtering results of the first type with various ranking or thresholding methods.

The system developed in this study provides all types of queries since it can search any type of objects occur in an image or video. The only restriction to the object types for querying is the supervision of necessary types to the system. The system stores BRDF information (as MPEG-7 descriptors) of the representative training images for each category and gives the result of a query as the decisions of all categories defined in the system. Using such a result, the system can find all occuring objects or whether an object with a particular type occurs, in an image/video. The decisions in the results are fuzzy decisions that define with what ratio the query object is similar to a category. To give precise results, the system provides filtering the decisions optionally with ranking or thresholding on the fuzzy results. Also, the fuzzy results can be used in more complicated CBIR systems during the indexing phase. A more complicated system can define relations between object categories and ontology on them and reach different results by combining the fuzzy decisions of more than one category. In other words, this system can be used as an image retrieval system itself, whereas it can be used as an indexing-data-provider of another retrieval system.

In fact this system is designed as a part of an ontology-supported video database model (OVDAM) which provides a reasonable approach to bridging the gap between low-level representative features and high-level semantic contents [67] [68]. OVDAM needs objects and low-level features of objects in order to extract events and concepts in videos. This system also supports all these needs of OVDAM.

4.2 Overview of the System

Three important ideas are considered during the object extraction process: Ncut Image Segmentation [56] to make segments on the images, Best Representative and Discriminative Features of objects to define objects with their features (which are MPEG-7 descriptors) and a Genetic Algorithm based approach to classify candidate objects from images. Since videos are assumed as a set of images, the system does not differ whether we consider object extraction from videos or images. According to this brief definition, it is clear that the system is designed with a component oriented approach. Some of the components are directly used from other studies due to their success and acceptance in the literature and combined with originally proposed methods in order to increase the success.

The system mainly composed of four components (Figure 4.1): Keyframe Extraction Module, Segmentation Module, Feature Extraction Module, Classification Module. Some of the modules consist of more than one submodules. There are also some small utility modules like Feature Value Normalization Module and Feature Importance Determination Module that are described in the next sections. The names of the modules define their responsibilities in the system, also they are described in detail.

The process mainly aims to find maximum number of segments which are objects or parts of the objects, obtain all possible candidate objects by using the segments and classify candidate objects. For classification a GA based method which can improve itself in time is used.

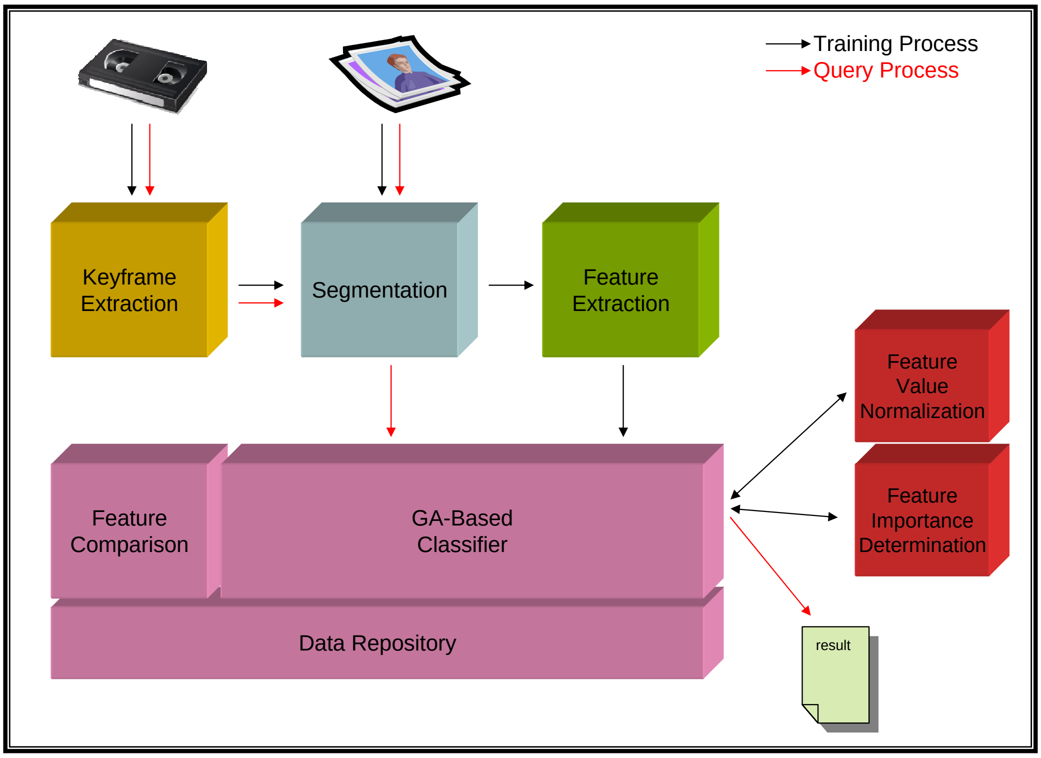


Figure 4.1: Components of The System and Flows During Training and Querying

4.3 Phases During the Use of the System

As mentioned before, the system can be used for both classified object extraction from videos or images as a standalone system, or using with/inside a CBIR system as an automatic object indexing provider. In both cases, the phases are the same, but only the Query phase for the standalone system is equivalent to Indexing for a CBIR system. Below, the phases are described.

4.3.1 Pre-Training

The Domain Experts define object classes. Optionally, they can decide on the important features of these classes by assigning importance values to each feature for each class (Importance of the features can be automatically generated during First-Training).

4.3.2 First-Training

First-training phase aims to obtain base training data for the system. The Domain Experts define objects on the training images and assign their object classes. Then MPEG-7 features of the objects are extracted and stored in the GA based classifier. Each new object for an object class is stored as a new chromosome in the system. In this phase no genetic operators are applied on the data. This phase can be named as the initialization part of the Genetic Algorithm.

First-Training part also includes two more processes: Feature Value Normalization (Sub-section 4.6.1) and Feature Importance Determination (Sub-section 4.6.2).

4.3.3 Second-Training

Second-training phase aims to make improvement on the base training data which is obtained in the first-training phase. As in the first-training phase, the Domain Experts define objects on the training images and assign their object classes. Besides, the system makes segmentation and features of segments extracted but then the segments are sent to the classifier in order to be classified. The classifier gives the results and then the decisions of the classifier are compared with definitions of the Domain Expert. The comparison results and the information that the experts gave are used as the ground truth for the fitness function of the Genetic Algorithm. According to the ground truth, genetic operations (fitness function, crossover and mutation) are applied on the data.

4.3.4 Querying(Indexing)

In querying(indexing) phase, all keyframes of video data are extracted, keyframes are segmented, features are extracted and the segments are sent to the classifier. If only querying is performed, the results are returned to the user, but if an indexing is also performed for a CBIR system, the results are stored in the index storage of the CBIR system. After quering/indexing, an optional feedback mechanism that relies on the evaluations of the Domain Experts can be used to apply genetic operators and make improvement on the training data.

4.4 Keyframe Extraction from Videos

The process is studied as object extraction from images since videos are a set of images (keyframes). Before starting with the image segmentation and object extraction from images, keyframes are obtained by using a keyframe extraction tool. There are many tools to obtain keyframes of videos. We prefer IBM MPEG Annotation Tool [33]. IBM's tool provides facilities to extract shots, keyframes, I-frames and frames from the videos. I-frames are important (representative) frames of shot. When keyframes are not enough in some domains, I-frames provide extra information. For example, in a football video, the tools give keyframes with large intervals since the general view does not change so much. Extracting keyframes with large intervals causes to miss some important frames. I-frames prevent this situation. (IBM MPEG Annotation Tool provides many other features, we only use the keyframe extraction feature.)

4.5 Segmentation and Segment Grouping

In order to extract the segments in the image, the Normalized-Cut Segmentation algorithm of Shi et al. [56] is used. This algorithm performs over-segmentation on images as a result of graph partitioning, which mostly yields objects or parts of the objects. But, since we need objects from the video frames, some further study need to be done on the result of that algorithm in order to combine the over-segments and obtain the object.

For combining the over-segments, a greedy brute-force method is proposed in order not to miss any object although this method is inefficient in terms of time. A more efficient mechanism for segment grouping is left as a future work. For the construction process, firstly each segment obtained is assumed to be a candidate object. Also all combinations of neigbouring segments are taken as candidate objects. All candidate objects are tried to be classified by the GA based classifier. Using ranking or thresholding filters, real objects are obtained.

4.6 MPEG-7 Feature Extraction

After segmentation, the features of the segments are extracted as training data or to make a comparison for decision in the querying phase. These features are MPEG-7 descriptors [38], such as color, texture and shape descriptors. Extraction of features is not a simple process; each of the descriptors can have different extraction methods. Performing low level feature extraction from images is another area of research that we do not deal with in this study. In addition, how we use the feature values is more important for our study than how we implement the extraction process. So it is preferred to use an existing system, with a component oriented approach, for feature extraction. Since we use MPEG-7 descriptors as features, the official software of MPEG, eXperimentation Model (MPEG-7 Reference Software) [41], is the choice as mentioned before.

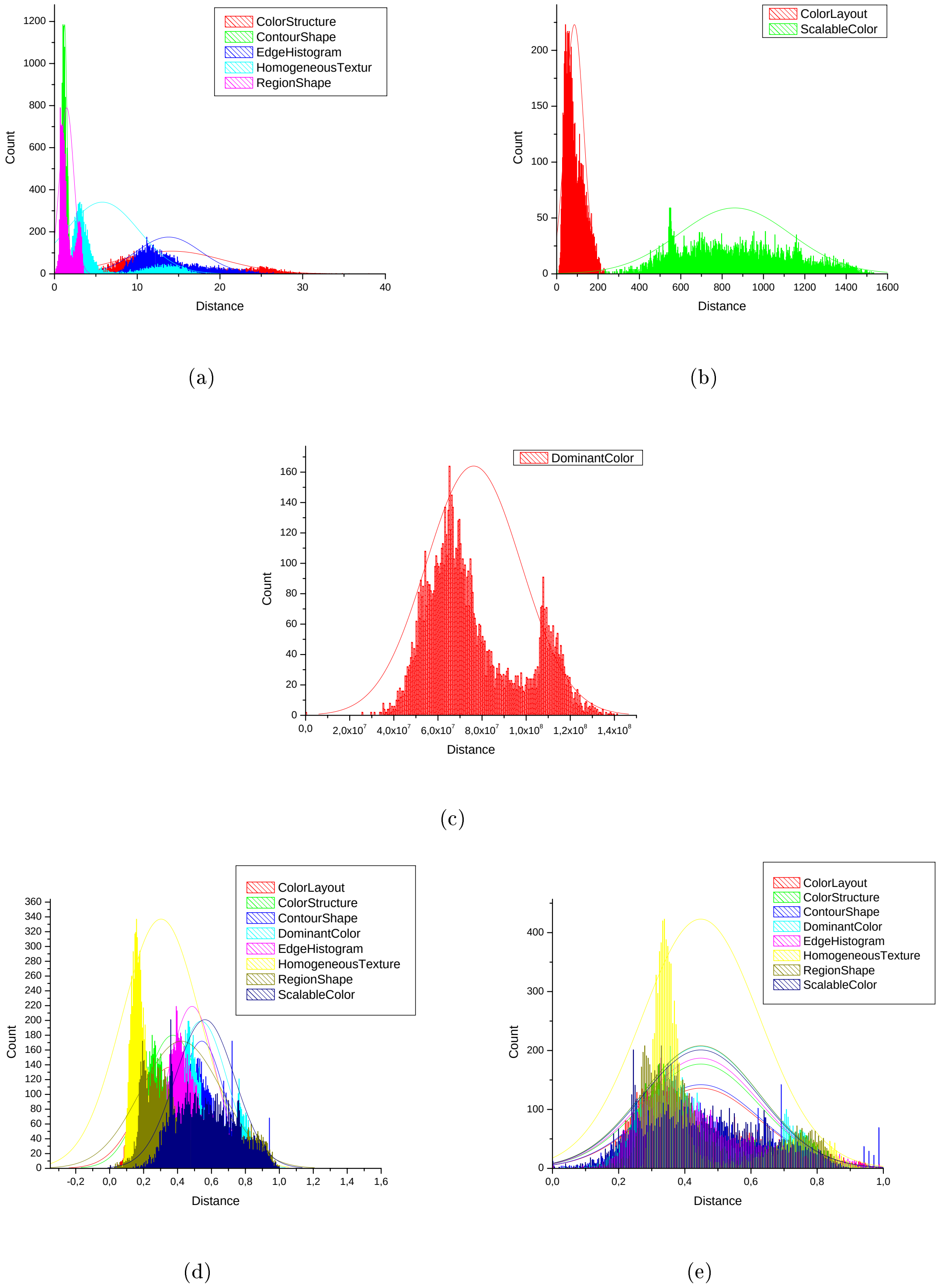
Feature extraction module is used during First-Training and features of all objects obtained from First-Training images are extracted. But, as mentioned in Sub-section 4.3.2, First-Training contains two more processes after feature extraction. Since these processes are related with features and feature values, they are given below.

4.6.1 Normalization on MPEG-7 Feature Similarities

It is explained that feature extraction and comparison of most of the MPEG-7 descriptors are supported in XM Software (Sub-section 3.1.2) and the implementations of these processes are supported by many different researchers (Section 2.4). This situation causes the range of the comparison results to differ. Some of the implementations prefer to give results in a small range like 0 and 1, whereas another one prefers to give between 0 and 10^8. Also, some of them tend to give values closer to their minimum values of their range. But some other ones gives closer to the maximum values of their range. Therefore, in order to use all descriptors together, we should normalize the results of the XM Software.

To understand the difference and see the distribution, an analysis is performed on the images of First-Training with 101 categories. It is necessary to use maximum number of different types of images and increase the diversity of the images in order not to make a restriction on the range and distribution of the results. In Figure 4.2, the distributions of the XM Software comparison results for a random image from First-Training images are given as histograms. In Figure 4.2(a), Figure 4.2(b) and Figure 4.2(c), the distributions before the normalization process is shown. It can be observed that most of the descriptors give results in very different ranges. The Figure 4.2(d) shows the distribution after the ranges of all the result are rescaled to [0-1]. After having same ranges, still the results of the descriptors demonstrate a difference; some of the descriptors tend to give better results than the others. This cannot be described by the selection of the image dataset because the images are selected from a various set of 101 categories, details given in Section 5.2. Also the results of the all descriptors are distributed approximately with a Normal Distribution, but their mean and standard deviations are different. So, a normalization should be performed after rescaling the ranges. Figure 4.2(e) shows the distribution after normalization. Finally, all results of all the descriptors present a similar distribution.

The process for normalization is as follows: After extracting features of all First-Training images obtained, a series of comparisons is performed via XM Software with random images from the First-Traning images. All results from all these queries are grouped according to the features. Then minimum, maximum, mean and standart deviation values are calculated for each group (min_i, max_i, μ_i, σ_i respectively, i indicates the feature). These values are



(a)

(b)

(c)

(d)

(e)

Figure 4.2: Distance Histograms On Software Results: (a),(b),(c) Before Normalization, (d) After Re-scaling to [0,1], (e) After Normalization

stored in the classifier for further use during comparison.

In Classification module, after getting the distance values from XM sotware to calculate the normalized similarity values, these stored values are used during comparison. How they are used is described below.

Let d_i be the distance result obtained from XM Software for feature i. Firstly d_i is rescaled for range;

$$d_i{}' = \frac{d_i - min_i}{max_i - min_i} \tag{4.1}$$

Then the rescaled value is normalized according to the following formula;

$$d_i{}'' = \frac{d_i{}' - \mu_i}{\sigma_i} \cdot avg(\sigma_i) + avg(\mu_i) \tag{4.2}$$

where $avg(\sigma_i)$ is the average of standard deviations of all features and $avg(\mu_i)$ is the average of means of all features.

Lastly the obtained value is equalized to 0 if less than 0 or 1 if more than 1.

4.6.2 Feature Importance Determination

The idea of using features according to their representation and discrimation abilities for each category particularly, is described in Section 3.3. Also in Section 4.3 it is declared that the importance values of features can be assigned optionally by the domain experts or the system can generate them automatically. But assigning correct values for each category is a bothering and difficult task for domain experts. Thus, an automatic mechanism is proposed.

The importance values (imp) of features for each category is also named as the Best Representative and Discriminative Feature (BRDF) index. As the name clearly defines, the index is calculated for a feature on a category according to the representation ability of feature for category and discrimation among other categories. The process is based on statistical calculations over the XM Software comparison results performed on First-Training images.

To calculate the BRDF indices, firstly the similarity values of First-Training images to each other is calculated by using the Feature Comparison module of the system. In fact, Feature Comparison module is a submodule under the classification module and has a responsibility to calculate raw decisions by performing query on XM Software, normalizing XM results and converting the normalized XM results to image similarities. Conversion of XM results to similarity values is only a 1's complement operation since normalized XM results are in a range of [0,1] and means the distance of images. As described before, XM

Software search application makes comparison of a query image with previously encountered images and calculates the distance between them as the result.

By performing above comparison for all images, a table given in Table 4.1 is obtained. In the table, the distances between images calculated by the Feature Comparison module for each feature are shown. By grouping the images in the same categories, Table 4.2 can be derived. This table is derived by getting averages and standard deviations of images of each group.

To calculate the BRDF index, four important parameters is extracted from Table 4.2:

- Mean of Category (μ): μ is the average similarity value of a category on itself, for a particular feature. For a selected category, the features having larger similarity values represent the category better. (The BRDF index is directly proportinal with mean of the category.)

- Standard Deviation of Category (σ): σ is an another important representative property. The features having smaller standart deviations are the ones giving closer similarity values on themselves. (The BRDF index is inversely proportional with the standard deviation of category.)

- Standart Mean Distance to Other Categories (δ_μ): Standart mean distance to other categories is a discriminative feature so it is calculated for a category according to other categories. It is calculated like standard deviation of all means for particular category and particular feature but mean of the selected category is used instead of mean of means.
$$\delta_\mu = \sqrt{\frac{\sum_{i=1}^{n} (\mu_{self} - \mu_i)^2}{n}} \tag{4.3}$$
where n is the number of categories and μ_i values are the values the mean values given in one row of Table 4.2. This calculation gives us the distance of a category mean value to the means of other categories. So having a larger distance means having better discrimination among all categories. (The BRDF index is directly proportional with δ_μ.)

- Number of Wrong Results (ω): Although above given three parameters are important and provide good representation and discrimination, the issue of wrong decision-giving is not considered. It is important for the system to give the largest similarity values for same category images with the query images. The wrong result count is defined as the number of mean values that are larger than mean value of the selected category

Table 4.1: XM Software Query for First-Training Images on Itself

Query Feature	Query Image	d_{image_1}	d_{image_2}	d_{image_3}	$\cdots$	d_{image_m}
f_1	$image_1$	d_{111}	d_{112}	d_{113}	$\cdots$	d_{11m}
f_1	$image_2$	d_{121}	d_{122}	d_{123}	$\cdots$	d_{12m}
f_1	$image_3$	d_{131}	d_{132}	d_{133}	$\cdots$	d_{13m}
f_1	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
f_1	$image_m$	d_{1m1}	d_{1m2}	d_{1m3}	$\cdots$	d_{1mm}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
f_t	$image_1$	d_{t11}	d_{t12}	d_{t13}	$\cdots$	d_{t1m}
f_t	$image_2$	d_{t21}	d_{t22}	d_{t23}	$\cdots$	d_{t2m}
f_t	$image_3$	d_{t31}	d_{t32}	d_{t33}	$\cdots$	d_{t3m}
f_t	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
f_t	$image_m$	d_{tm1}	d_{tm2}	d_{tm3}	$\cdots$	d_{tmm}

Table 4.2: XM Software Results Grouped By Category for First-Training Images on Itself

		Cat_1		Cat_2		$\cdots$	Cat_n	
Query		Mean	Std.Dev.	Mean	Std.Dev.	$\cdots$	Mean	Std.Dev.
	f_1	μ_{111}	σ_{111}	μ_{112}	σ_{112}	$\cdots$	μ_{11n}	σ_{11n}
Cat_1				$\vdots$				
	f_t	μ_{1t1}	σ_{1t1}	μ_{1t2}	σ_{1t2}	$\cdots$	μ_{1tn}	σ_{1tn}
$\vdots$				$\vdots$				
	f_1	μ_{n11}	σ_{n11}	μ_{n12}	σ_{n12}	$\cdots$	μ_{n1n}	σ_{n1n}
Cat_n				$\vdots$				
	f_t	μ_{nt1}	σ_{nt1}	μ_{nt2}	σ_{nt2}	$\cdots$	μ_{ntn}	σ_{ntn}

for a feature. As the wrong count increases, the representation ability decreases. (The BDRF index in inversely proportional with the wrong count.)

Considering the effects of above parameters, the BRDF index of a particular feature f on a particular category Cat can be calculated using the below formula:

$$BRDF_{f,Cat} = \frac{\mu \cdot (1 - \sigma) \cdot \delta_{\mu}}{max(\omega, 1)} \tag{4.4}$$

To understand the above given formulations and definitions, Figure 4.3 can be helpful. The graphs in Figure 4.3 are generated during the analysis study of above formulations. The graphs are generated using a data table like in Table 4.1 for a random sample image. Each graph can be used for comparing importance values of two features. Also the distribution of the points along an axis shows us the mean and standart deviation of a category for a feature, visually. Each black point in the graphs represent a decision given by an image which is in the same category with the query image, each red point is vice versa.

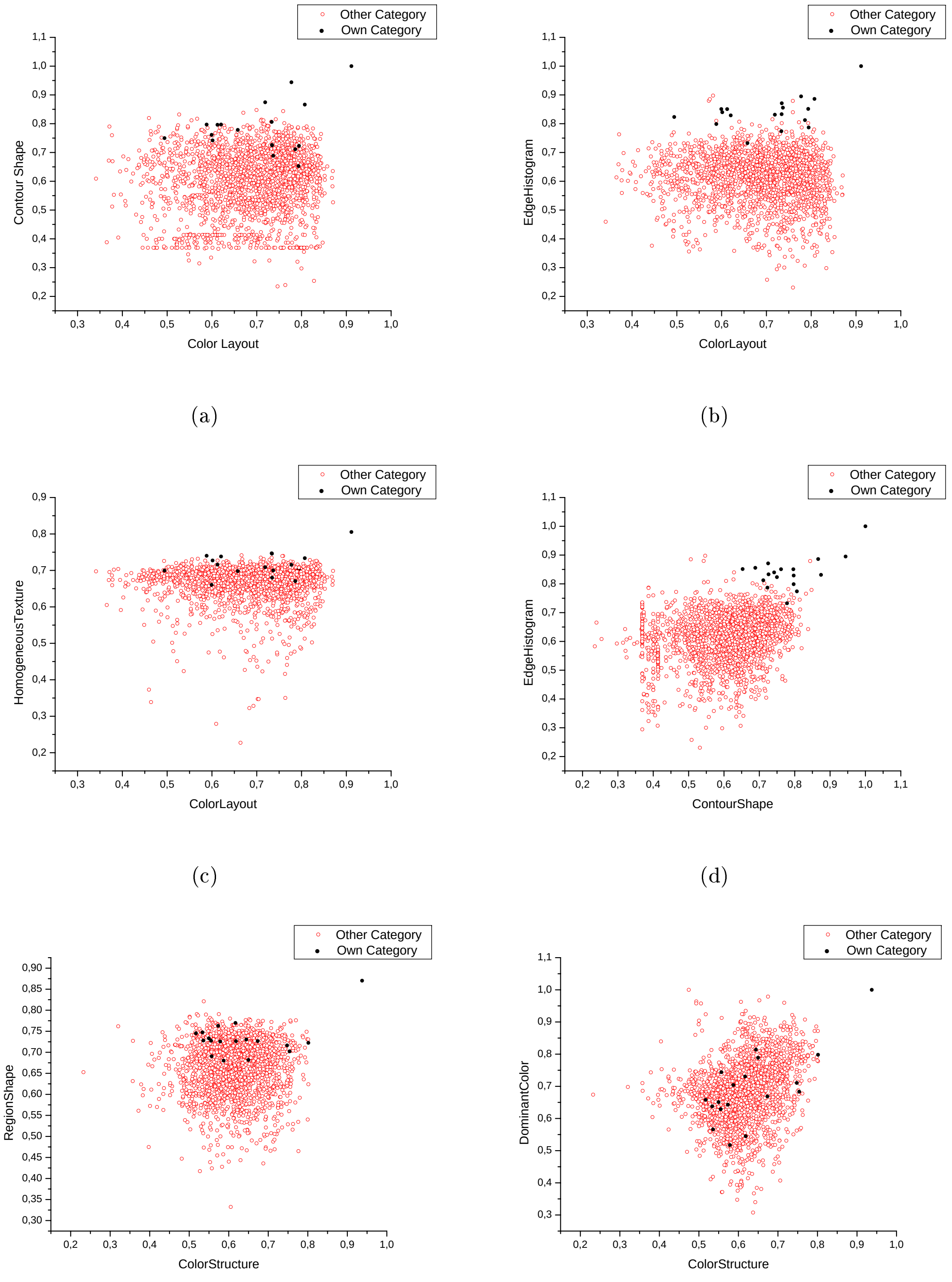
In Figure 4.3(a) and Figure 4.3(c), it is seen that Contour Shape and Homogeneous Texture features are better than Color Layout since their distribution along y axis are in a narrow range, but the distribution along x axis has broad range, for Own Category data. With the same sense, in Figure 4.3(f) both descriptors are bad for representation and in Figure 4.3(d) both are good.

4.7 Genetic Algorithms Based Classifier

As expressed before, the system uses a set of features to make decisions on the images. Instead of getting the average values of the features from MPEG-7 descriptors of samples and using these average values for defining the object types (object classes); we store a set of feature values and use a genetic algorithm mechanism to make the set more qualified.

The object classification task can be defined as assigning an object s_i to an object class c_j by approximating a function $\Phi' : S \times C \rightarrow \{T, F\}$ by maximizing the coincidence of Φ' with the actual categorization Φ, where $S = \{s_1, \ldots, s_n\}$ is the set of objects, $C = \{c_1, \ldots, c_m\}$ is the set of classes and $\{T, F\}$ are the Boolean values for true and false [54].

The core of the genetic algorithm approach is the idea of "survival of the fittest" and the approach suggests solving the search or optimization problems by allowing fittest (more successful) solutions to live more and other to die. A reproduction mechanism creates new (offspring) solutions by combining fittest parent solutions. Using crossovers and mutations during the reproduction increases the variety of offspring solutions [3].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4.3: Feature Importance Comparison On XM Software Results

In classification problems, the classifier may assign the object to a single class or multiple classes. Here, we consider multiple class approach that means we learn decisions of the categories and assign the object to the categories with a correctness value.

4.7.1 Representation and Decision-Making

Considering the multiple categorization, the problem is classifying an object S to some categories. To decide on this classification, the principals of natural genetic is applied on the problem according to the below given principals.

Principle 1 *Each class C (in other words, category) has a set of chromosomes to decide on whether a given object S belongs to C, with which correctness ratio. The set of chromosomes is given by $X_C = \{\chi_1, \chi_2, \ldots, \chi_n\}$ where χ_i denotes a single chromosome of the class. A chromosome is gained by a class by identifying a new object in the training phase, in other words classifying each object during training makes the class gain a new chromosome.*

Principle 2 *For the decision making of a class, decisions of all chromosomes of the class are used according to the following formula:*

$$H_C = \sum_{i=1}^{n} \left(\frac{\eta_i}{\sum_{j=1}^{n} \eta_j} \cdot h_i \right) \tag{4.5}$$

where H_C is decision of class C, h_i is the decision of chromosome χ_i from class C and η_i is the effectiveness of chromosome χ_i on the decision of the class C.

Principle 3 *Each chromosome χ_i has a set of genes to give the above decision. The set of genes is given by $G_i = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$ where γ_j denotes a single gene of the chromosome χ_i. Each gene γ_j has three properties $\gamma_j(f_j, v_j, imp_j)$; f_j is a feature of the object from MPEG-7 descriptors which is unique for chromosome χ_i, v_j is the value of the feature f_j and imp_j is the importance of the feature that is detailed in Sub-section 4.6.2. All imp values of a feature f for an object category are equal; imp is the representation constant of the feature for an object category. All chromosomes in a class have the same number of genes and also the features represented by the genes are the same. In other words, each feature of the class is represented by a gene in the chromosomes and genes have fix positions on the chromosomes.*

Principle 4 *For the decision making of a chromosome, decisions of all genes of the chro-*

mosome are used according to the following formula:

$$h_i = \sum_{j=1}^{m} \left(\frac{imp_j}{\sum\limits_{k=1}^{m} imp_k} \cdot SIM(f_j, v_j, S) \right) \tag{4.6}$$

where h_i is the decision of chromosome χ_i , m is the number of genes in the chromosome and $SIM(f_j, v_j, S)$ is the similarity function.

Principle 5 *The similarity function $SIM(f_j, v_j, S)$ checks whether the feature f_j of object S is similar with the value v_j and gives a similarity value between 0 and 1 as the result. In other words, in Principle 4, $SIM(f_j, v_j, S)$ is the decision of gene γ_j on the given object S. The function calculates the result with the help of Feature Comparison module.*

These principles can be summarized as the following; genes of a chromosome determine the decision of their related chromosome and all chromosomes determine the final decision of the class. In other words, chromosomes cooperate to make a decision for the class, although they compete for appearing in the next generations and becoming more dominant on decision giving process.

The decisions of categories are given in a fuzzy manner in order to overcome incorrect results coming from fuzzy domains. That means, the decision of a class C on an object S, $Decision_C(S) = H_C$, is a value between 0 and 1 and gives the correctness ratio of the object S for the category C.

Representation of a sample category is given in Figure 4.4. The category is 'car_side' and the BRDF model contains three features with corresponding importance values: Scalable Color (0.36), Contour Shape (0.37), Edge Histogram (0.27). Each chromosome has effectiveness value of its own and genes for each feature thas is meaningful for the category. Each gene contains the feature and an image for the value of the feature.

4.7.2 Initialization of GA

To perform genetic operations, obtain genetic diversity and optimisation for the problem, there should be an initial population of individuals. In Sub-section 3.4.1, it is stated that the population can be generated randomly or seeded by a training data. This study proposes to obtain the initial population by using the First-Training data as mentioned in Sub-section 4.3.2.

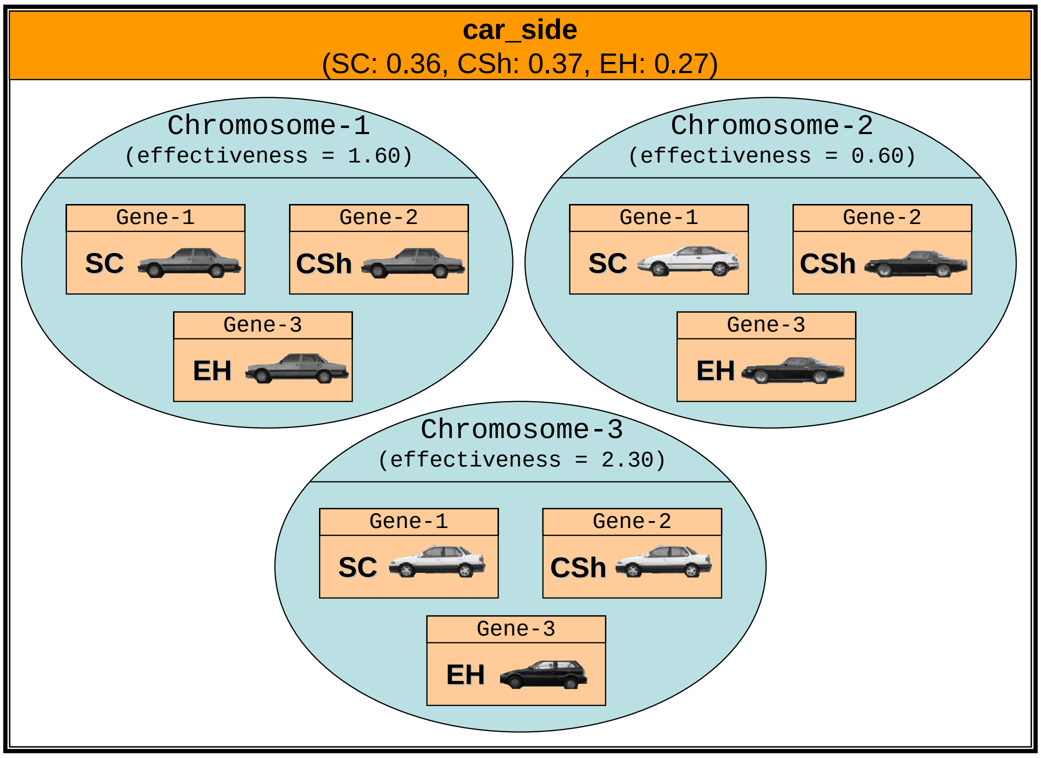


Figure 4.4: Representation of A Sample Category

4.7.3 Use of Genetic Operations

The genetic operations are the core of a GA and a GA provides more fitting individuals by genetic operators. In this study the genetic operations are applied to the system, during Second-Training. The iterations are given in Algorithm 2.

The operations “Effectiveness Correction”, “Crossover”, “Mutation” mentioned in Algorithm 2, lines 3, 7 and 8, are described in detail in Sub-section 4.7.4, Sub-section 4.7.5 and Sub-section 4.7.6 respectively.

4.7.4 Fitness Function

The fitness function is used to see the accuracy of the results and apply the most important genetics principal “survival of the fittest” to the system. It is used for understanding how much a given decision is fit.

In GA studies, fitness function is usually used during crossover operation to decide which chromosomes/individuals are successful according to ground truth. In this study, the fitness function has two roles; the first one is to determine the success of chromosomes, the second one is to assign effectiveness values to chromosomes according to their success on the ground

Algorithm 2 Genetic Operations Algorithm

```
Input :   Second-Training images, Classifier
Output :   Genetic Operations applied Classifier
for all images given in Second-Training do
    Calculate decisions of all categories on the image
    Perform Effectiveness Correction on chromosomes of all categories
end for
for all real categories of images given in Second-Training do
    Obtain images of category that are given in Second-Training
    Perform Crossover for category with parameter list of images
    Perform Mutation for category with parameter list of images
end for
```

truth. This second role strengthens the idea of "survival of the fittest" in the system. As the ground truth, the Second-Training data and inputs are used. During Second-Training, like First-Training the images (also, objects selected in the images) are given to the system with the correct classification of the objects. Below, C_S is used for the correct classification that is given to the system.

The fitness value of a chromosome is calculated according to the decision of the chromosome without multiplying with the effectiveness value of the chromosome. Also whether the chromosome decision can be accepted as correct is another important criteria for the fitness value. So the fitness value of a chrosome χ_i can be calculated as;

$$Fitness_{\chi_i} = \begin{cases} h_i, & h_i \geq thr_{GA} \wedge \chi_i \in C_S \\ 0, & otherwise \end{cases} \tag{4.7}$$

where thr_{GA} is the threshold for genetic operators correct acceptance. If more than one image is used during the Second-Training process, the fitness value is the sum of all.

To strengthen the mechanism of "survival of the fittest", the effectiveness η_i of the chromosomes that gives the same decision h_i with correct classification C_S, is increased by a factor κ. This is called as "Effectiveness Correction".

$$\eta_i^{t+1} = \begin{cases} \eta_i^t + \kappa_{own}, & h_i \geq thr_{GA} \wedge \chi_i \in C_S \\ \eta_i^t + \kappa_{oth}, & h_i < thr_{GA} \wedge \chi_i \notin C_S \\ \eta_i^t - \kappa_{own}, & h_i < thr_{GA} \wedge \chi_i \in C_S \\ \eta_i^t - \kappa_{oth}, & h_i \geq thr_{GA} \wedge \chi_i \notin C_S \end{cases} \tag{4.8}$$

The two different κ values are used for chromosomes in the correct class of the object and in other classes (κ_{own} and κ_{oth} respectively) in order to provide different effects for Second-Training images with the same category and other categories. The effect of an image belonging to the category of the chromosome should be more effective than the effect of images belonging to other categories on increasing or decreasing the effectiveness value.

By increasing the effectiveness of the fittest chromosomes and decreasing the effectiveness of others, in the long run, the chromosomes which vote for incorrect classification lose their existence and fitting chromosomes get more effective on the resulting $Decision_C(S)$ of the class.

4.7.5 Crossovers

Crossover is the feature interchange between two chromosomes. It occurs during the mating of two different chromosomes. For the mating process, in each turn, two randomly selected chromosomes are used. But the probability for participating in mating are direct-proportional with the fitness value of the chromosome described in Sub-section 4.7.4.

As seen on Algorithm 2, crossover is performed for only real category of each image, whereas effectiveness correction is performed on all categories for each image.

Let χ_a and χ_b are parent chromosomes and they produce χ_c and χ_d after the crossover. So the effectiveness rates are η_a, η_b and the genes of the chromosomes are G_a, G_b in advance. Then, η_c and η_d is found as the average of effectiveness rates of parent chromosomes χ_a and χ_b: $\eta_c = \eta_d = (\eta_a + \eta_b)/2$. The new genes G_c and G_d of the new chromosomes χ_c and χ_d are calculated according to the Algorithm 3. The algorithm is written to be applied on a selected category, with Second-Training images of the selected category is given as parameter.

For a particular category, how many tries for crossover operation is performed (how many generations are produced) is determined according to how many images are obtained during Second-Training on the selected category. In other words, the termination condition of producing new generations is based on the count of images obtained. The selection on the number of generations is determined considering that each new Second-Training image is used as a new information source for the system and used to make improvement on the system. But it should be noticed that this is an assumptive selection and can be increased or decreased in order to achieve better performances. For each try, firstly, $Fitness$ values of the chromosomes of the category is calculated. On each try, an empty set of chromosomes tried to be filled by performing crossovers until the size of the set achieves the size of the category. A 'try' means reproducing a new generation. So it is obvious that the size of

Algorithm 3 Crossover Algorithm

Input : Category (*category*), Second-Training images in *category*, Classifier

Output : Crossover applied Classifier

```
for all $images$ do
                ▷ Try crossover for number of Second-Training images in $category$ times
    Calculate $Fitness$ of chrososomes in the $category$
    Define $NewChromosomes$ set as an empty set
    while $size(NewChromosomes) < size(category)$ do
        Select 2 random chromosomes according to $Fitness$
        Select crossover genes randomly.
        Perform gene interchange between $\chi_a$ and $\chi_b$; obtain $\chi_c$ and $\chi_d$
        Calculate $Fitness$ of $\chi_c$ and $\chi_d$
        if Both $Fitness_{\chi_c}$ and $Fitness_{\chi_d}$ are better than $Fitness_{\chi_a}$ and $Fitness_{\chi_b}$ then
            Add $\chi_c$ and $\chi_d$ to the set $NewChromosomes$
        else
            Add best of $\chi_a$, $\chi_b$, $\chi_c$ and $\chi_d$ to the $NewChromosomes$ according to $Fitness$
        end if
    end while
    Calculate average decision of $category$ on $images$ as $CatDec$
    Calculate average decision of $NewChromosomes$ on $images$ as $NewDec$
    if NewDec > CatDec then
        Remove all chromosomes in $category$
        Add all chromosomes in $NewChromosomes$ to $category$
    end if
end for
```

individuals does not change through the next generations, the algorithm guaranties to have a constant number of individuals during the life-cycle.

To obtain a new generation, the algorithm requires performing mating between randomly selected pairs of parents until the required size is achieved. For each mating, two random chromosomes are selected, by using a roulette-wheel selection mechanism. Then a uniform crossover mask is obtained randomly. By using the crossover mask, the crossover operation is performed. After obtaining new chromosomes, the fitness values of two new chromosomes and two old chromosomes are compared. If both of new chromosomes are better than the old ones, both of them are added into the chromosome set. Otherwise, only the best of four chromosomes is added into the set. This means a steady-state selection mechanism is used and parent individuals can occur on the next generation. This mechanism also guarantees not to have worse individuals, at worst case the same individuals with the parents are obtained as the next generation.

After achieving the required size, the chromosome set is thought as a category and decisions of this temporary category are calculated on the Second-Training images. If average decision is better than the decision of the original category, all chromosomes are removed from the category and the chromosomes in the set are put into it. Otherwise the set is discarded. This action also guarantees not to have worse decision giving category after each iteration.

Although the crossover algorithm given in Algorithm 3 seems more complex than the simple GA given in Algorithm 1, the flow and operations are similar. The loop between lines 3 and 8 on Algorithm 1 corresponds to the loop between lines 1 and 21 of Algorithm 3. Also the selection and reproduction operations in the loops are clearly similar.

By using above algorithm, considering the long run, it can be stated that the chromosomes which votes for incorrect classification can neither mate nor effect the resulting decision much and lose their existence.

4.7.6 Mutation

In traditional GA, mutation is an infrequent event that randomly changes the information that the genes have. But in this approach, mutation process are used for learning new information from newly encountered images. With an analogy to the fact that mutation is caused by the conditions of environment, we can use mutation for learning the new information in the environment (Second-Training images). Actually, although the information that Second-Training images contains are as valuable as the First-Training images, without mutation we do not use them directly, we use only for comparisons. If they are so valuable, we should import information that they have.

The mutation is applied according to the Mutation Algorithm given in Algorithm 4. Like crossover, mutation is also performed for only correct category of each image, as seen on Algorithm 2.

Algorithm 4 Mutation Algorithm

```
Input :   Category (category), Second-Training images in category, Classifier
Output :   Mutation applied Classifier
for all images do
                        ▷ Try mutation for number of Second-Training images in category times
    Calculate $Fitness$ of chrososomes in the category
    Select 1 random chromosome according to $Fitness$ : $\chi_{selected}$
    Set $ChrFitness$ as the $Fitness_{\chi_{selected}}$
    Select mutation genes according to Feature Importance Values
    Perform gene change on $\chi_{selected}$
    Calculate new $Fitness$ of $\chi_{selected}$ as $ChrFitnessMutated$
    if $ChrFitnessMutated > ChrFitness$ then
        Make changes during mutation permanent
    else
        Discard mutation changes
    end if
end for
```

For a particular category, mutation is performed for each image obtained during Second-Training on the selected category. Since each image obtained in the First-Training imported into the system as a new chromosome, the images in the Second-Training can be thought as outer-chromosomes. In this aspect of view, the mutation can be assumed as a crossover between an outer-chromosome and a selected (inner) chromosome. But mutation has special parameter for determining interchanging genes. Using a parameter mf (mutation factor), with which ratio the features of the image is imported into the chromosome is defined. After performing gene change operations, the $Fitness$ values of old state and new state of the chromosome is compared. When mutation causes a decrease in the $Fitness$, it is discarded.

CHAPTER 5

EMPRICAL STUDY

After describing the entire system, this chapter expresses the emprical studies on the system. Organization of the chapter is as follows: Firstly, implementation details are given and the dataset used in the tests are introduced. Then the tests performed on the implemented system is given with the results. Lastly the evaluation of the test results comparison with other systems are stated.

5.1 Implementation Details

The implementation of the system is carried out with a component oriented approach. Some of the required modules are directly used from other researches, some are originally implemented, some are combination of two. Details are given below in each module's definition.

The system is aimed to be platform independent and designed with parts of a functional application which is capable of being run on many different server platforms and a browser-based Graphical User Interface (GUI) (Figure 5.1). All the functional components which are the modules shown in Figure 4.1 should run independently from the platform. Considering this requirement and also the fact that a convenient system should provide the user a single entry point to perform all of the processes as a single process for simplicity, it is aimed to hide the relations between modules and also external components. Concretely, in the expected system, the user only gives the video or image to the system and gets the results, he/she does not deal with the IBM Software, XM Software, Ncut segmentator or the classifier, also do not know when to extract keyframes, make segmentation and perform classification.

The implementation mostly achieved to provide the needs of the expected system. For image queries, the user faces a single entry point. For the video case, the integration of keyframe extraction can not be adapted to the system, the user should perform the keyframe

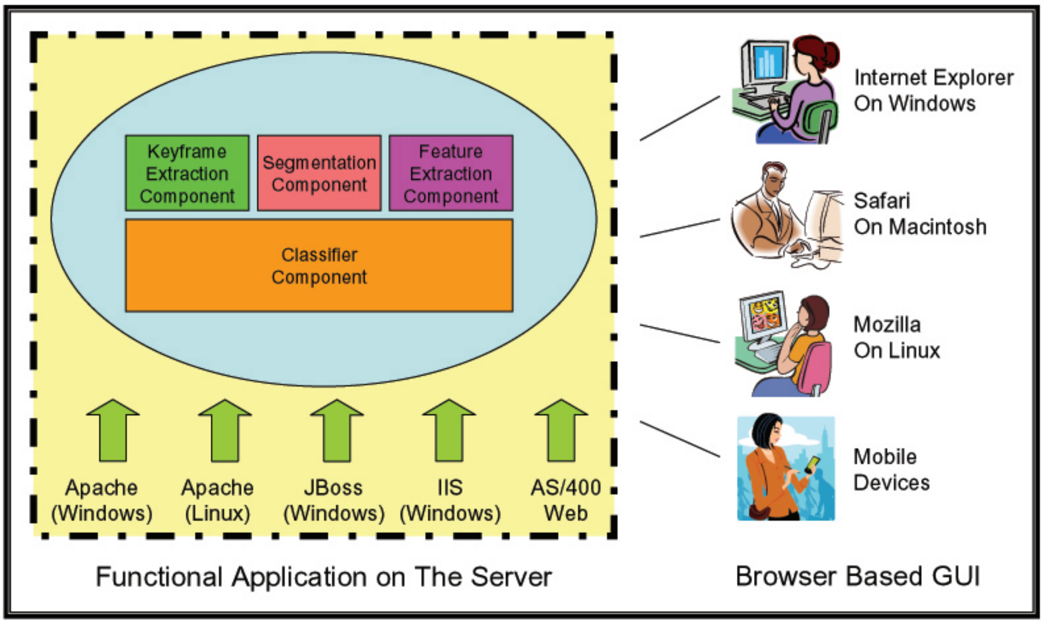


Figure 5.1: Platform Independent Design of the System

extraction first, then give the resulting keyframes to the implemented system. Details of the state is given in Sub-section 5.1.1.

The implemented system can be analyzed in two parts as mentioned above, Functional Application and Browser Based GUI. To provide the platform independency, the main flow of the system and the modules that are not outsourced are implemented in Java.

5.1.1 Functional Application

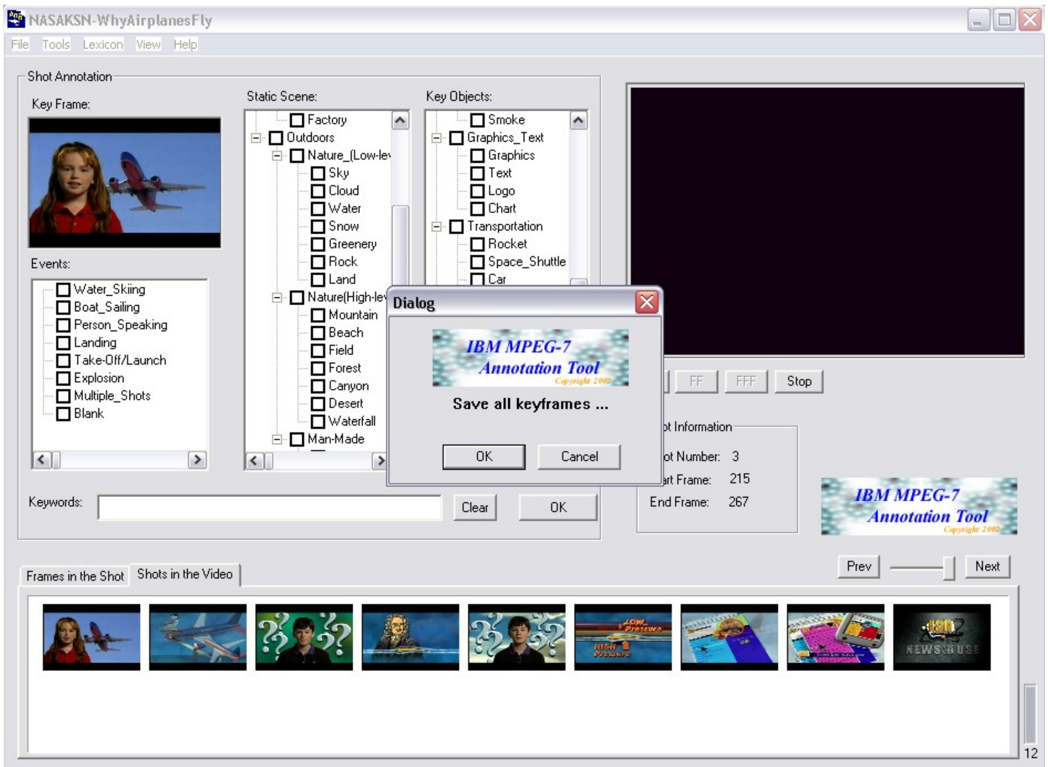
Functional application is the core of the system that performs all operations like extraction, segmentation and classification. The relations between modules of the system is told in Chapter 4. The implementation details, problems and assumptions on the modules during implementation are given below. In Figure 4.1, all these modules are also shown with the relations between them.

Keyframe Extraction Module

As declared in Section 4.4, keyframe extraction functionality of IBM MPEG Annotation Tool [33] is used for keyframe extraction. This tool is a standalone application and does not provide an API (Application Programming Interface) for extracting keyframes of videos and also it only runs under operating Systems of Microsoft Windows® 95 or above, that means it is platform dependent. Due to these restrictions, the implemented system cannot directly perform keyframe extraction from videos. So when a video is to be processed, firstly it is subjected to keyframe extraction by using IBM tool, and then the resulting keyframes are given to the implemented system.

Although this situation is different from expectations, it does not have a major effect on the operationality of the system. Considering the fact that the core of this study is not the keyframe extraction, this case is left as a minor problem for future work.

In Figure 5.2 a screenshot from IBM MPEG Annotation Tool is given. It is taken while a keyframe extaction is being performed. At the bottom part of the screen, the keyframes of the video is shown.



Dialog

Save all keyframes ...

ot Number: 3
art Frame: 215
End Frame: 267

Figure 5.2: Screenshot from IBM MPEG Annotation Tool

Segmentation Module

Segmentation module takes an image as input and returns multiple image files each of which is a segment or a segment group consisting of neigbouring segments. In the module, firstly in the Java implementation of main flow, the segmentation function is called which is the MATLAB implementation of Cour et al. [11]. For such a call, the MATLAB function is converted to an executable by MATLAB compiler (mcc). This function takes the image and requested number of segments as parameter and gives a segmentation map on the image. The choice for requested number of segments can be left to the system or the user can define it via the user interface. Then with a Java implementation, the segmentation map is interpreted and corresponding candidate objects (segments or segment groups) are exported as PNG (Portable Network Graphics) images with the brute-force approach described in Section 4.5. The exported images are in type PNG since it provides lossless data compression and enables tranparency in the images. Tranparency is important for the segment images because the residuary parts that should be transparent can cause problems during the feature extraction and comparison.

Both Java application and MATLAB procedure can be executed on different platforms, so this module does not have the problem of adaptation to the system.

Feature Extraction Module

As described before, feature extraction is performed by using XM Software [41]. XM software is distributed as C++ code and provides a command line based feature extraction and search & retrieval system. By using specific compilers for different platforms, it possible to execute the XM software on different platforms.

In the module, the main flow in Java implementation prepares the parameter file, image list file and image files for the XM Software automatically and calls the feature extraction methods of XM software for chosen features. The call for executables of XM Software is performed via implementing JNI (Java Native Interface) classes in Java. The first choice for such calls was sending commands to command prompt over Runtime object of Java, but due to the time inefficiency of this method, it was backed down. The change halved the average execution time.

XM Software stores the extracted features in binary or DDL files. This file is also given to the XM Software as parameter, so the result file of XM Software is stored in Data Repository sub-module of Classification module.

The general design of the proposed system requires storing the extracted features from First-Training images and also Second-Training images that are used in mutation. But XM Software has a restriction for the storage files; every time an extraction is performed on XM Software, it renews the storage file. Adding into a storage file or combining the multiple ones is a risky attempt that can cause corruption. So as an implementation decision, all images in both First-Training and Second-Training are given to the XM Software at the beginning. This does not make a confusion during the decision making of categories because every category knows which images encountered until that time. (e.g. At the end of the First-Training, the chromosomes of the categories knows only First-Training images and know nothing about the Second-Training images.)

The implementation of all features (MPEG-7 descriptors) is not carried out. This is not because of the difficulty of implementing features, it is easy and always possible to new feature implementations to the system. But the increase of implemented features makes the feature extraction and feature comparison times longer. So the meaningful features with the concept of this study, which are color, shape and texture descriptors, are implemented. Besides, adding new feature implementations does not change the results of this study. Because most of the objects are defined with color, shape and texture descriptors. If a new descriptor which is less important than these is included into the system, the feature importance determining process identifies it and gives a low BRDF index to it and the result is not affected. If a new descriptor which is more important than currently implemented ones, this makes system to give better results.

Feature Value Normalization Module

Feature value normalization is implemented in Java according to the rules given in Sub-section 4.6.1. The implementation guarantees to use only First-Training images for normalization.

Feature Importance Determination Module

Feature importance determination is implemented in Java according to the rules given in Sub-section 4.6.2. The implementation guarantees to use only First-Training images for normalization.

Classification Module

Classification module consist of three sub-modules:

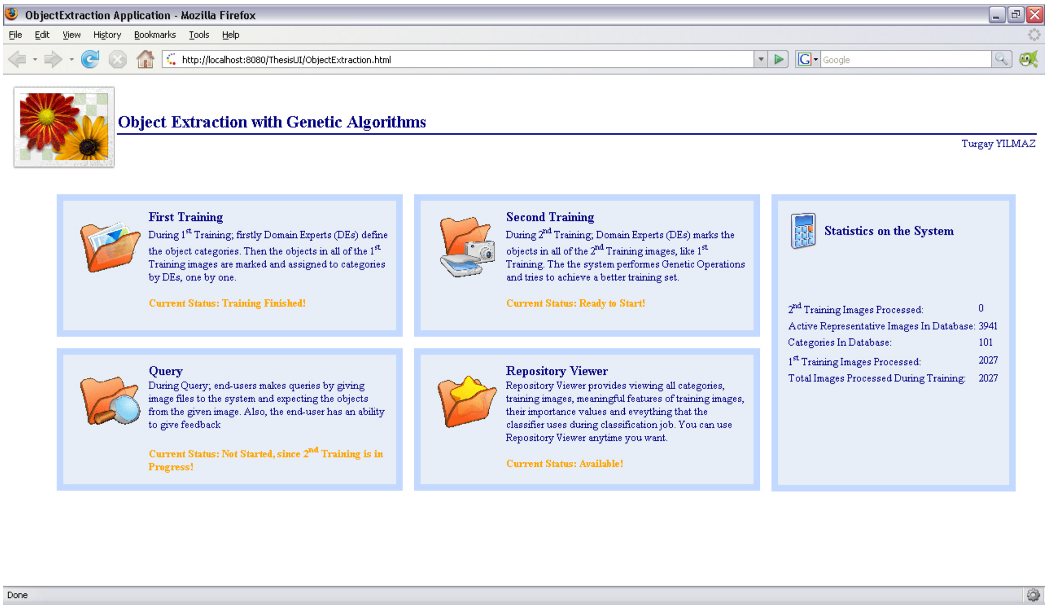
- Feature Comparison Sub-Module: Feature comparison is performed by using XM Software [41]. The main flow in Java implementation prepares the required parameters and files automatically and calls the search & retrieval methods of XM software via JNI classes, like in Feature Extraction Module. XM Software search & retrieval method takes the image list and feature-store file that is generated in feature extraction, these files are taken from Data Repository sub-module. The method, then gives comparison results as distance values. The Java implementation of the module performes the normalization process according to the calculations given in Sub-section 4.6.1 and converts the distance values to similarity values with a 1's complement operation. The obtaining result is a raw information and contains more information than needed for calculation of category decisions, it is sent to the GA-Based Classification sub-module to be used in the decision making.

- GA-Based Classification Sub-Module: The module is implemented in Java with the structure given in Section 4.7. By using the similarity values calculated in Feature Comparison sub-module, the decisions of the categories are calculated. Also, the genetic operations are implemented under this sub-module. Although it is seen as a sub-module, the most important operations are performed here. The implementation of the structure is carried out in such a way that every gen knows the feature and an image file that it represents so that only necessary information is taken from tha raw information of Feature Comparison sub-module. At the end of the First-Training, the implementation guarantees that a category contains a number of chromosomes that is equal to the first-training images encountered and each chromosome has genes with number of implemented features. Each gene represents a different feature and all genes of a chromosome represents the same image. After crossovers and mutations, the situation tends to change, crossover changes the genes between chromosomes, mutation adds one or more representing images to the genes. When a gene represents more than one images after mutations, the best result of them is used from feature comparison, according to the implementation. Another important detail on GA Classifier is the Fitness Function. Two different implementation of fitness function is provided such that one enables to choose according to a given threshold value, the other one according to the rank of the result. In thresholding method, a category or a chromosome is accepted as correct if its fitness value is higher than the given threshold. In ranking method, a category is accepted as correct if the rank of the category according is in the acceptable ranks range among other categories and a chromosome is accepted as correct if its fitness is better than the average of all chromosomes.

- Data Repository Sub-Module: Data Repository sub-module stores the training image lists, images, feature files extracted from XM Software and GA based classifier structure. All data is stored in files directly due to the incomplexity, a database system is not necessary. GA based classifier structure is stored by serializing Java objects implemented, other data is stored in text files. According to the implementation decisions, the image lists contain all images from both First-Training and Second-Training but the serialized GA based classfier structure only knows what is encountered so far.

5.1.2 Browser Based GUI

A graphical user interface is prepared for all operations to be controlled by the user. As shown by a screenshot of the main page of the GUI given in Figure 5.3, the GUI provides following capabilities; performing First-Training, performing Second-Training, performing Query on the system and view the current state of data repository. Also there is an information part that gives statistics on the system, on the right side of the screen.


ObjectExtraction Application - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://localhost:8080/ThesisUI/ObjectExtraction.html
Google
Done

Object Extraction with Genetic Algorithms

Turgay YILMAZ

First Training

During 1st Training; firstly Domain Experts (DEs) define the object categories. Then the objects in all of the 1st Training images are marked and assigned to categories by DEs, one by one.

Current Status: Training Finished!

Second Training

During 2nd Training; Domain Experts (DEs) marks the objects in all of the 2nd Training images, like 1st Training. The the system performes Genetic Operations and tries to achieve a better training set.

Current Status: Ready to Start!

Statistics on the System

2nd Training Images Processed:	0
Active Representative Images In Database:	3941
Categories In Database:	101
1st Training Images Processed:	2027
Total Images Processed During Training:	2027

Query

During Query; end-users makes queries by giving image files to the system and expecting the objects from the given image. Also, the end-user has an ability to give feedback

Current Status: Not Started, since 2nd Training is in Progress!

Repository Viewer

Repository Viewer provides viewing all categories, training images, meaningful features of training images, their importance values and eveything that the classifier uses during classification job. You can use Repository Viewer anytime you want.

Current Status: Available!

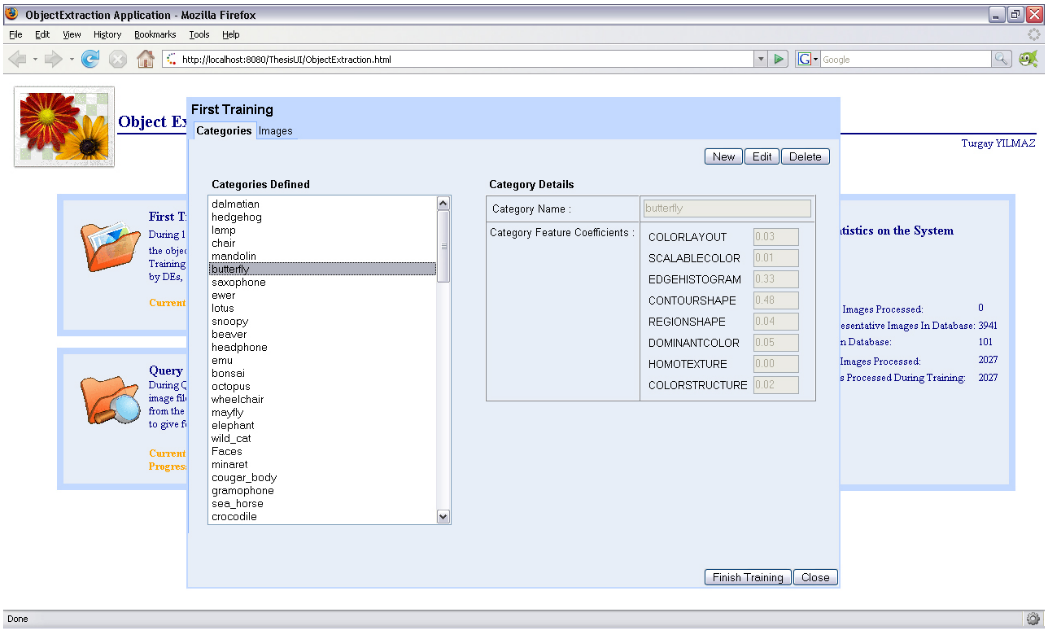
Figure 5.3: Screenshot from GUI, Main Screen

Since First-Training, Second-Training and Query phases are consecutive in the system, the GUI do not allow using all of them at the same time. Only Repository part is always usable, other parts are actived in order. Before finishing First-Training, Second-Training and Query are deactivated, after finishing it only Second-Training becomes actived. Query is active if only Second-Training is finished.

The contents of these four parts are given in detail below.

First Training Screen

First Training screen has two parts; the first one is category definition (Figure 5.4), the second one is adding new image to a particular category (Figure 5.5) and marking the object in it (Figure 5.6). In the category definition, new categories are defined for the system and then optionally the importance values of features are assigned. If they are not assigned, when First Training is finished (via button on the right bottom corner of Figure 5.4), they are automatically generated via Feature Importance Determination module (Sub-section 5.1.1).



Categories Defined

dalmatian
hedgehog
lamp
chair
mandolin
butterfly
saxophone
ewer
lotus
snoopy
beaver
headphone
emu
bonsai
octopus
wheelchair
mayfly
elephant
wild_cat
Faces
minaret
cougar_body
gramophone
sea_horse
crocodile

Category Details

Category Name : butterfly

Category Feature Coefficients :

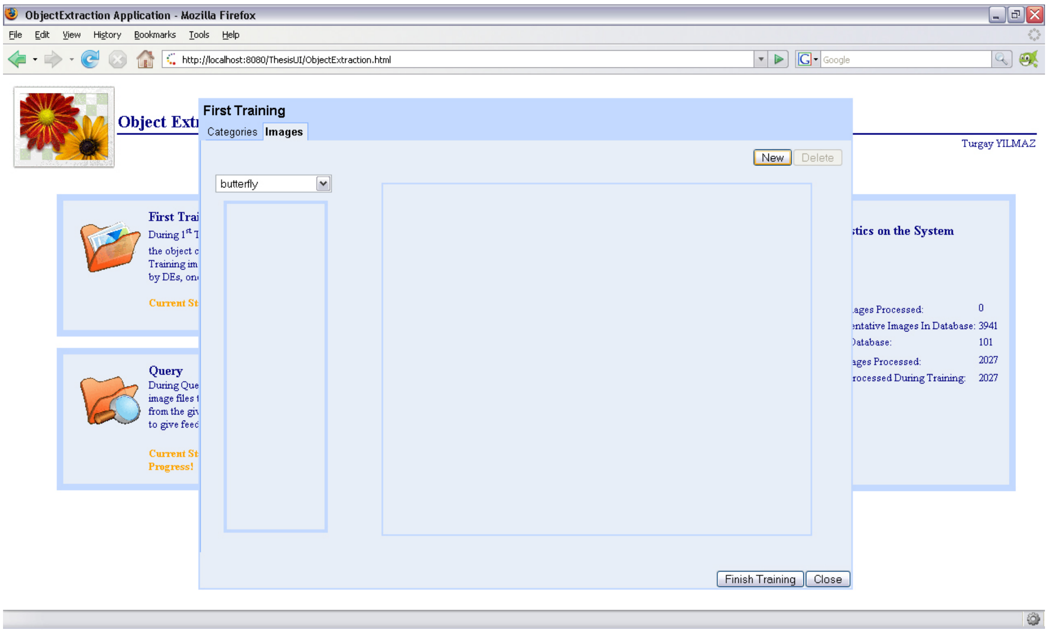
COLORLAYOUT	0.03
SCALABLECOLOR	0.01
EDGEHISTOGRAM	0.33
CONTOURSHAPE	0.48
REGIONSHAPE	0.04
DOMINANTCOLOR	0.05
HOMOTEXTURE	0.00
COLORSTRUCTURE	0.02

Images Processed:	0
esentative Images In Database:	3941
n Database:	101
Images Processed:	2027
s Processed During Training:	2027

Figure 5.4: Screenshot from GUI, First Training Screen, Category Definition

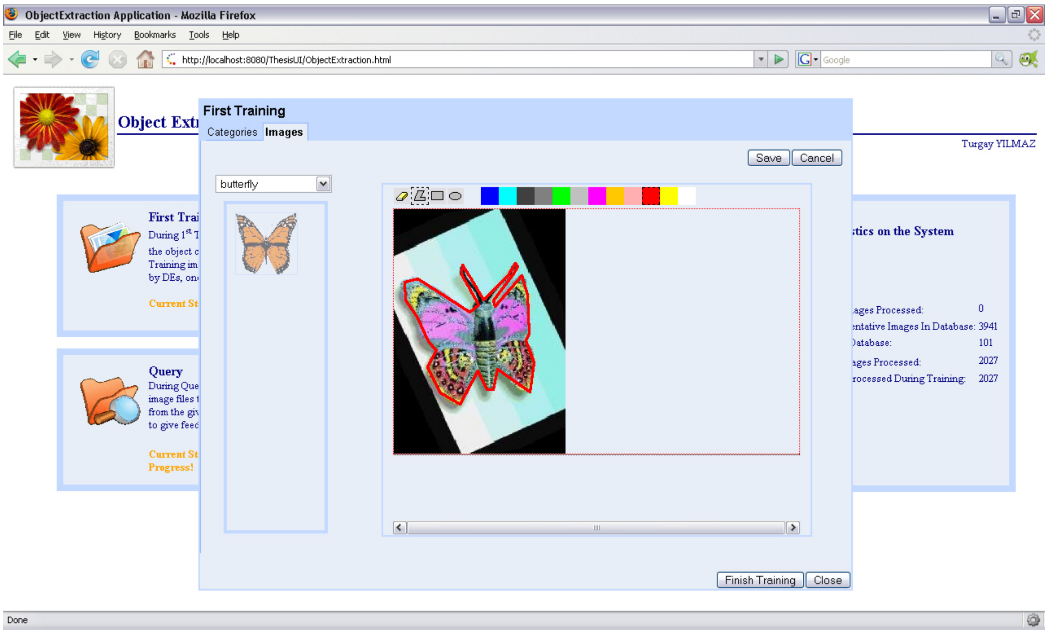
Second Training Screen

Second Training screen provides inserting new Second-Training images to the system (Figure 5.7). After choosing the image, the screen forces the user to mark the object on the image and select the category of the object(Figure 5.8). After finishing the Second-Training via the button on the right bottom corner of Figure 5.8, the genetic operations are applied on the repository according to the information the user gives.



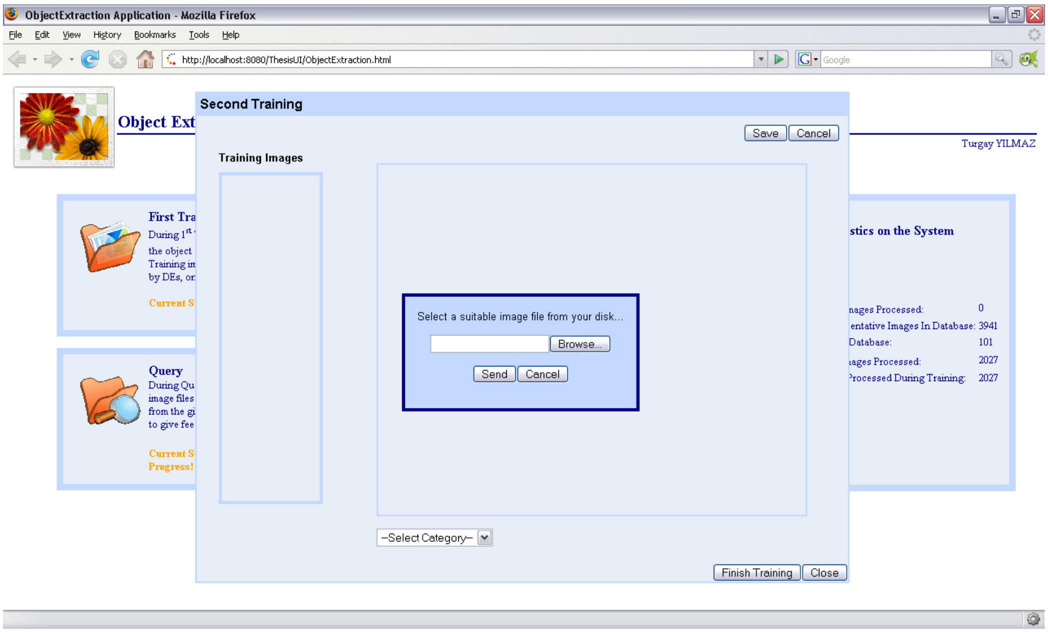
ages Processed: 0
ntative Images In Database: 3941
Database: 101
ages Processed: 2027
rocessed During Training: 2027

Figure 5.5: Screenshot from GUI, First Training Screen, Adding New Image



ages Processed: 0
ntative Images In Database: 3941
Database: 101
ages Processed: 2027
rocessed During Training: 2027

Figure 5.6: Screenshot from GUI, First Training Screen, Marking Object on New Image



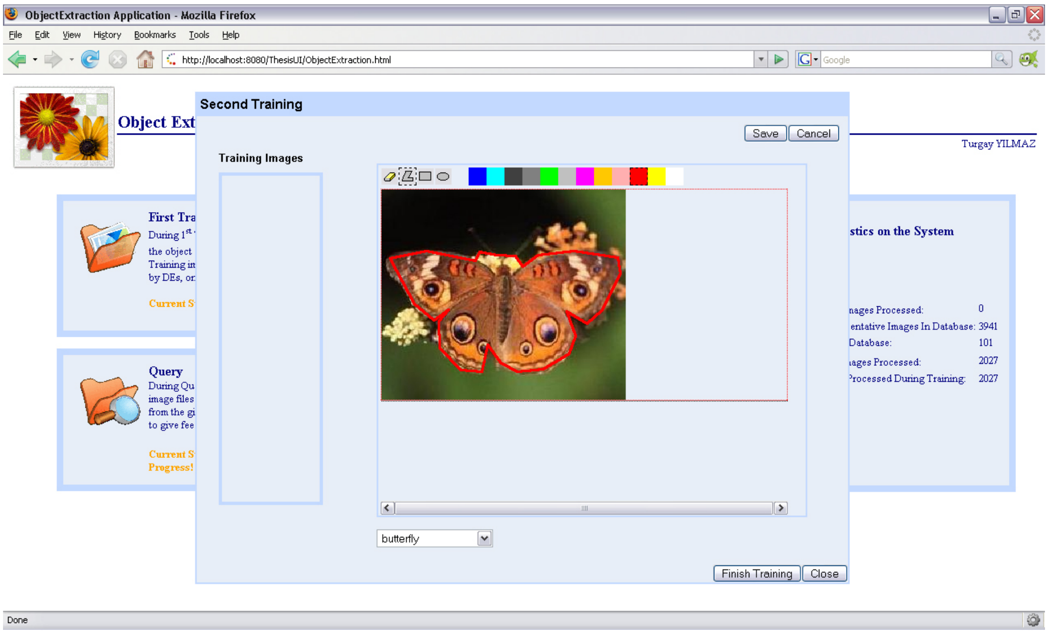
Select a suitable image file from your disk...

During 1st
the object
Training in
by DEs, or

During Qu
image files
from the gi
to give fee

nages Processed: 0
entative Images In Database: 3941
Database: 101
ages Processed: 2027
Processed During Training: 2027

Figure 5.7: Screenshot from GUI, Second Training Screen, New Image Selection



butterfly

During 1st
the object
Training in
by DEs, or

During Qu
image files
from the gi
to give fee

nages Processed: 0
entative Images In Database: 3941
Database: 101
ages Processed: 2027
Processed During Training: 2027

Figure 5.8: Screenshot from GUI, Second Training Screen, Marking Object on New Image

Query Screen

The Query screen enables user to search which objects occur in an image. For this purpose, fistly the user chooses the query image (Figure 5.9), then optionally defines the image complexity level (Figure 5.10) and performs the query via corresponding button on the screen. The image complexity level changes the desired number of segments from image so affects the query time and success; if 'Very Simple' is selected, query does not take a long time but the results obtained may be unsatisfactory, if 'Very Complex' is selected, the query takes longer but gives better results. The results are shown on the right side of the screen (Figure 5.11).

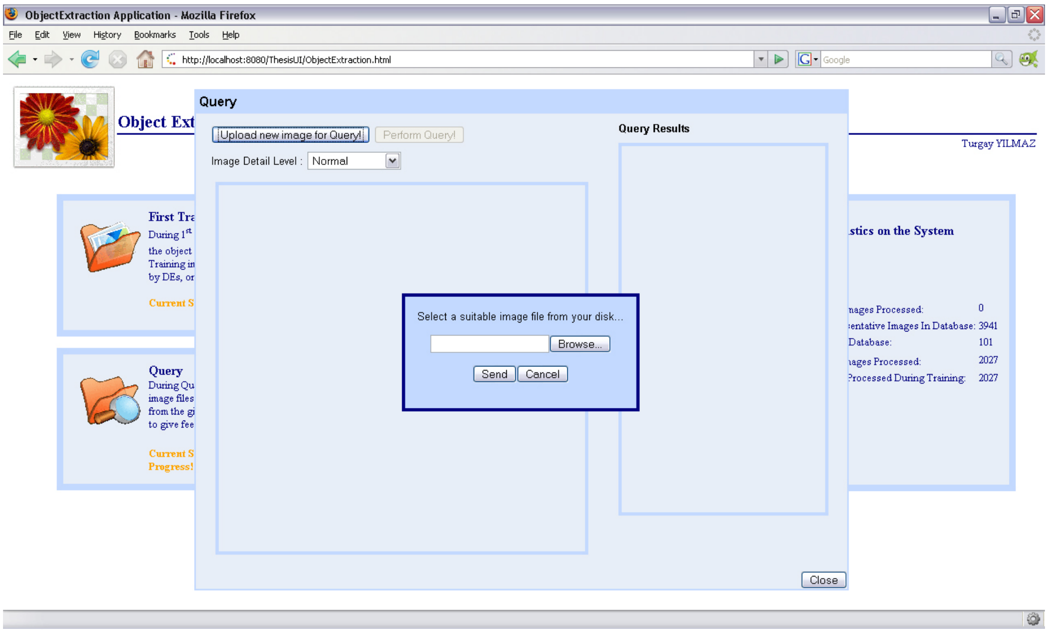


Image Detail Level : Normal

Select a suitable image file from your disk...

nages Processed: 0
entative Images In Database: 3941
Database: 101
nages Processed: 2027
Processed During Training: 2027

Figure 5.9: Screenshot from GUI, Query Screen, Selecting New Query Image

Repository Screen

Repository screen shows the contents of all categories, chromosomes and genes. When a category is selected on the screen, all chromosomes and genes are listed (Figure 5.12). When clicked on a gene (each small image corresponds to a gene), the detail of the gene is given (Figure 5.13). The detail of a gene is the image itself and MPEG-7 descriptor definition for that gene.

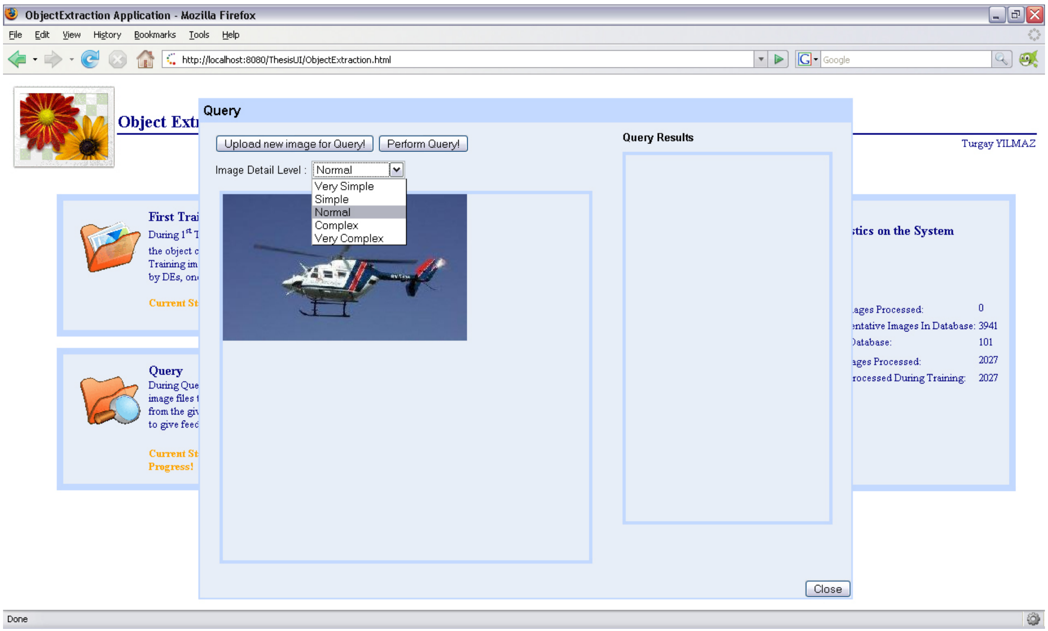


Image Detail Level : Normal

stics on the System

ages Processed: 0
entative Images In Database: 3941
Database: 101
ages Processed: 2027
rocessed During Training: 2027

Figure 5.10: Screenshot from GUI, Query Screen, Selecting Complexity Level

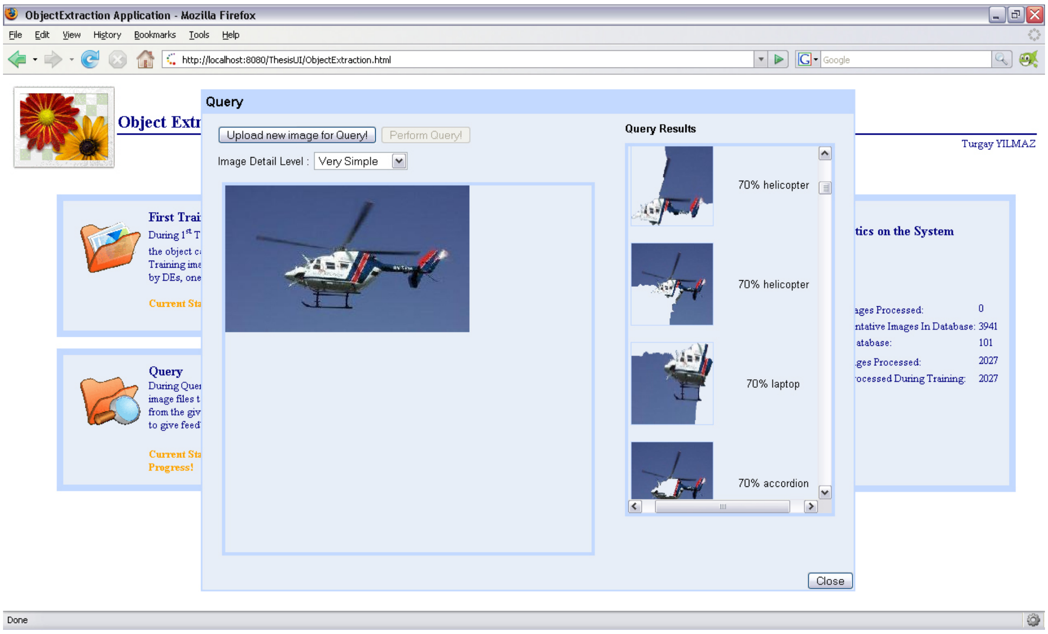


Image Detail Level : Very Simple

70% helicopter

70% helicopter

70% laptop

70% accordion

tics on the System

ages Processed: 0
ntative Images In Database: 3941
atabase: 101
ges Processed: 2027
ocessed During Training: 2027

Figure 5.11: Screenshot from GUI, Query Screen, Results Obtained

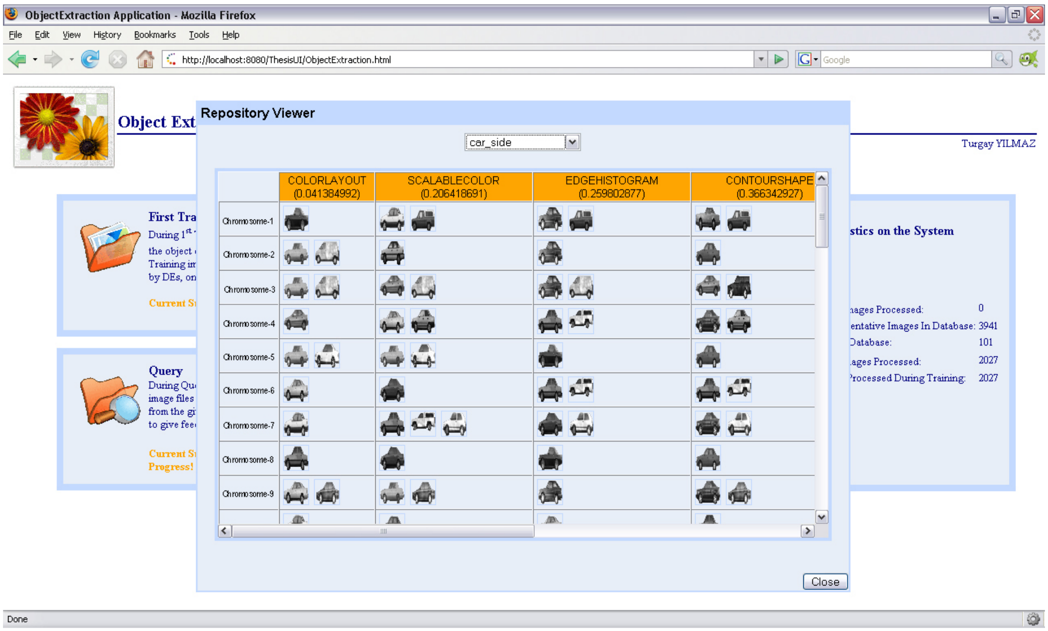
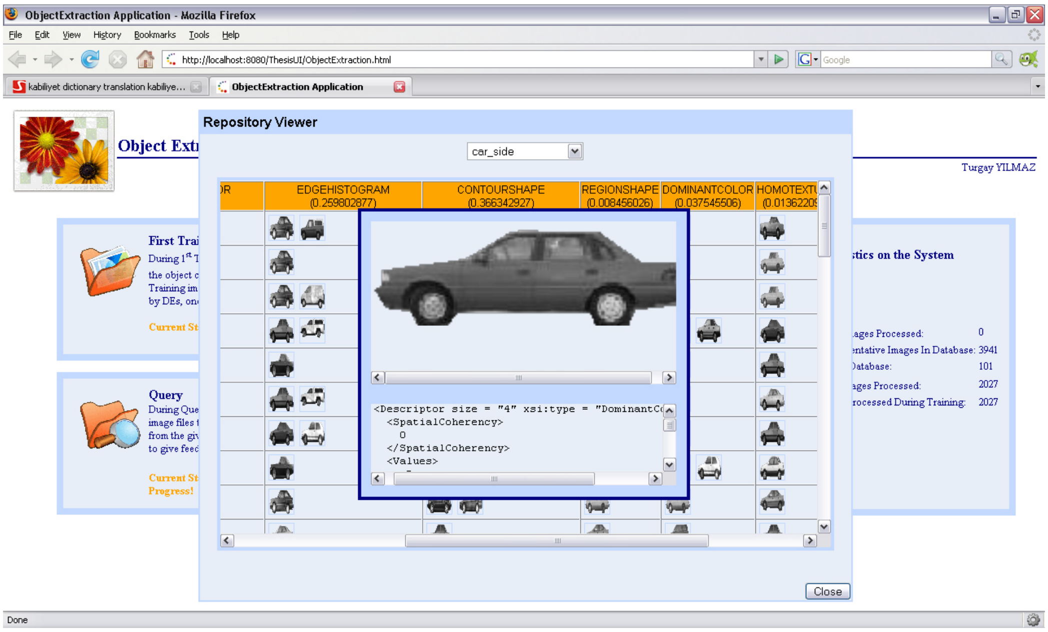


Figure 5.12: Screenshot from GUI, Repository Screen, Chromosomes of a Category


ObjectExtraction Application - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:8080/ThesisUI/ObjectExtraction.html
Google
kabiliyet dictionary translation kabiliye...
ObjectExtraction Application
Object Extr
Turgay YILMAZ
Repository Viewer
car_side
OR
EDGEHISTOGRAM (0.259802877)
CONTOURSHAPE (0.366342927)
REGIONSHAPE (0.008456026)
DOMINANTCOLOR (0.037545506)
HOMOTEXTU (0.01362209
First Trai
During 1st T
the object c
Training im
by DEs, one
Current St
Query
During Que
image files t
from the giv
to give feed
Current St
Progress!
stics on the System
ages Processed: 0
entative Images In Database: 3941
Database: 101
ages Processed: 2027
rocessed During Training: 2027
Close
Done

```
<Descriptor size = "4" xsi:type = "DominantC
  <SpatialCoherency>
    0
  </SpatialCoherency>
  <Values>
```

Figure 5.13: Screenshot from GUI, Repository Screen, A Gene Detail

5.1.3 Other Facts on Implementation

Other facts on implementation is as follows:

- The implementation is developed on a Windows XP, Centrino Duo 1.66 GHz, 1 GB RAM machine. Also all test are performed on the same machine.
- The Java implementation is developed with Java Development Kit (JDK) version 1.5.0_9 on IntelliJ IDEA 6.0.4
- The Java implementation contains 9486 lines of code in 102 files.
- As the web server of the application, JBoss AS 4.2.1.GA is used.
- The user interface is developed on Java by using library Google Web Toolkit Version 1.4.59.
- Following Java libraries are used:
  - JMatIO Java Library v0.2, for reading MATLAB files from Java
  - Apache Jakarta Commons FileUpload Library v1.2, for uploading file into the web server
  - Apache Jakarta Commons IO Library v1.3.2, for file input output with the web server
  - Jakarta Commons Math Library Version 1.1, for statistical calculations
- In the browser based UI, High Performance JavaScript Graphics Library v. 3.01 of Walter Zorn is used.
- For handling MATLAB related parts, MATLAB Version 7.1.0.246 is used.
- For compiling C/C++ files of XM Software, Microsoft Visual Studio 6.0 is used.

5.2 Dataset

For the experiments, CalTech 101 image dataset [17] is used. The dataset gives both images and the objects marked in the images, also all of the images are grouped under categories they belong to. It contains pictures of objects belonging to 101 categories, about 30 to 800 images per category and most of them having about 50 images. A total list of categories and number of images in the categories are given in Table 5.1. The size of each image is roughly

300 x 200 pixels. The annotations (marked objects) are given as MATLAB matrices. To use the object annotations from Java implementation, a small program is written in Java with a MATLAB-Java conversion library (JMatIO [19]).

Not all of the images in the dataset are used in the experiments. Since it is important to use approximate number of images for each category during the training phases and the distribution of images in the dataset is not so smooth, the number of images used for each category is bounded. Also the dataset is divided into three to form First-Training, Second-Training and Test datasets. Bound of number of images for each one is determined as 30.

It is not necessary to use all categories during tests for examining the success of the system. The successes of each category and also the average is examined during evaluation so the results obtained by using a part of all categories can give us approximate results as all categories. With this consideration, 10 categories are used for tests.

According to the declaration in Chapter 4, initialization (First-Training) of the system is more important than the improvement part (Second-Training). For a more successful system, the number of categories kept should be as much as possible. So during First-Traning all 101 categories are used and introduced to the system. But improvement is done only for test purposes, so only the categories selected for tests are used in Second-Training phase.

Furthermore, the same number of images are not used from the selected categories. Table 5.2 gives the number of images used for selected categories. Using different number of images for different categories provides us to compare the results according to different number of images encountered. The details about how many images are used from each category for each phase is given in Table 5.2.

In addition to the images used, a few video files is used during the tests. The video files are taken from the Open Video Project [30].

5.3 Experiments and Results

The experimental part of the system is arranged under two parts. In the first part, the performance of the proposed model is tried to be measured. In the second part, the whole system is tried to be used for a video and total process time of the system is calculated. Details are given in the corresponding subsections after the subsection about training.

Table 5.1: CalTech 101 Image Dataset, All Images

No	Category	Size	No	Category	Size	No	Category	Size
1	accordion	55	36	ewer	85	71	panda	38
2	airplanes	800	37	Faces	435	72	pigeon	45
3	anchor	42	38	Faces_easy	435	73	pizza	53
4	ant	42	39	ferry	67	74	platypus	34
5	barrel	47	40	flamingo	67	75	pyramid	57
6	bass	54	41	flamingo_head	45	76	revolver	82
7	beaver	46	42	garfield	34	77	rhino	59
8	binocular	33	43	gerenuk	34	78	rooster	49
9	bonsai	128	44	gramophone	51	79	saxophone	40
10	brain	98	45	grand_piano	99	80	schooner	63
11	brontosaurus	43	46	hawksbill	100	81	scissors	39
12	buddha	85	47	headphone	42	82	scorpion	84
13	butterfly	91	48	hedgehog	54	83	sea_horse	57
14	camera	50	49	helicopter	88	84	snoopy	35
15	cannon	43	50	ibis	80	85	soccer_ball	64
16	car_side	123	51	inline_skate	31	86	stapler	45
17	ceiling_fan	47	52	joshua_tree	64	87	starfish	86
18	cellphone	59	53	kangaroo	86	88	stegosaurus	59
19	chair	62	54	ketch	114	89	stop_sign	64
20	chandelier	107	55	lamp	61	90	strawberry	35
21	cougar_body	47	56	laptop	81	91	sunflower	85
22	cougar_face	69	57	Leopards	200	92	tick	49
23	crab	73	58	llama	78	93	trilobite	86
24	crayfish	70	59	lobster	41	94	umbrella	75
25	crocodile	50	60	lotus	66	95	watch	239
26	crocodile_head	51	61	mandolin	43	96	water_lilly	37
27	cup	57	62	mayfly	40	97	wheelchair	59
28	dalmatian	67	63	menorah	87	98	wild_cat	34
29	dollar_bill	52	64	metronome	32	99	windsor_chair	56
30	dolphin	65	65	minaret	76	100	wrench	38
31	dragonfly	68	66	Motorbikes	798	101	yin_yang	60
32	electric_guitar	75	67	nautilus	55		**TOTAL**	**8676**
33	elephant	64	68	octopus	35		**MIN**	**31**
34	emu	53	69	okapi	39		**MAX**	**800**
35	euphonium	64	70	pagoda	47		**AVG**	**86**

Table 5.2: CalTech 101 Image Dataset, Used Images for Tests

No	Category	1st Tr.	2nd Tr.	Test	No	Category	1st Tr.	2nd Tr.	Test	No	Category	1st Tr.	2nd Tr.	Test
1	accordion	18	-	-	36	ewer	28	-	-	71	panda	12	-	-
2	airplanes	30	-	-	37	Faces	30	-	-	72	pigeon	15	-	-
3	anchor	14	-	-	38	Faces_easy	30	-	-	73	pizza	17	-	-
4	ant	14	-	-	39	ferry	22	-	-	74	platypus	11	-	-
5	barrel	15	-	-	40	flamingo	22	-	-	75	pyramid	19	-	-
6	bass	18	-	-	41	flamingo_head	15	-	-	76	revolver	27	-	-
7	beaver	15	-	-	42	garfield	11	-	-	77	rhino	19	-	-
8	binocular	11	-	-	43	gerenuk	11	-	-	78	rooster	16	-	-
9	bonsai	30	-	-	44	gramophone	17	-	-	79	saxophone	13	-	-
10	brain	30	-	-	45	grand_piano	30	30	30	80	schooner	21	-	-
11	brontosaurus	14	-	-	46	hawksbill	30	-	-	81	scissors	13	-	-
12	buddha	28	-	-	47	headphone	14	-	-	82	scorpion	28	-	-
13	butterfly	30	-	-	48	hedgehog	18	-	-	83	sea_horse	19	-	-
14	camera	16	-	-	49	helicopter	29	29	29	84	snoopy	11	-	-
15	cannon	14	-	-	50	ibis	26	-	-	85	soccer_ball	21	-	-
16	car_side	30	30	30	51	inline_skate	10	10	10	86	stapler	15	-	-
17	ceiling_fan	15	-	-	52	joshua_tree	21	-	-	87	starfish	28	-	-
18	cellphone	19	-	-	53	kangaroo	28	-	-	88	stegosaurus	19	-	-
19	chair	20	-	-	54	ketch	30	-	-	89	stop_sign	21	-	-
20	chandelier	30	-	-	55	lamp	20	-	-	90	strawberry	11	-	-
21	cougar_body	15	-	-	56	laptop	27	27	27	91	sunflower	28	-	-
22	cougar_face	23	-	-	57	Leopards	30	-	-	92	tick	16	-	-
23	crab	24	-	-	58	llama	26	-	-	93	trilobite	28	-	-
24	crayfish	23	-	-	59	lobster	13	-	-	94	umbrella	25	-	-
25	crocodile	16	-	-	60	lotus	22	-	-	95	watch	30	-	-
26	crocodile_head	17	-	-	61	mandolin	14	-	-	96	water_lilly	12	-	-
27	cup	19	-	-	62	mayfly	13	-	-	97	wheelchair	19	-	-
28	dalmatian	22	-	-	63	menorah	29	-	-	98	wild_cat	11	-	-
29	dollar_bill	17	17	17	64	metronome	10	10	10	99	windsor_chair	18	-	-
30	dolphin	21	-	-	65	minaret	25	25	25	100	wrench	12	-	-
31	dragonfly	22	-	-	66	Motorbikes	30	30	30	101	yin_yang	20	-	-
32	electric_guitar	25	-	-	67	nautilus	18	-	-		TOTAL	2027	229	229
33	elephant	21	-	-	68	octopus	11	-	-		MIN	10	10	10
34	emu	17	-	-	69	okapi	13	-	-		MAX	30	30	30
35	euphonium	21	21	21	70	pagoda	15	-	-		AVG	20	23	23

5.3.1 Training of The System

As mentioned before, the system is trained through two training phases; for initialization and for improvement of the system. In both training phases, the CalTech 101 dataset images are used with number of images given in Table 5.2. In the First-Training, firstly objects are extracted from dataset images and stored as separate image files. Some samples of dataset images and extracted object are shown in Figure 5.14. Then feature extraction of all images are performed.

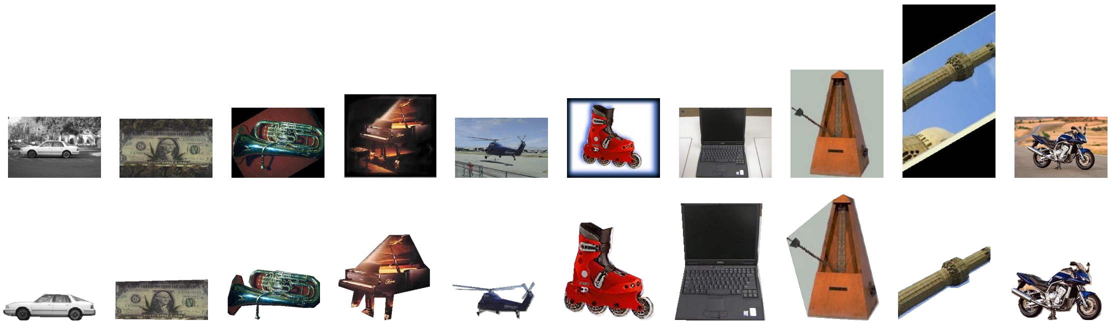


Figure 5.14: Samples from Caltech 101 Dataset Images and Object Annotations

After feature extraction, the normalization process is performed. The analysis work done in Sub-section 4.6.1 (fig-normalization) clearly shows the effect of normalization on the First-Training data decisions. During First-Training, lastly feature importance determination is carried out. The results obtained from feature importance determination are given in Table 5.3.

For Second-Training, images from the selected categories are given to the system. The images are given in four steps, in each step 1/4 of the images are given in order to see the effect of GA with increasing number of images.. In each step the success of the system is examined (given in Sub-section 5.3.2).

During genetic operations, there are some parameters that can affect the results of the system. These are the effectiveness correction factors κ_{own} and κ_{oth}, the mutation factor mf and fitness function correct acceptance choice.

Some random or experimental constants are not preferred for effectiveness correction factors. They are calculated according to the number of training images used in First-

Table 5.3: Feature Importance Values for Selected Categories. CL: ColorLayout, CSt: ColorStructure, DC: DominantColor, SC: ScalableColor, CSh: ContourShape, RS: RegionShape, EH: EdgeHistogram, HT: HomogeneousTexture

	Color Features					Shape Features			Texture Features		
Category	CL	CSt	DC	SC	**Total**	CSh	RS	**Total**	EH	HT	**Total**
car_side	0.0414	0.0664	0.0375	0.2064	**0.3518**	0.3663	0.0085	**0.3748**	0.2598	0.0136	**0.2734**
dollar_bill	0.0730	0.0021	0.0220	0.0688	**0.1659**	0.3638	0.0089	**0.3727**	0.4508	0.0107	**0.4614**
euphonium	0.0120	0.0060	0.0486	0.0154	**0.0820**	0.4302	0.0098	**0.4400**	0.4485	0.0295	**0.4780**
grand_piano	0.0819	0.0416	0.0250	0.0530	**0.2015**	0.4713	0.0690	**0.5402**	0.2528	0.0055	**0.2583**
helicopter	0.0547	0.0298	0.0679	0.0280	**0.1804**	0.3096	0.0439	**0.3534**	0.4594	0.0068	**0.4662**
inline_skate	0.0439	0.0283	0.0871	0.1359	**0.2951**	0.4080	0.0436	**0.4516**	0.2288	0.0245	**0.2533**
laptop	0.0493	0.0153	0.0259	0.0475	**0.1380**	0.7150	0.0185	**0.7335**	0.1262	0.0023	**0.1285**
metronome	0.0320	0.0478	0.0651	0.0565	**0.2013**	0.3885	0.0449	**0.4333**	0.3576	0.0078	**0.3654**
minaret	0.0042	0.0091	0.0023	0.0145	**0.0302**	0.6140	0.0136	**0.6276**	0.3404	0.0018	**0.3422**
Motorbikes	0.0524	0.1099	0.1485	0.1445	**0.4553**	0.2215	0.0514	**0.2729**	0.2491	0.0227	**0.2718**

Training. For a category C, the values are calculated as;

$$\kappa_{own} = \frac{1}{count\ of\ images\ in\ C} \tag{5.1}$$

$$\kappa_{oth} = \frac{1}{count\ of\ All\ images - count\ of\ images\ in\ C} \tag{5.2}$$

Mutation factor is chosen as $mf = 0.5$ in order to import half of the genes in each mutation. This constant is chosen intuitively with the aim not to make Second-Training images more dominant than the First-Training images.

For the fitness function correct acceptance, the ranking method is preferred in order to accept the chromosomes which are better than the average and reach a state all chromosomes having same or close fitness values.

Execution times of above given operations are shown in Table 5.4.

Table 5.4: Execution Times for Training Operations

Operation	Execution Time(s)
Object Annotation Extraction	150
Feature Extraction	8,500
Feature Value Normalization	400
Feature Importance Determination	50,000
Genetic Operations	6,000
Total	65,050

5.3.2 Image Trial and Performance Measurement

The main contribution of this study is the GA based object classifier. So it is more important to see the performance of the classfier rather than the whole system.

The whole system can be thought as combination of three components: video to image extractor, image to segment extractor, segment to object classifier. First two components are directly embedded into the proposed system, so measuring the success of the methods used in these systems is unnecessary. It is better to refer to the evaluations in the original studies. [33] [56] [10]

Thus, the tests are performed with the candidate objects in the images. Since we use CalTech 101 dataset, the candidate objects are extracted from images by using the annotations that the dataset provides.

Tests are performed in five steps: after First-Training images are encountered, 1/4, 2/4, 3/4 and all of Second-Training images are encountered. In the tables and graphs, they are represented as S_0, S_1, S_2, S_3, S_4, respectively (Table 5.5). The number of images that are used in the First-Training and Second-Training phases are given in Table 5.2. The change of data in given order shows the improvement achieved by using GA and genetic operations. The success of S_0 shows the success of base model without using genetic operations.

Table 5.5: Testing Times or Test Steps, given in occurence order

Step Name	Description
S_0	The time after First-Training
S_1	The time after improving the system by performing genetic operations with 1/4 of Second-Training images
S_2	The time after improving the system by performing genetic operations with 2/4 of Second-Training images
S_3	The time after improving the system by performing genetic operations with 3/4 of Second-Training images
S_4	The time after improving the system by performing genetic operations with 4/4 of Second-Training images (after finishing Second-Training)

In each step, performance on First-Training images, Second-Training images and Test images are measured. As given in Table 5.2, each of these three tests is performed with 229 images from 10 categories. So, totally 15 tests are performed.

The test results and evaluations are given below.

Decisions of Categories and How To Use Them

As the output of the system, the classifier gives decisions of 101 categories for each query object(image). A sample output for a single query object (Figure 5.15) is given in Table 5.6.

Figure 5.15: A Sample Query Object, An image with category 'car_side'

Table 5.6: Decisions on A Sample Query Object, An image with category 'car_side'

Category	Dec.	Category	Dec.	Category	Dec.	Category	Dec.	Category	Dec.
car_side	0.90	stapler	0.62	flamingo	0.54	dalmatian	0.43	laptop	0.26
Leopards	0.76	crocodile_head	0.60	headphone	0.54	butterfly	0.42	starfish	0.26
euphonium	0.72	pigeon	0.60	kangaroo	0.54	ketch	0.42	strawberry	0.23
ferry	0.72	sea_horse	0.60	lobster	0.54	pyramid	0.42	Faces	0.22
airplanes	0.71	dollar_bill	0.59	crab	0.53	elephant	0.40	rooster	0.21
mandolin	0.69	platypus	0.59	ant	0.52	joshua_tree	0.40	accordion	0.18
crocodile	0.68	saxophone	0.59	chair	0.52	schooner	0.40	Faces_easy	0.17
helicopter	0.68	hawksbill	0.58	dragonfly	0.52	inline_skate	0.39	water_lilly	0.17
minaret	0.68	lamp	0.58	emu	0.52	stegosaurus	0.38	cougar_face	0.15
Motorbikes	0.68	wild_cat	0.58	ceiling_fan	0.51	buddha	0.37	barrel	0.14
bass	0.67	cannon	0.57	okapi	0.51	hedgehog	0.37	brain	0.14
cellphone	0.67	metronome	0.57	anchor	0.50	binocular	0.36	stop_sign	0.14
dolphin	0.65	scissors	0.57	ewer	0.50	grand_piano	0.36	yin_yang	0.14
electric_guitar	0.65	watch	0.57	flamingo_head	0.50	menorah	0.35	cup	0.11
garfield	0.65	ibis	0.56	llama	0.50	camera	0.34	tick	0.11
wrench	0.65	rhino	0.56	scorpion	0.50	chandelier	0.34	soccer_ball	0.07
cougar_body	0.64	windsor_chair	0.56	bonsai	0.48	pizza	0.33	nautilus	0.01
gerenuk	0.63	brontosaurus	0.55	umbrella	0.47	sunflower	0.33		
gramophone	0.63	crayfish	0.55	octopus	0.46	wheelchair	0.33		
revolver	0.63	mayfly	0.55	pagoda	0.45	lotus	0.31		
snoopy	0.62	beaver	0.54	panda	0.44	trilobite	0.31		

For each of the performed 15 tests, 229 decision results like the one given in Table 5.6. To interpret these raw results, in other words to understand what the classifier want to say with these results, the result data should be thresholded. Two methods are used to interpret the results:

- Thresholding according to the decision values: The classifier is accepted as it classifies the object to the categories that gives decision values bigger than the threshold value (The system supports multiple categorization). From now on, 'thresholding according to the decision values' is used as 'thresholding' or '$Thr(n)$' shortly.

- Thresholding according to the rank of decision: First the categories are sorted acconding to their decision values. The classifier is accepted as it classifies the object to the categories that have a ranking better than the rank threshold. From now on, 'thresholding according to the rank of decision' is used as 'ranking' or '$R(n)$' shortly.

Average Decisions of Categories On Own Category Images

Table 5.7 is acquired by taking the average of decision values given to the images for each category. The 'Average' line in the table shows the average of decisions of all images given by their real category. Figure 5.16 displays the change of average decision in time (test steps). Expected result is an increase in the decisions of the categories. According to Figure 5.16 and Table 5.7, the average decisions on all datasets increases. Considering total change, all categories also increases their decisions. It is acceptable to have some decreases between particular steps, since they can be easily affected from the random image selection. If some 'bad' images are retrieved during any step of Second-Training, this situation decreases the decision. It is not possible to see such case for the test results on Second-Training images because the implementation guarantees to increase the decisions Second-Training images (during genetic operations, as declared in Sub-section 4.7.5).

Average Normalized Modified Retrieval Rank ($ANMRR$) of Categories

Average Normalized Modified Retrieval Rank ($ANMRR$) is a performance measure metric for retrieval that is defined by the MPEG-7 research group. The purpose of the metric is to allow an evaluation of different descriptors that is unbiased with respect to different sample and ground truth sizes, and correlates well with perceptual judgment about the retrieval success rate [44]. Scores are calculated according to the rank of the results and not their value. Lower $ANMRR$ means better performance. The calculation is as follows:

$$ANMRR = \frac{1}{Q}\sum_{q=1}^{Q} NMRR(q) \tag{5.3}$$

$$NMRR(q) = \frac{AVR(q) - 0.5 \cdot [1 + NG(q)]}{1.25 \cdot K(q) - 0.5 \cdot [1 + NG(q)]} \tag{5.4}$$

Table 5.7: Average Decisions of Categories On Own Category Images

		S_0	S_1	S_2	S_3	S_4
on First-Training Images	car_side	0.8893	0.8997	0.8993	0.9007	0.9003
	dollar_bill	0.8147	0.8418	0.8459	0.8541	0.8412
	euphonium	0.8062	0.8290	0.8386	0.8424	0.8443
	grand_piano	0.7743	0.7927	0.7970	0.8057	0.8053
	helicopter	0.7514	0.7548	0.7600	0.7714	0.7755
	inline_skate	0.7730	0.7940	0.8010	0.8130	0.8140
	laptop	0.8044	0.8093	0.8196	0.8304	0.8281
	metronome	0.8060	0.8310	0.8330	0.8340	0.8390
	minaret	0.7908	0.8164	0.8200	0.8320	0.8640
	Motorbikes	0.8397	0.8557	0.8613	0.8640	0.8597
	Average	0.8076	0.8234	0.8285	0.8357	0.8383
		S_0	S_1	S_2	S_3	S_4
on Second-Training Images	car_side	0.8920	0.8920	0.8953	0.8967	0.9000
	dollar_bill	0.7829	0.8076	0.8141	0.8224	0.8318
	euphonium	0.7733	0.8005	0.8133	0.8195	0.8205
	grand_piano	0.7493	0.7657	0.7697	0.7787	0.7763
	helicopter	0.7417	0.7538	0.7600	0.7731	0.7776
	inline_skate	0.7660	0.7980	0.8140	0.8470	0.8510
	laptop	0.8011	0.8089	0.8196	0.8359	0.8356
	metronome	0.7580	0.7780	0.7850	0.7890	0.7960
	minaret	0.7820	0.8124	0.8176	0.8308	0.8528
	Motorbikes	0.7870	0.7987	0.8067	0.8087	0.8103
	Average	0.7875	0.8035	0.8108	0.8202	0.8248
		S_0	S_1	S_2	S_3	S_4
on Test Images	car_side	0.8967	0.8927	0.8923	0.8953	0.8930
	dollar_bill	0.8335	0.8688	0.8694	0.8776	0.8682
	euphonium	0.7671	0.7900	0.7971	0.8024	0.8024
	grand_piano	0.7503	0.7690	0.7713	0.7753	0.7747
	helicopter	0.7345	0.7455	0.7507	0.7655	0.7631
	inline_skate	0.7460	0.7660	0.7750	0.7880	0.7910
	laptop	0.8030	0.8107	0.8167	0.8322	0.8330
	metronome	0.7610	0.7850	0.7870	0.7900	0.7910
	minaret	0.8056	0.8304	0.8352	0.8452	0.8604
	Motorbikes	0.7393	0.7483	0.7527	0.7560	0.7523
	Average	0.7863	0.8010	0.8049	0.8129	0.8129

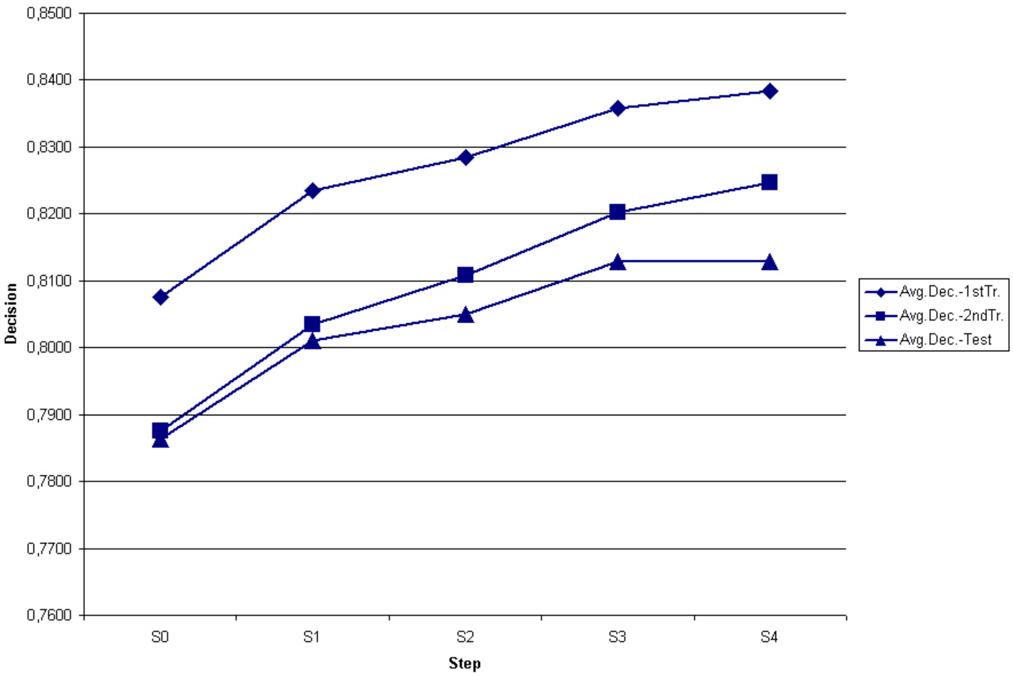


Figure 5.16: Average Decisions of Categories On Own Category Images

$$AVR(q) = \sum_{k=1}^{NG(q)} \frac{Rank^*(k)}{NG(q)} \tag{5.5}$$

$$Rank^*(k) = \begin{cases} Rank(k) & , Rank(k) \leq K(q) \\ 1.25 \cdot K(q) & , Rank(k) > K(q) \end{cases} \tag{5.6}$$

where $NG(q)$ is the ground truth size of query q and $K(q)$ is the 'relevant ranks' (the ranks that would still count as feasible in terms of subjective evaluation of retrieval). For relatively large $NG(q)$ (about 20-25 items), $K(q)$ is used about twice the $NG(q)$ size, while for smaller $NG(q)$ more tolerance is allowed (e.g. four times of $NG(q)$ size). [7]

As an example, the following can be given: Suppose that a query q has 10 similar images in the database ($NG(q) = 10$, $K = 20$). If the result contains 5 true retrieval with ranks 1,2,5,9,11,12,22,25,30,40 then;

$$\begin{aligned} AVR(q) &= \sum_{k=1}^{10} \frac{Rank^*(k)}{10} \\ &= \frac{1+2+5+9+11+12+25+25+25+25}{10} = 14 \end{aligned} \tag{5.7}$$

$$NMRR(q) = \frac{14-5.5}{25-5.5} = 0.4359 \tag{5.8}$$

If only one query is performed, then $ANMRR(q) = 0.4359$.

To calculate $ANMRR$ for the proposed system, $NMRR$ is calculated for each query and then $ANMRR$ is found by averaging. For $NMRR$, the following number are used: $NG(q) = 1$ since the classifier are working for only 1 image, $K(q) = 4 \cdot NG(q) = 4$ since $NG(q)$ is small.

The calculated $ANMRR$ table is given in Table 5.8. 'Total' line represents the average of all queries performed. Also the figure given in Figure 5.17 displays the change of 'Total' $ANMRR$ during test steps. The expectation from the test results is to see a decrease in the $ANMRR$ values. Considering the change between S_0 and S_4, Figure 5.17 and Table 5.8, all $ANMRR$ values decreases. As mentioned before, it acceptable to have some unexpected change during any of the steps. An increase in any of the steps can be caused by the selection of 'bad' random images in the corresponding step, during Second-Training.

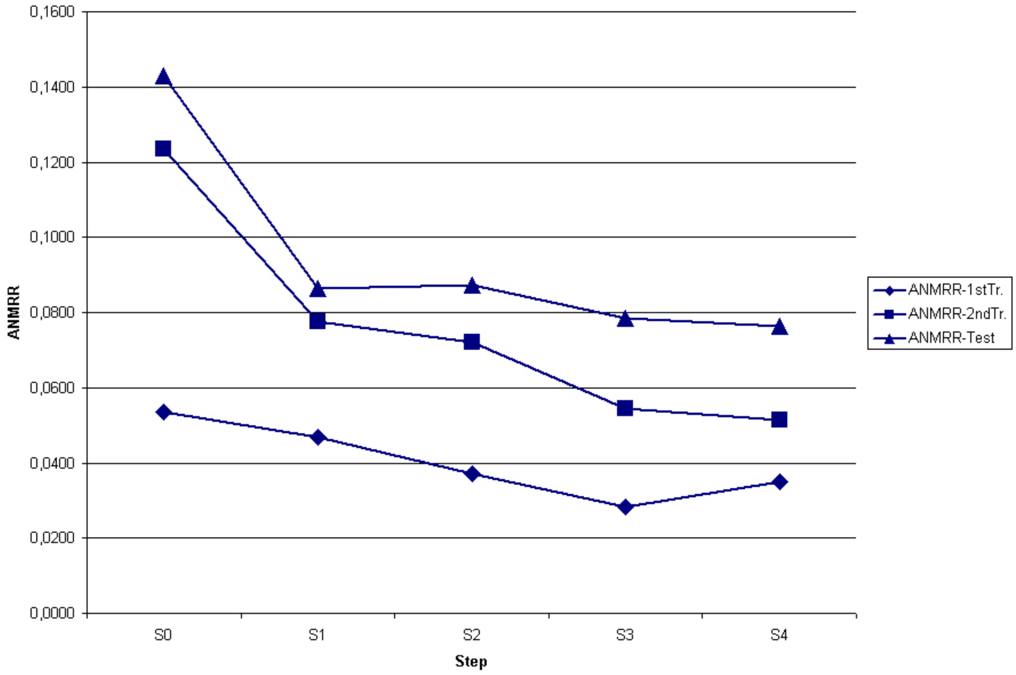


Figure 5.17: Average Normalized Modified Retrieval Rank (ANMRR) of Categories

Precision & Recall of Categories with Thresholding

Precision and recall are the other important metrics to see the performance of the retrieval systems. They are calculated according to the formulas below:

$$Precision = \frac{number\ of\ Relevant\ Retrived}{number\ of\ All\ Retrieved} \tag{5.9}$$

Table 5.8: Average Normalized Modified Retrieval Rank (ANMRR) of Categories

		S_0	S_1	S_2	S_3	S_4
on First-Training Images	car_side	0.0000	0.0000	0.0000	0.0000	0.0000
	dollar_bill	0.2059	0.1176	0.1618	0.1176	0.1618
	euphonium	0.0833	0.0714	0.0714	0.0714	0.0595
	grand_piano	0.0417	0.0583	0.0417	0.0083	0.0333
	helicopter	0.0690	0.0690	0.0603	0.0431	0.0431
	inline_skate	0.0000	0.1000	0.0250	0.0000	0.0000
	laptop	0.0185	0.0278	0.0000	0.0093	0.0185
	metronome	0.0500	0.0500	0.0500	0.0500	0.0500
	minaret	0.1100	0.0300	0.0200	0.0200	0.0100
	Motorbikes	0.0000	0.0167	0.0000	0.0083	0.0167
	Total	0.0535	0.0469	0.0371	0.0284	0.0349
on Second-Training Images		S_0	S_1	S_2	S_3	S_4
	car_side	0.0000	0.0000	0.0000	0.0000	0.0000
	dollar_bill	0.2647	0.2059	0.1618	0.1618	0.1029
	euphonium	0.2500	0.1667	0.1429	0.1429	0.1310
	grand_piano	0.1833	0.1167	0.1417	0.0917	0.1000
	helicopter	0.0776	0.0172	0.0086	0.0086	0.0086
	inline_skate	0.0250	0.0000	0.0000	0.0000	0.0000
	laptop	0.1204	0.0741	0.0926	0.0370	0.0463
	metronome	0.3000	0.2500	0.2000	0.1500	0.1500
	minaret	0.1000	0.0300	0.0300	0.0100	0.0100
	Motorbikes	0.0583	0.0500	0.0333	0.0333	0.0333
	Total	0.1234	0.0775	0.0721	0.0546	0.0513
on Test Images		S_0	S_1	S_2	S_3	S_4
	car_side	0.0000	0.0000	0.0000	0.0000	0.0000
	dollar_bill	0.3529	0.1324	0.1618	0.1471	0.1765
	euphonium	0.2976	0.2857	0.2976	0.2857	0.2857
	grand_piano	0.1917	0.1250	0.1250	0.1250	0.1333
	helicopter	0.1466	0.1121	0.0948	0.0776	0.0776
	inline_skate	0.1750	0.0750	0.0750	0.0500	0.0250
	laptop	0.0741	0.0556	0.0648	0.0093	0.0000
	metronome	0.1750	0.0500	0.0500	0.1000	0.0250
	minaret	0.0200	0.0100	0.0100	0.0100	0.0000
	Motorbikes	0.1500	0.0500	0.0417	0.0500	0.0583
	Total	0.1430	0.0862	0.0873	0.0786	0.0764

$$Recall \quad = \quad \frac{number\ of\ Relevant\ Retrived}{number\ of\ All\ Relevants} \tag{5.10}$$

As understood from the formulas; precision shows what ratio of the retrievals of the system is correct, besides recall shows what ratio of ground truth is retrieved.

Using the results obtained during the test, different precision and recall values can be calculated by using different 'thresholding' values. The curves acquired by applying different 'thresholding' values are given in Figure 5.18, Figure 5.19 and Figure 5.20. In the figures, curves of each category and also the total system is drawn separately.

An average best point in a precision-recall curve can be accepted as the point with both two values are closer to 1. This can be easily figured out by finding the point where product of precision and recall is maximum among all set. In Table 5.9, the best precision and recall values according to this assumption are given. In the table best precision and recall values of each category and total system is calculated separately. The values of total system does not mean the average of all categories, but the formulas of precision and recall are used for considering all queries. (Assume that 250 queries are performed. By using a 'thresholding' of 0.90, 200 of them can be assigned to some categories and 150 of them are true. Then precision is 0.75 and recall is 0.6.)

In Figure 5.21, the change in precision and recall values of total system is represented according to the test steps.

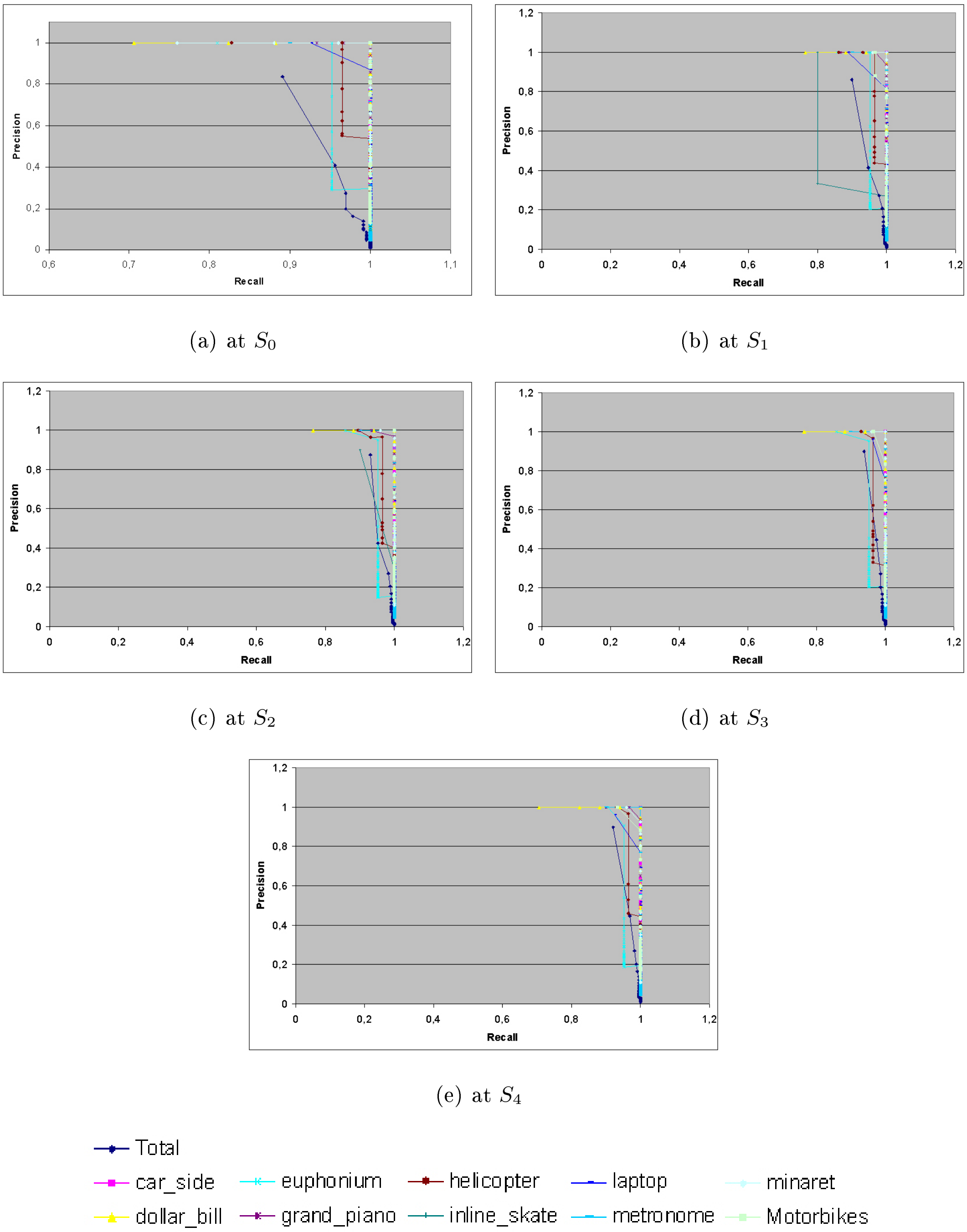
Precision & Recall of Categories for Best Total Threshold

It is hard to find a best thresholding point that makes all the categories and total system achieve the best results. For comparisons, the value that makes the whole system to achieve its best results, but not each category, can be used as the thresholding point.

In the previous part, the best precision and recall values of each category are given separately. Here, precision and recall values of each category is given with a 'thresholding' value that is the 'thresholding' value of total system at its best point (Table 5.10).

Precision & Recall of Categories with Ranking

As mentioned before, it is hard to find the best thresholding point that makes the system to achieve its best results. Instead, a ranking mechanism can be used. In previous two parts, precision and recall values calculated by using 'thresholding', here 'ranking' is used to obtain the precision-recall curve. The curves of each category and total system is given in Figure 5.22, Figure 5.23 and Figure 5.24. In Table 5.11, best precision and recall values of



(a) at S_0

(b) at S_1

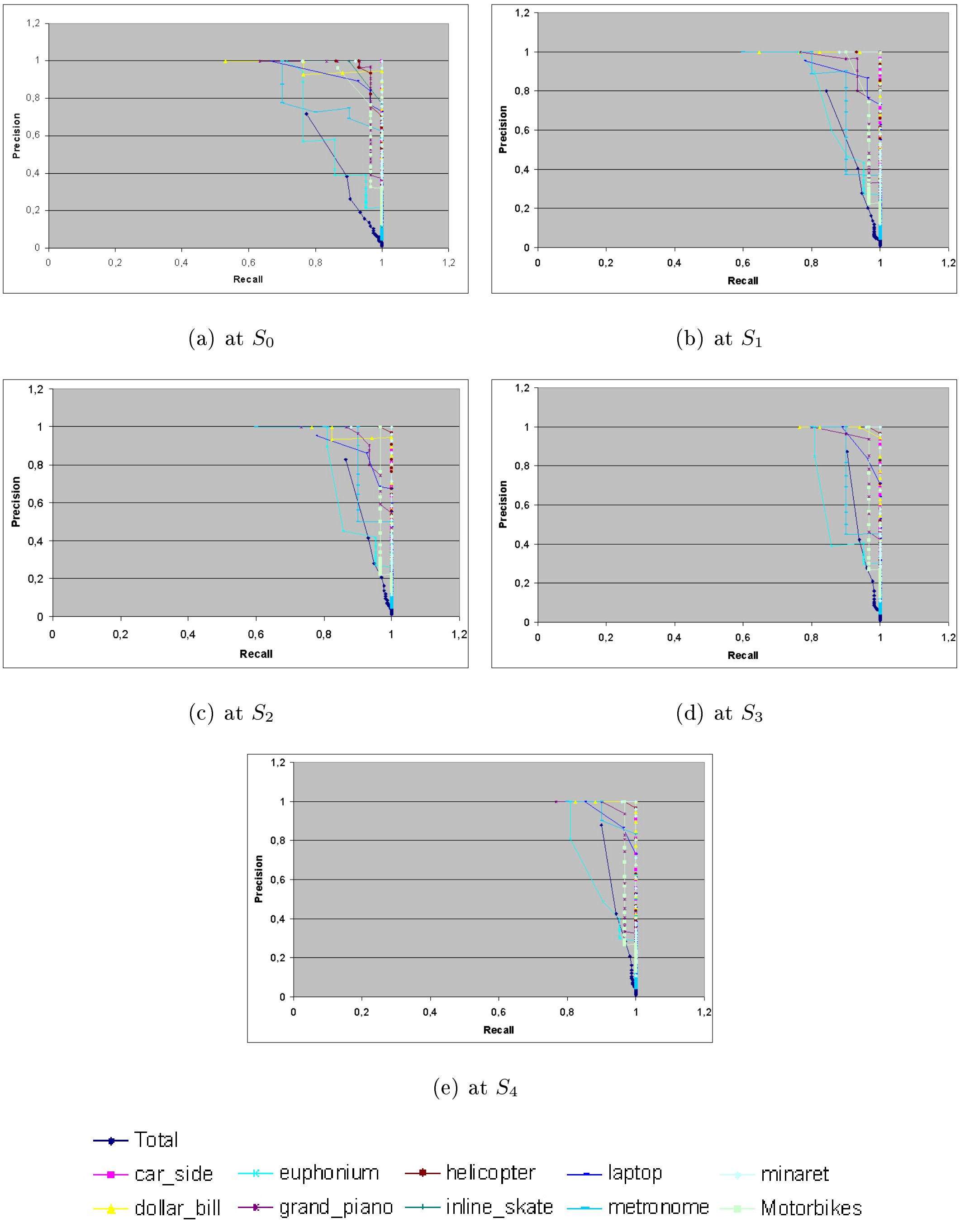
(c) at S_2

(d) at S_3

(e) at S_4

(f) Legend

Figure 5.18: Precision vs Recall Curve for All Categories with Thresholding, on First-Training Images



(a) at S_0

(b) at S_1

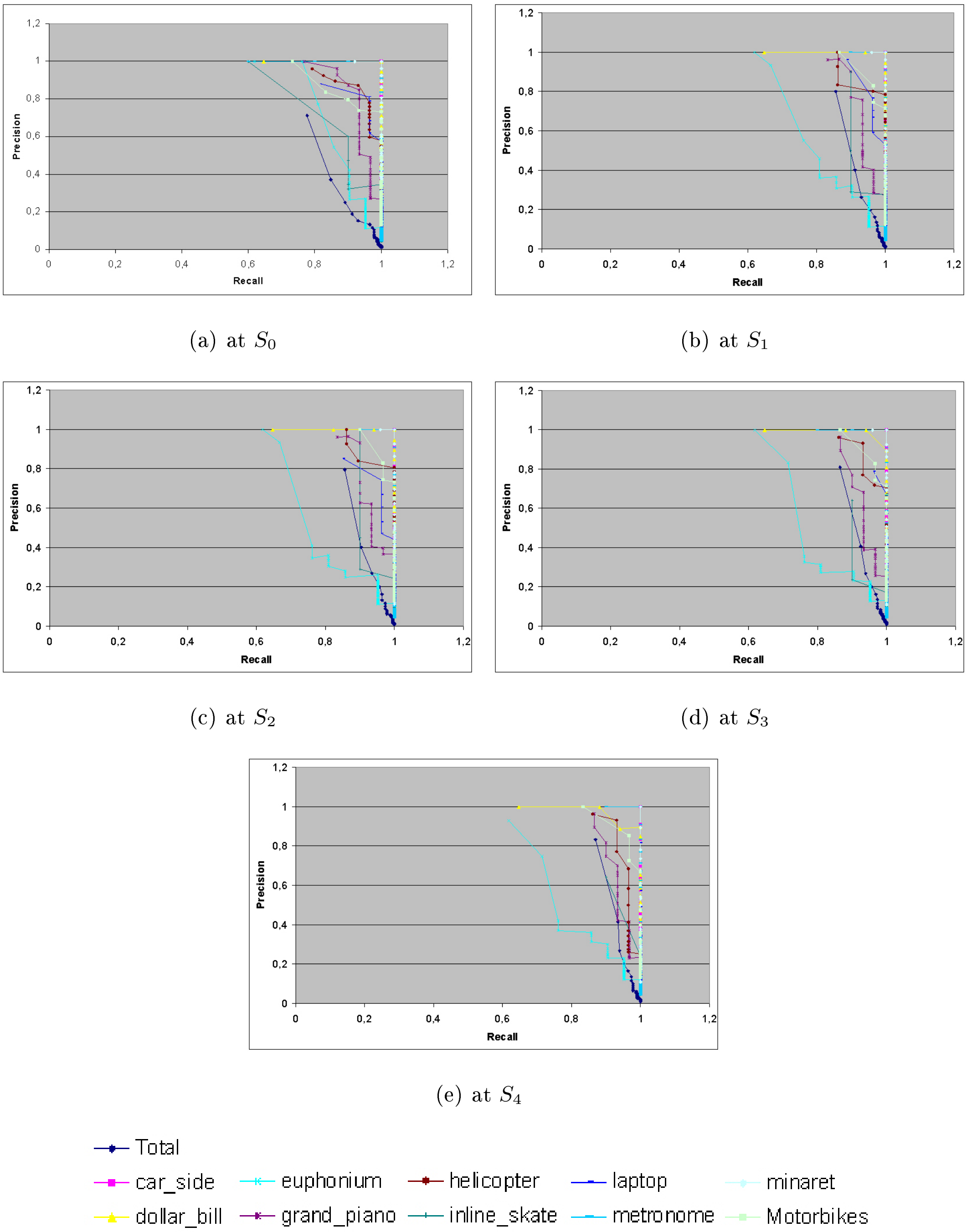
(c) at S_2

(d) at S_3

(e) at S_4

(f) Legend

Figure 5.19: Precision vs Recall Curve for All Categories with Thresholding, on Second-Training Images



(a) at S_0

(b) at S_1

(c) at S_2

(d) at S_3

(e) at S_4

(f) Legend

Figure 5.20: Precision vs Recall Curve for All Categories with Thresholding, on Test Images

Table 5.9: Precision & Recall of Categories with Thresholding

		S1			S2			S3			S4			S5		
		Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr
on First-Training Images	car_side	1.0000	1.0000	0.8600	1.0000	1.0000	0.8600	1.0000	1.0000	0.8700	1.0000	1.0000	0.8700	1.0000	1.0000	0.8700
	dollar_bill	1.0000	1.0000	0.7500	1.0000	1.0000	0.7300	1.0000	1.0000	0.7300	1.0000	1.0000	0.7500	1.0000	1.0000	0.7500
	euphonium	1.0000	0.9524	0.7600	0.9524	0.9524	0.7400	1.0000	0.8571	0.7800	1.0000	0.8571	0.7800	0.9524	0.9524	0.7400
	grand_piano	0.9355	0.9667	0.6900	1.0000	0.9000	0.7300	1.0000	0.9000	0.7400	1.0000	0.9000	0.7500	1.0000	0.9000	0.7400
	helicopter	1.0000	0.9310	0.7200	0.9032	0.9655	0.7000	1.0000	0.8621	0.7300	1.0000	0.8966	0.7300	1.0000	0.8966	0.7400
	inline_skate	1.0000	0.9000	0.7500	1.0000	0.7000	0.7600	1.0000	0.8000	0.7600	0.6250	1.0000	0.7700	0.6250	1.0000	0.7700
	laptop	0.9630	0.9630	0.7600	0.9630	0.9630	0.7600	0.9643	1.0000	0.7700	1.0000	1.0000	0.7900	1.0000	0.9630	0.7800
	metronome	1.0000	1.0000	0.7300	0.9091	1.0000	0.7100	1.0000	0.9000	0.7600	1.0000	0.9000	0.7600	0.9091	1.0000	0.7200
	minaret	1.0000	1.0000	0.7000	1.0000	1.0000	0.7000	1.0000	1.0000	0.7100	1.0000	1.0000	0.7300	1.0000	1.0000	0.8000
	Motorbikes	0.9667	0.9667	0.7300	0.9667	0.9667	0.7200	0.9667	0.9667	0.7200	0.9667	0.9667	0.7200	0.9667	0.9667	0.7200
	Total	0.6920	0.6769	0.7900	0.7843	0.6987	0.8000	0.8280	0.6725	0.8100	0.8047	0.7555	0.8000	0.8224	0.7686	0.8000
		S1			S2			S3			S4			S5		
		Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr
on Second-Training Images	car_side	1.0000	1.0000	0.8300	1.0000	1.0000	0.8300	1.0000	1.0000	0.8400	1.0000	1.0000	0.8400	1.0000	1.0000	0.8500
	dollar_bill	1.0000	0.9412	0.7200	1.0000	0.9412	0.7400	1.0000	0.9412	0.7600	1.0000	0.9412	0.7800	1.0000	0.9412	0.7800
	euphonium	1.0000	0.8095	0.7200	1.0000	0.8095	0.7600	1.0000	0.8095	0.7600	1.0000	0.8095	0.7800	1.0000	0.8095	0.7700
	grand_piano	0.9643	0.9000	0.7000	1.0000	0.8667	0.7200	1.0000	0.8333	0.7200	1.0000	0.8667	0.7300	1.0000	0.8667	0.7300
	helicopter	0.9615	0.8621	0.7100	1.0000	0.9655	0.7200	1.0000	0.9655	0.7300	0.9667	1.0000	0.7200	0.9667	1.0000	0.7200
	inline_skate	0.9091	1.0000	0.7400	1.0000	1.0000	0.7700	1.0000	1.0000	0.7900	1.0000	1.0000	0.8100	1.0000	1.0000	0.8100
	laptop	0.9643	1.0000	0.7600	0.9643	1.0000	0.7700	0.9310	1.0000	0.7600	0.9643	1.0000	0.7800	1.0000	1.0000	0.7800
	metronome	0.9000	0.9000	0.7200	0.9000	0.9000	0.7300	1.0000	0.9000	0.7400	1.0000	0.9000	0.7400	1.0000	0.9000	0.7600
	minaret	1.0000	1.0000	0.7000	1.0000	1.0000	0.7200	1.0000	1.0000	0.7300	1.0000	1.0000	0.7600	1.0000	1.0000	0.8000
	Motorbikes	0.8710	0.9000	0.7400	0.8710	0.9000	0.7500	0.9000	0.9000	0.7600	0.8710	0.9000	0.7600	0.8710	0.9000	0.7600
	Total	0.5936	0.5677	0.7800	0.7277	0.6419	0.7900	0.7406	0.6856	0.7900	0.7578	0.7380	0.7900	0.7619	0.7686	0.7900
		S1			S2			S3			S4			S5		
		Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr	Prec	Rec	Thr
on Test Images	car_side	0.9677	1.0000	0.7900	0.9677	1.0000	0.8000	0.9677	1.0000	0.8000	0.9677	1.0000	0.8100	0.9677	1.0000	0.8000
	dollar_bill	1.0000	1.0000	0.7900	1.0000	1.0000	0.8300	1.0000	1.0000	0.8300	1.0000	1.0000	0.8300	1.0000	1.0000	0.8200
	euphonium	0.8947	0.8095	0.7100	1.0000	0.7143	0.7400	1.0000	0.6190	0.7600	1.0000	0.6190	0.7800	0.8889	0.7619	0.7400
	grand_piano	0.8966	0.8667	0.7000	0.9259	0.8333	0.7200	0.9600	0.8000	0.7300	1.0000	0.8000	0.7400	1.0000	0.8000	0.7400
	helicopter	0.9259	0.8621	0.7100	0.9000	0.9310	0.7100	0.9032	0.9655	0.7100	0.9333	0.9655	0.7200	0.8750	0.9655	0.7200
	inline_skate	1.0000	0.6000	0.7600	0.8750	0.7000	0.7600	1.0000	0.7000	0.7700	0.6667	0.8000	0.7700	0.6923	0.9000	0.7700
	laptop	0.9286	0.9630	0.7600	1.0000	1.0000	0.7800	0.8667	0.9630	0.7600	1.0000	1.0000	0.8000	1.0000	1.0000	0.8000
	metronome	1.0000	0.9000	0.7400	1.0000	0.9000	0.7400	1.0000	0.9000	0.7500	1.0000	0.9000	0.7500	1.0000	0.9000	0.7600
	minaret	0.9615	1.0000	0.6900	1.0000	1.0000	0.6900	1.0000	1.0000	0.7100	1.0000	1.0000	0.7300	1.0000	1.0000	0.7900
	Motorbikes	0.7368	0.9333	0.7000	0.6977	1.0000	0.7000	0.6667	1.0000	0.7000	0.7179	0.9333	0.7100	0.7436	0.9667	0.7100
	Total	0.3527	0.7424	0.7500	0.6212	0.5371	0.8000	0.6318	0.5546	0.8000	0.7043	0.5721	0.8100	0.5575	0.6987	0.7800

Table 5.10: Precision & Recall of Categories for Best Total Threshold

		S1		S2		S3		S4		S5	
		Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
on First-Training Images	*Threshold*	*0.79*		*0.80*		*0.81*		*0.80*		*0.80*	
	car_side	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	dollar_bill	1.0000	0.8824	1.0000	0.8235	1.0000	0.8235	1.0000	0.8235	1.0000	0.7647
	euphonium	1.0000	0.7143	1.0000	0.7143	1.0000	0.6667	1.0000	0.7143	1.0000	0.7619
	grand_piano	1.0000	0.5333	1.0000	0.6667	1.0000	0.6333	1.0000	0.7000	1.0000	0.7000
	helicopter	1.0000	0.0690	undef	0.0000	undef	0.0000	1.0000	0.2414	1.0000	0.2414
	inline_skate	1.0000	0.5000	1.0000	0.6000	1.0000	0.6000	1.0000	0.6000	1.0000	0.6000
	laptop	1.0000	0.7778	1.0000	0.8148	1.0000	0.6667	1.0000	0.9259	1.0000	0.8889
	metronome	1.0000	0.7000	1.0000	0.7000	1.0000	0.7000	1.0000	0.7000	1.0000	0.7000
	minaret	1.0000	0.6800	1.0000	0.7600	1.0000	0.7600	1.0000	0.8400	1.0000	1.0000
	Motorbikes	1.0000	0.9000	1.0000	0.9000	1.0000	0.9000	1.0000	0.9000	1.0000	0.9000
	Total	0.6920	0.6769	0.7843	0.6987	0.8280	0.6725	0.8047	0.7555	0.8224	0.7686

		S1		S2		S3		S4		S5	
		Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
on Second-Training Images	*Threshold*	*0.78*		*0.79*		*0.79*		*0.79*		*0.79*	
	car_side	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	dollar_bill	1.0000	0.5882	1.0000	0.7059	1.0000	0.7647	1.0000	0.7647	1.0000	0.8824
	euphonium	1.0000	0.6190	1.0000	0.6190	1.0000	0.6667	1.0000	0.7619	1.0000	0.7143
	grand_piano	1.0000	0.3667	1.0000	0.4667	1.0000	0.5000	1.0000	0.5000	1.0000	0.5000
	helicopter	1.0000	0.0690	1.0000	0.0690	1.0000	0.1724	1.0000	0.2759	1.0000	0.4138
	inline_skate	1.0000	0.3000	1.0000	0.7000	1.0000	1.0000	0.9091	1.0000	0.8333	1.0000
	laptop	1.0000	0.8519	1.0000	0.8889	0.9565	0.8148	1.0000	0.9630	1.0000	0.9630
	metronome	1.0000	0.3000	1.0000	0.5000	1.0000	0.5000	1.0000	0.6000	1.0000	0.7000
	minaret	1.0000	0.6400	1.0000	0.8000	1.0000	0.8800	1.0000	0.9600	1.0000	1.0000
	Motorbikes	1.0000	0.6333	1.0000	0.6667	1.0000	0.7000	1.0000	0.7000	1.0000	0.7000
	Total	0.5936	0.5677	0.7277	0.6419	0.7406	0.6856	0.7578	0.7380	0.7619	0.7686

		S1		S2		S3		S4		S5	
		Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
on Test Images	*Threshold*	*0.75*		*0.80*		*0.80*		*0.81*		*0.78*	
	car_side	0.9677	1.0000	0.9677	1.0000	0.9677	1.0000	0.9677	1.0000	0.9677	1.0000
	dollar_bill	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	euphonium	1.0000	0.5714	1.0000	0.5714	1.0000	0.5714	1.0000	0.5714	1.0000	0.5714
	grand_piano	1.0000	0.6333	1.0000	0.3000	1.0000	0.3333	1.0000	0.3000	1.0000	0.4667
	helicopter	1.0000	0.5517	undef	0.0000	1.0000	0.0345	1.0000	0.0690	1.0000	0.4828
	inline_skate	1.0000	0.6000	1.0000	0.2000	1.0000	0.3000	1.0000	0.3000	0.6667	0.6000
	laptop	0.8710	1.0000	1.0000	0.8148	1.0000	0.7407	1.0000	0.8889	1.0000	1.0000
	metronome	1.0000	0.7000	1.0000	0.5000	1.0000	0.5000	1.0000	0.5000	1.0000	0.7000
	minaret	1.0000	0.9200	1.0000	0.9200	1.0000	0.9600	1.0000	0.9600	1.0000	1.0000
	Motorbikes	1.0000	0.4333	1.0000	0.1000	1.0000	0.1667	1.0000	0.1667	1.0000	0.2667
	Total	0.3527	0.7424	0.6212	0.5371	0.6318	0.5546	0.7043	0.5721	0.5575	0.6987

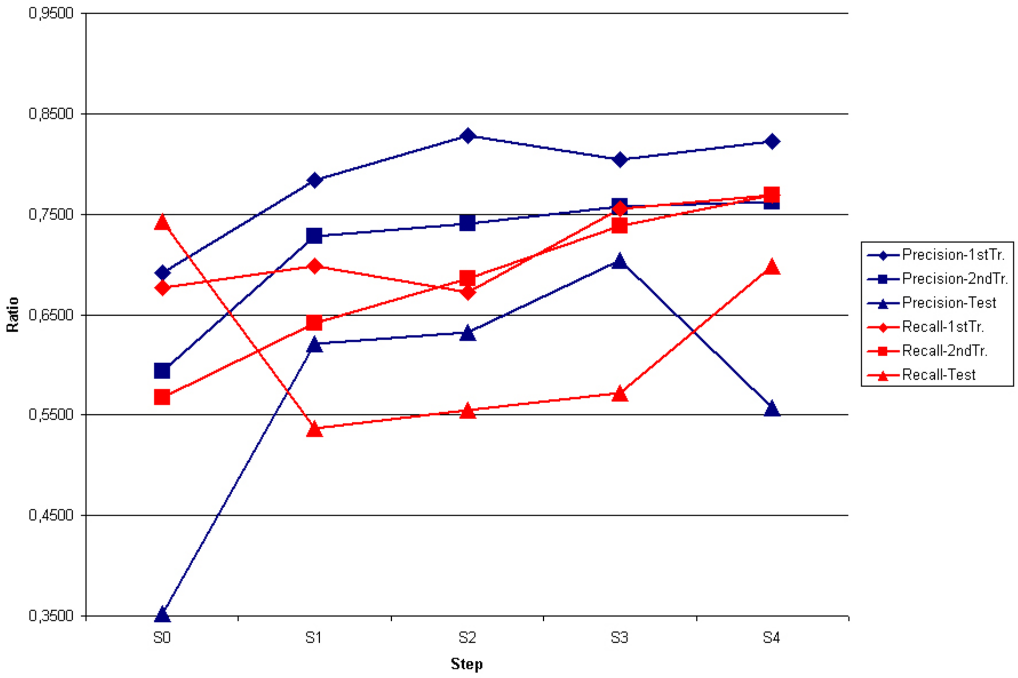


Figure 5.21: Best Precision & Recall of Total System with Thresholding

each category and total system is given. As in the previous one, the values of total system does not mean the average of all categories. In Figure 5.25, the change in precision and recall values of total system is represented according to the test steps. As seen in the graphs and tables, using a ranking mechanism instead of thresholding makes the system more successful.

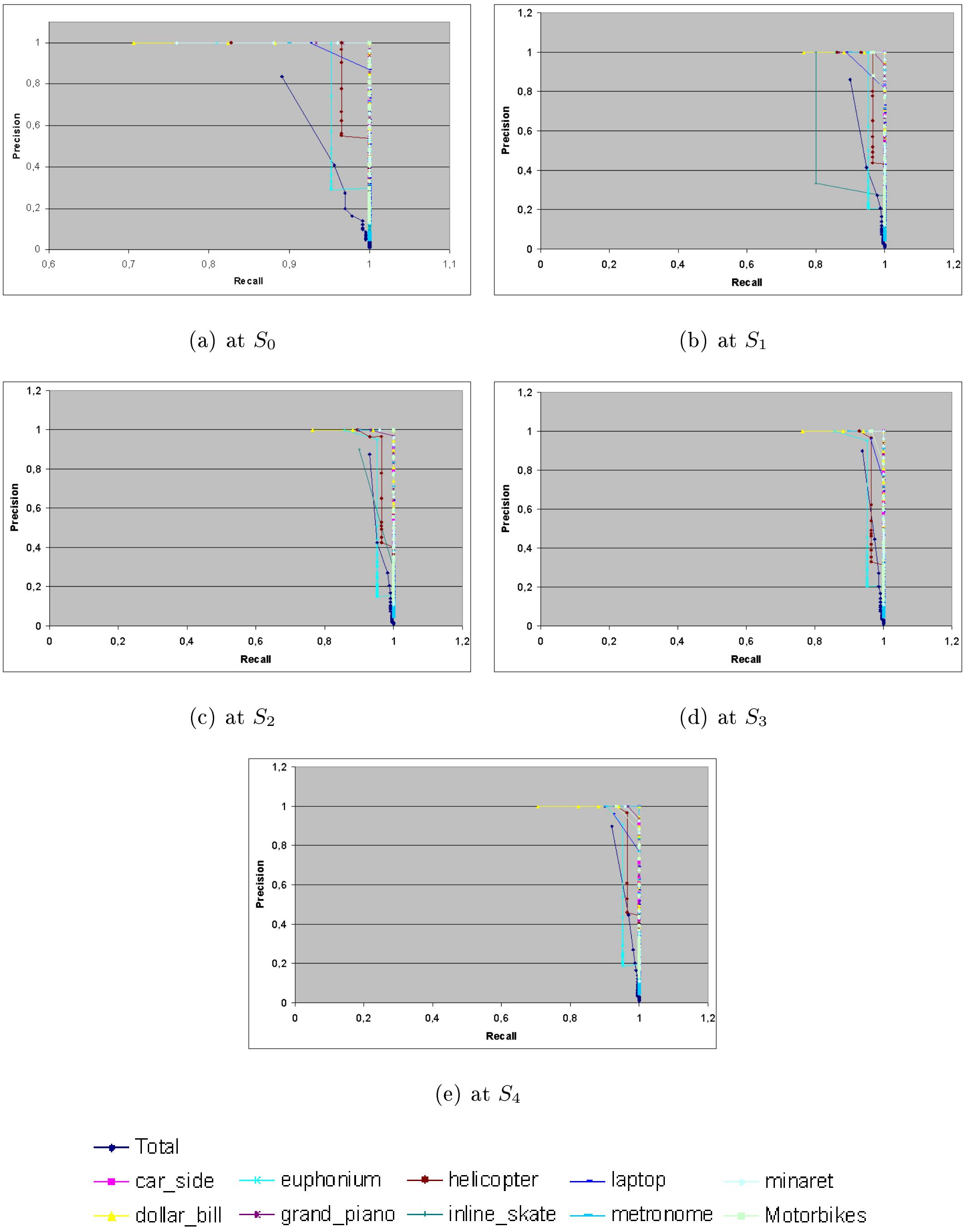
Precision & Recall of Categories for Best Total Rank

Lastly, in this part, precision and recall values that are calculated by using the 'ranking' at the best point of total system is given for each category (Table 5.12). Since these results is calculated according to the best success of whole system and ranking mechanism is accepted better than thresholding, these results are mostly used in evaluation and comparison section.

5.3.3 Video Trial and Time Measurement

As declared in the previous chapters, the system can be used both for object extraction from videos and images as a standalone system and for automatic content indexing with any more complicated CBIR system. In other words, dealing with a video is an important feature of the system. So to see the whole cycle of processing a video and total execution time for the process in important.

For the test, a video file taken from the Open Video Project [30] is used. The selected



(a) at S_0

(b) at S_1

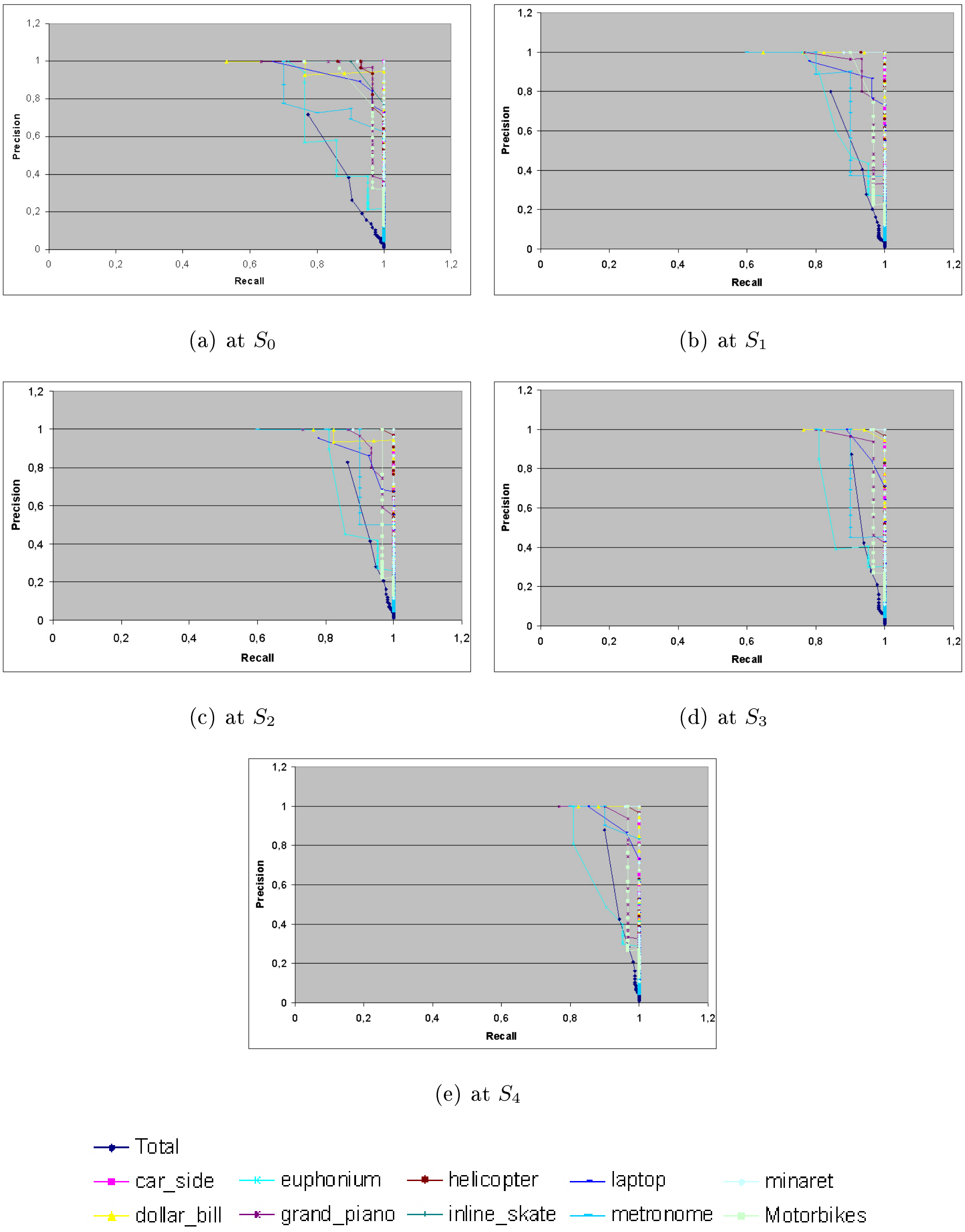
(c) at S_2

(d) at S_3

(e) at S_4

(f) Legend

Figure 5.22: Precision vs Recall Curve for All Categories with Ranking, on First-Training Images



(a) at S_0

(b) at S_1

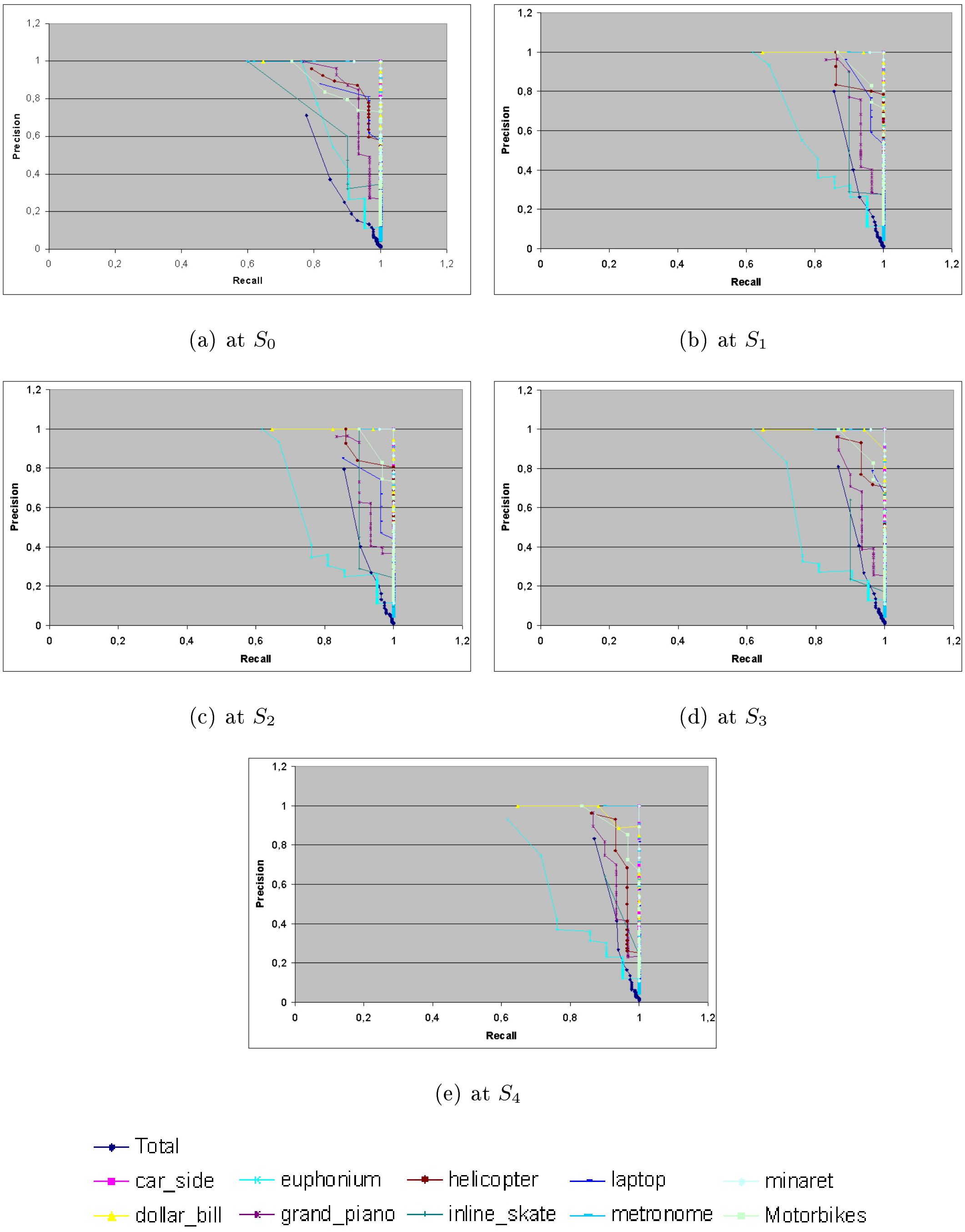
(c) at S_2

(d) at S_3

(e) at S_4

(f) Legend

Figure 5.23: Precision vs Recall Curve for All Categories with Ranking, on Second-Training Images



(a) at S_0

(b) at S_1

(c) at S_2

(d) at S_3

(e) at S_4

(f) Legend

Figure 5.24: Precision vs Recall Curve for All Categories with Ranking, on Test Images

Table 5.11: Precision & Recall of Categories with Ranking

		S1			S2			S3			S4			S5		
		Prec	Rec	R	Prec	Rec	R	Prec	Rec	R	Prec	Rec	R	Prec	Rec	R
on First-Training Images	car_side	1.0000	1.0000	1	1.0000	1.0000	1	1.0000	1.0000	1	1.0000	1.0000	1	1.0000	1.0000	1
	dollar_bill	1.0000	1.0000	6	1.0000	1.0000	5	1.0000	1.0000	5	1.0000	1.0000	5	1.0000	1.0000	5
	euphonium	1.0000	0.9524	2	1.0000	0.9524	2	0.9524	0.9524	2	0.9524	0.9524	2	1.0000	0.9048	1
	grand_piano	1.0000	1.0000	6	1.0000	0.9667	3	0.9677	1.0000	3	1.0000	1.0000	2	1.0000	0.9667	2
	helicopter	1.0000	0.9655	2	1.0000	0.9655	3	0.9655	0.9655	3	0.9655	0.9655	2	0.9655	0.9655	2
	inline_skate	1.0000	1.0000	1	1.0000	0.8000	1	0.9000	0.9000	1	0.9091	1.0000	1	0.8333	1.0000	1
	laptop	1.0000	0.9259	1	1.0000	0.8889	1	1.0000	1.0000	1	0.9630	0.9630	1	0.9615	0.9259	1
	metronome	1.0000	1.0000	3	1.0000	1.0000	3	1.0000	1.0000	3	1.0000	1.0000	3	1.0000	1.0000	3
	minaret	1.0000	1.0000	5	1.0000	1.0000	4	1.0000	1.0000	3	1.0000	1.0000	3	1.0000	0.9600	1
	Motorbikes	1.0000	1.0000	1	1.0000	0.9667	1	1.0000	1.0000	1	1.0000	0.9667	1	1.0000	0.9333	1
	Total	0.8361	0.8908	1	0.8619	0.8996	1	0.8765	0.9301	1	0.8996	0.9389	1	0.8979	0.9214	1
		S1			S2			S3			S4			S5		
		Prec	Rec	R	Prec	Rec	R	Prec	Rec	R	Prec	Rec	R	Prec	Rec	R
on Second-Training Images	car_side	1.0000	1.0000	1	0.9677	1.0000	1	0.9677	1.0000	1	0.9375	1.0000	1	0.9677	1.0000	1
	dollar_bill	0.9444	1.0000	6	1.0000	1.0000	5	0.9444	1.0000	5	0.9444	1.0000	5	1.0000	1.0000	4
	euphonium	0.9412	0.7619	2	1.0000	0.7619	1	1.0000	0.8095	1	1.0000	0.8095	1	1.0000	0.8095	1
	grand_piano	0.9667	0.9667	6	0.9655	0.9333	3	0.9643	0.9000	3	0.9355	0.9667	3	0.9355	0.9667	3
	helicopter	1.0000	0.9310	2	1.0000	1.0000	2	0.9667	1.0000	2	0.9667	1.0000	2	0.9667	1.0000	2
	inline_skate	1.0000	0.9000	1	1.0000	1.0000	1	1.0000	1.0000	1	0.8333	1.0000	1	0.8333	1.0000	1
	laptop	0.8929	0.9259	2	0.8667	0.9630	2	0.8621	0.9259	2	1.0000	0.8889	1	1.0000	0.8519	1
	metronome	1.0000	0.7000	1	0.9000	0.9000	6	1.0000	0.9000	3	1.0000	0.9000	3	1.0000	0.9000	3
	minaret	1.0000	1.0000	4	1.0000	1.0000	2	1.0000	1.0000	2	1.0000	1.0000	2	1.0000	1.0000	2
	Motorbikes	0.9630	0.8667	1	1.0000	0.9000	1	1.0000	0.9667	1	1.0000	0.9667	1	1.0000	0.9667	1
	Total	0.7166	0.7729	1	0.8008	0.8428	1	0.8285	0.8646	1	0.8734	0.9039	1	0.8803	0.8996	1
		S1			S2			S3			S4			S5		
		Prec	Rec	R	Prec	Rec	R	Prec	Rec	R	Prec	Rec	R	Prec	Rec	R
on Test Images	car_side	1.0000	1.0000	1	1.0000	1.0000	1	1.0000	1.0000	1	1.0000	1.0000	1	1.0000	1.0000	1
	dollar_bill	1.0000	1.0000	6	1.0000	1.0000	5	1.0000	1.0000	5	1.0000	0.9412	3	0.8947	1.0000	6
	euphonium	1.0000	0.7619	3	0.9333	0.6667	2	0.9333	0.6667	2	1.0000	0.6190	1	0.9286	0.6190	1
	grand_piano	0.9630	0.8667	3	0.9630	0.8667	2	0.9310	0.9000	3	0.9630	0.8667	1	0.9630	0.8667	2
	helicopter	0.8710	0.9310	4	1.0000	0.8621	1	1.0000	0.8621	1	0.9310	0.9310	2	0.9310	0.9310	2
	inline_skate	1.0000	0.6000	1	0.9000	0.9000	1	1.0000	0.9000	1	0.6429	0.9000	1	0.6429	0.9000	1
	laptop	0.8125	0.9630	2	0.9600	0.8889	1	0.8519	0.8519	1	0.7879	0.9630	1	0.8182	1.0000	1
	metronome	1.0000	1.0000	4	1.0000	1.0000	3	1.0000	1.0000	3	1.0000	1.0000	4	1.0000	1.0000	2
	minaret	1.0000	1.0000	2	1.0000	1.0000	2	1.0000	1.0000	2	1.0000	1.0000	2	1.0000	1.0000	1
	Motorbikes	1.0000	0.7333	1	1.0000	0.8667	1	1.0000	0.9000	1	1.0000	0.8667	1	1.0000	0.8333	1
	Total	0.7120	0.7773	1	0.8000	0.8559	1	0.7967	0.8559	1	0.8115	0.8646	1	0.8326	0.8690	1

Table 5.12: Precision & Recall of Categories for Best Total Rank

		S1		S2		S3		S4		S5	
		Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
on First-Training Images	car_side	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	dollar_bill	1.0000	0.7059	1.0000	0.7647	1.0000	0.7647	1.0000	0.7647	1.0000	0.7059
	euphonium	1.0000	0.8095	1.0000	0.8571	1.0000	0.8571	1.0000	0.8571	1.0000	0.9048
	grand_piano	1.0000	0.9333	1.0000	0.8667	1.0000	0.9000	1.0000	0.9667	1.0000	0.9000
	helicopter	1.0000	0.8276	1.0000	0.8621	1.0000	0.8966	1.0000	0.9310	1.0000	0.9310
	inline_skate	1.0000	1.0000	1.0000	0.8000	0.9000	0.9000	0.9091	1.0000	0.8333	1.0000
	laptop	1.0000	0.9259	1.0000	0.8889	1.0000	1.0000	0.9630	0.9630	0.9615	0.9259
	metronome	1.0000	0.9000	1.0000	0.9000	1.0000	0.9000	1.0000	0.9000	1.0000	0.9000
	minaret	1.0000	0.7600	1.0000	0.9600	1.0000	0.9600	1.0000	0.9600	1.0000	0.9600
	Motorbikes	1.0000	1.0000	1.0000	0.9667	1.0000	1.0000	1.0000	0.9667	1.0000	0.9333
	Total	0.8361	0.8908	0.8619	0.8996	0.8765	0.9301	0.8996	0.9389	0.8979	0.9214
		S1		S2		S3		S4		S5	
		Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
on Second-Training Images	car_side	1.0000	1.0000	0.9677	1.0000	0.9677	1.0000	0.9375	1.0000	0.9677	1.0000
	dollar_bill	1.0000	0.5294	1.0000	0.6471	1.0000	0.7647	1.0000	0.7647	1.0000	0.8235
	euphonium	1.0000	0.7143	1.0000	0.7619	1.0000	0.8095	1.0000	0.8095	1.0000	0.8095
	grand_piano	1.0000	0.6333	1.0000	0.7667	1.0000	0.7333	1.0000	0.8000	1.0000	0.7667
	helicopter	1.0000	0.8621	1.0000	0.9310	1.0000	0.9655	1.0000	0.9655	1.0000	0.9655
	inline_skate	1.0000	0.9000	1.0000	1.0000	1.0000	1.0000	0.8333	1.0000	0.8333	1.0000
	laptop	1.0000	0.6667	0.9545	0.7778	0.9545	0.7778	1.0000	0.8889	1.0000	0.8519
	metronome	1.0000	0.7000	1.0000	0.6000	1.0000	0.6000	1.0000	0.8000	1.0000	0.8000
	minaret	1.0000	0.7600	1.0000	0.8800	1.0000	0.8800	1.0000	0.9600	1.0000	0.9600
	Motorbikes	0.9630	0.8667	1.0000	0.9000	1.0000	0.9667	1.0000	0.9667	1.0000	0.9667
	Total	0.7166	0.7729	0.8008	0.8428	0.8285	0.8646	0.8734	0.9039	0.8803	0.8996
		S1		S2		S3		S4		S5	
		Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec	Rec
on Test Images	car_side	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	dollar_bill	1.0000	0.6471	1.0000	0.6471	1.0000	0.6471	1.0000	0.6471	1.0000	0.6471
	euphonium	1.0000	0.6190	1.0000	0.6190	1.0000	0.6190	1.0000	0.6190	0.9286	0.6190
	grand_piano	1.0000	0.7333	0.9615	0.8333	0.9615	0.8333	0.9630	0.8667	1.0000	0.8333
	helicopter	0.9583	0.7931	1.0000	0.8621	1.0000	0.8621	0.9615	0.8621	0.9615	0.8621
	inline_skate	1.0000	0.6000	0.9000	0.9000	1.0000	0.9000	0.6429	0.9000	0.6429	0.9000
	laptop	0.8800	0.8148	0.9600	0.8889	0.8519	0.8519	0.7879	0.9630	0.8182	1.0000
	metronome	1.0000	0.6000	1.0000	0.9000	1.0000	0.9000	1.0000	0.8000	1.0000	0.9000
	minaret	1.0000	0.9200	1.0000	0.9600	1.0000	0.9600	1.0000	0.9600	1.0000	1.0000
	Motorbikes	1.0000	0.7333	1.0000	0.8667	1.0000	0.9000	1.0000	0.8667	1.0000	0.8333
	Total	0.7120	0.7773	0.8000	0.8559	0.7967	0.8559	0.8115	0.8646	0.8326	0.8690

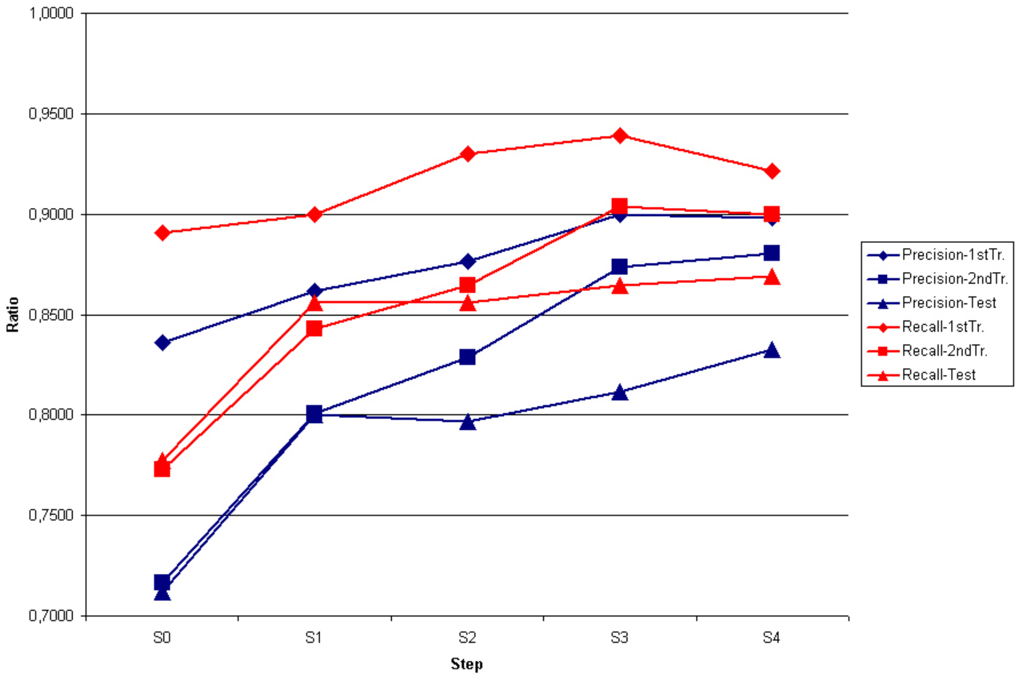


Figure 5.25: Best Precision & Recall of Total System with Ranking

video is related with aircrafts and helicopters, 352x240 pixels in size and 4:45 minutes in length.

Firstly, keyframes of the video are extracted with the IBM MPEG-7 Annotation Tool Figure 5.2. The tool extracted 48 frames with dimesions 352x240 pixels. The images are shown in Figure 5.26.

Then, the system makes segmentation on these keyframes and generates candidate objects by combining the neigboring segments. Sample segmentation results are shown in Figure 5.27. In Figure 5.27(b) and Figure 5.27(e) two different keyframes are given. The segments obtained from the images are given in Figure 5.27(a) and Figure 5.27(d). By combination of some neigboring segments, the candidate objects are acquired. Sample two are given in Figure 5.27(c) and Figure 5.27(f).

As the segmentation parameter, 10 is used (The images are segmented into 10 parts). Also, to see the effect of segmentation size choise, an example calculation of process times is given in Table 5.13.

Lastly, all object candidates obtained from segment grouping are sent to the classifier and decisions are calculated. For the sample images given in Figure 5.27(b) and Figure 5.27(e), the top-10 decisions are given in Table 5.14

After processing the sample video, all execution times are obtained as given in Table 5.15.

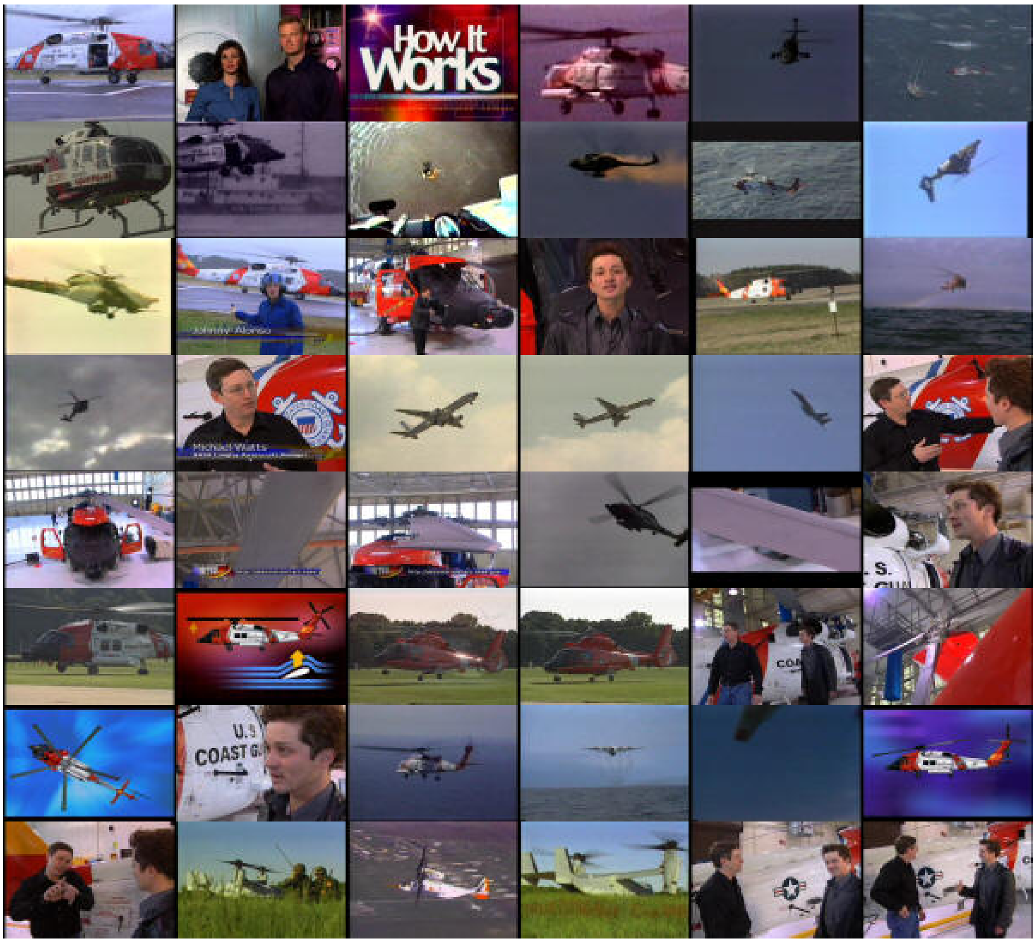


Figure 5.26: Keyframes Extracted From The Test Video

Table 5.13: Segmentation Times According to Segment Size

No.Of Segments	Segmentation Time	Grouping Time	No.Of Candidate Objects
5	15	7	28
8	20	42	209
10	22	143	729
15	41	3986	27,762
20	56	106,578	826,684

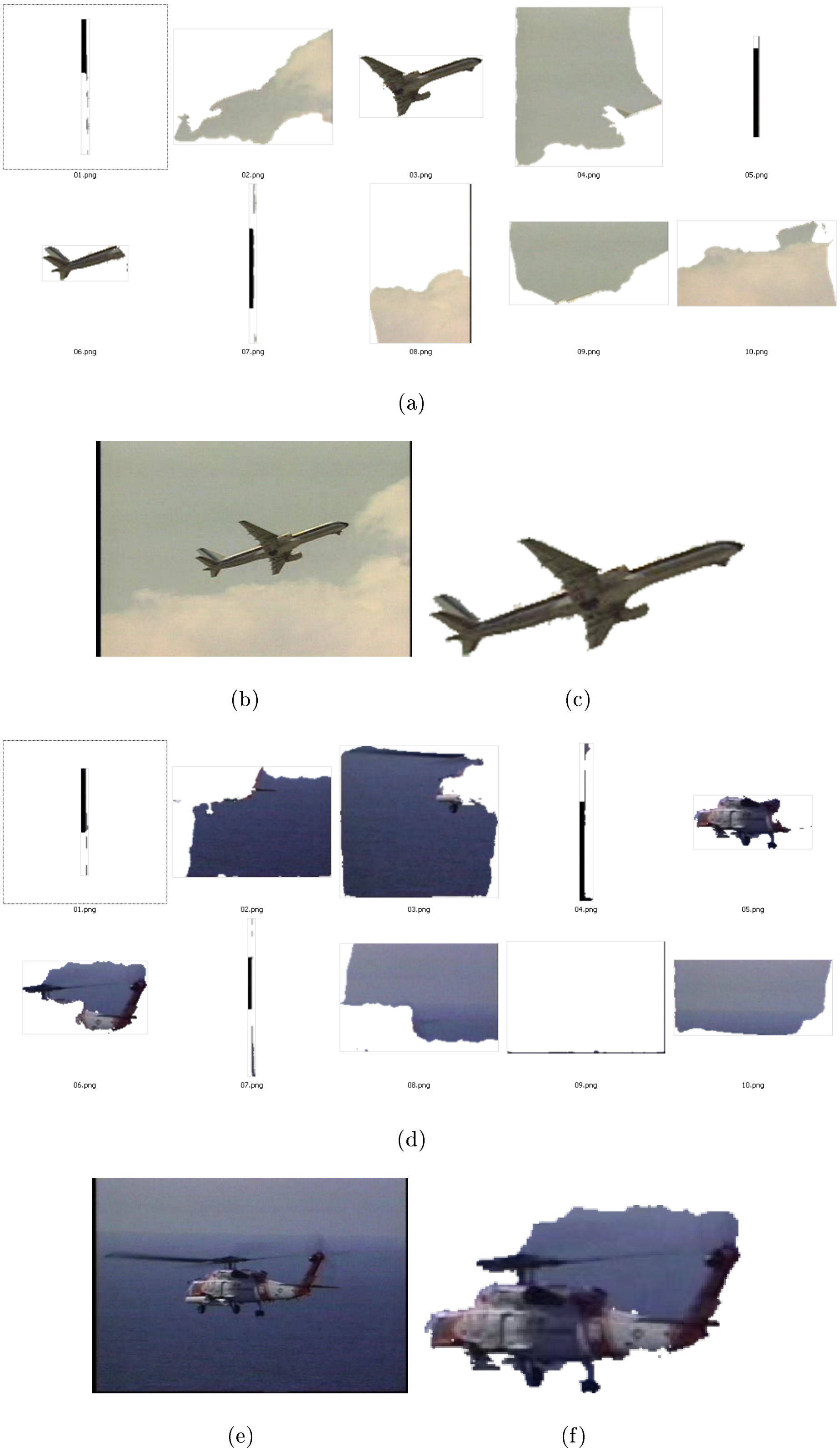


Figure 5.27: Sample Segmentation Results: (a)Segments of image in (b), (b)A keyframe with an airplane, (c)A candidate object combined from (b), (d)Segments of image in (e), (e)A keyframe with a helicopter, (f)A candidate object combined from (e)

In the table, item count in 'keyframe extraction' shows the number of videos used, the one in 'segmentation' and 'segment grouping' show the number of keyframes. Besides the one in the 'classification' shows the number of candidate objects.

Table 5.14: Sample Category Decisions

Figure 5.27(c)		Figure 5.27(f)	
Category	Decision	Category	Decision
Leopards	0,78	helicopter	0,73
airplanes	0,73	water_lilly	0,71
gerenuk	0,72	bass	0,7
ceiling_fan	0,71	inline_skate	0,7
cougar_body	0,71	pigeon	0,7
ant	0,69	rhino	0,7
car_side	0,68	hawksbill	0,69
electric_guitar	0,68	panda	0,69
flamingo	0,65	stapler	0,69
saxophone	0,64	laptop	0,68

Table 5.15: Total Processing Time of a Video. (* Expected total time)

Process	Total Time(s)	Item count	Avg. Time(s)
Keyframe extraction	32	1	32
Segmentation	1,152	48	24
Segment Grouping	6,816	48	142
Classification	864,600*	34,584	25
Total	872,600		

5.4 Evaluation and Comparison

The evaluation on the test results can be done both for performance of the system and for execution times of processes.

5.4.1 Performance

In this study, a GA based model that uses multi features of MPEG-7 with a BDRF methodology is proposed. So for the performance evaluation, firstly effect of GA to the system should be examined. Also it should be compared if using a BDRF methodology with multiple features is better than using a single BRF and if the performance of using multiple features together is better than the performance of each MPEG-7 descriptor. As a dataset based comparison, the system is compared with the other studies that uses same dataset (CalTech 101).

GA Effect

According to the results given in all tables in 5.3.2 that are related to the performance of the classifier, the positive effect of GA is obviously seen. Generally, all of decisions given, ANMRR, precision and recall values increased through next steps which are defined according to the number of performed genetic operations.

Average decision(Table 5.7) in the system increased by approximately 4% on the First-Training set, 4.7% on the Second-Training set and 3.7% on the Test set. The increase in Second-Training is a requirement for the implementation so it is not considerable. But the increases in other two sets shows the success of the system.

The ANMRR values displays decreasing graph (Figure 5.17). Decreasing values of ANMRR represent improving performance. The improvement on the First-Training set is approximately 34.8%, Second-Training set 58.4% and Test set 46.6%. The precision and recall values are not so much different, the values tend to increase during steps. It is normal to obtain more increase in Second-Training, but even obtaining increase in First-Training and Test sets can be accepted as a success of GA methodology.

Also, the graphs given for categories (Figure 5.18, Figure 5.19, Figure 5.20, Figure 5.22, Figure 5.23 and Figure 5.24) can be examined to see the effect on the categories. Considering the sliding of points in the graphs to the right and up of the graph in consecutive steps, it can be stated that the values are increasing. But lines of some the categories (euphonium, helicopter and inline_skate) reside at worse points than others. The behaviour of these categories can be explained with neither training sizes nor feature importances used since a direct relation between them cannot be constructed by using the data values inTable 5.2 and Table 5.3.

BRDF vs BRF Comparison

Uysal et al. [61] propose to use a single best representative feature for each category and calculate the best one according to performance of each feature. This study proposes a statistical method to find BRDF values for multiple features.

Table 5.16: Performance of the BRF approach [61]

Class	# of Correctly Labeled images (out of 5)
Antelope	3
Horse	3
Bird	2
Leopard	5
Cotton Texture	2
Plane	4
Fish	4
Polar Bear	4
Flag	3
Sun Set	4
Recall	0.68

Although different dataset are used with Uysal et al. [61], the recall values is compared. Uysal et al. obtains a recall of 0.68 (Table 5.16), in this study 0.6987 recall is obtained with thresholding and 0.8690 recall is obtained with ranking.

MPEG-7 Comparison

In [7], Manjunath et al. calculates ANMRR values of different color and texture descriptors of MPEG-7. Since ANMRR is a performance measure that is independent from the dataset, the results can be compared properly.

For the performances of the MPEG-7 descriptors, Table 5.17 and Table 5.18 are extracted as a summary from [7] and [9] respectively.

The ANMRR values obtained in this study (Figure 5.17) is 0.0349, 0.0513 and 0.0764 on First-Training, Second-Training and Test images respectively.

Table 5.17: ANMRR Values for MPEG-7 Decriptors [7]

MPEG-7 Descriptor	Performance
Scalable Color	ANMRR in [0.05, 0.1]
Dominant Color	ANMRR in [0.197, 0.252]
Color Layout	ANMRR in [0.15, 0.20]
Color Structure	ANMRR in [0.046, 0.105]
Homogeneous Texture	Precision ≈ 77%
Edge Histogram	ANMRR in [0.28, 0.36]

Table 5.18: ANMRR Values for MPEG-7 Decriptors [9]

MPEG-7 Descriptor	Performance
Scalable Color	ANMRR in [0.05, 0.11]
Dominant Color	ANMRR in [0.16, 0.25]
Color Layout	ANMRR in [0.36, 0.50]
Color Structure	ANMRR in [0.05, 0.11]

CalTech 101 Comparison

In [62], Wang et al. give the mean recognition rates obtained in all studies using CalTech dataset in a graph. The graph is given in Figure 5.28.

Figure 5.28 gives mean recognition rates for different number of training images. Recognition rate means what ratio of given images are classifed correctly and equals to the recall in this study. The results obtained in this study are also put on the graph in Figure 5.28. The graph shows that the obtained results in this study is better than the results of the other systems.

5.4.2 Execution Times

Another evaluation criteria is the execution times. The execution times of the system is given in Sub-section 5.3.1 and Sub-section 5.3.3 in detail.

Considering the training execution times (Table 5.4) which are obtained for more than 2250 images, the average time for each training image seems acceptable. In fact the improvements performed on Feature Value Normalization and Feature Importance Determination processes provided this situation. For example, at the beginning of the implementations,

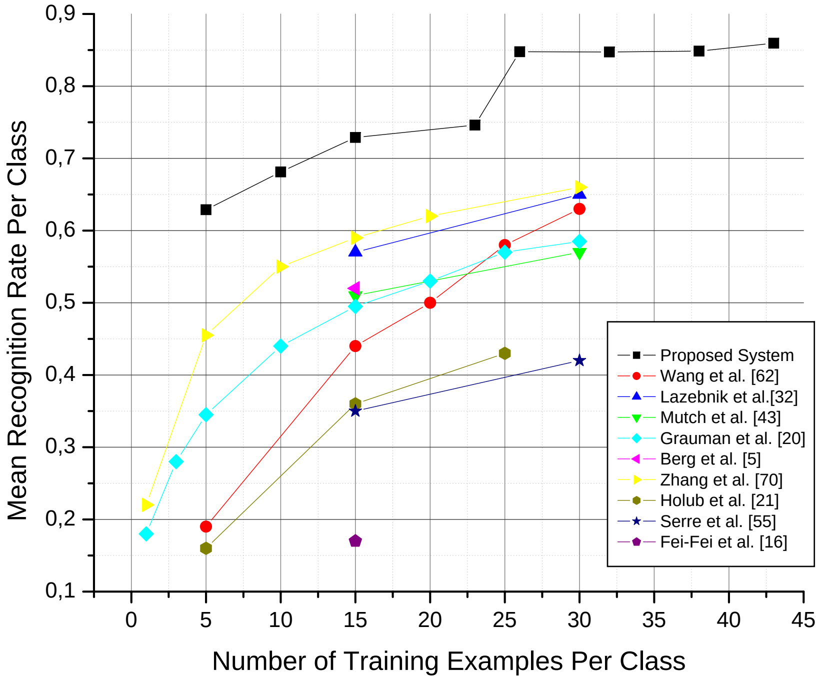


Figure 5.28: Performances of Studies Using Caltech 101 Dataset

the normalization process took 150 times longer. By using heuristic approaches in the implementations, the execution time is decreased. The longest time is spent by importance determination, but it is not possible to decrease that time since almost all of it is spent in the XM Software queries.

Also, time inefficiencies occur during the segmentation process. In tests, the images are segmented into 10 pieces. In complex images, this parameter cannot be enough, it should be increased. But as seen on the results given in Table 5.13, increasing the segmentation parameter increases the time spent for segment grouping exponentially. According to the test results given in the table, it is not possible to use big numbers for the parameter. This means either the segment grouping method should be improved or the parameter should be used up to 10 or closer to 10. For this study, the parameter is chosen 10 at maximum and an improved segment grouping method is left as a future work.

Considering the execution times given in Table 5.15, it seems that total time for processing the sample video is approximately 10 days, which is not acceptable. This time ineffiency is mostly caused by large number of candidate objects. As mentioned before, if the segment grouping methodology is improved, the number of candidate objects can decrease. Thus, processing a video can be handled in an acceptable shorter time.

Also, according to the results given for classification in the same table, the average time for classification seems acceptable. But it should be better, if the system can be used as a standalone system. In fact all of this time is spent for the search operations performed on XM Software. The search on XM Software is performed for each feature used, so if the number of features is increased the time for classification increases. But it is not possible to make an improvement on the system except making improvement on the XM or using something other than XM Software. This situation can create a future work.

CHAPTER 6

CONCLUSION AND FUTURE WORK

In this thesis, a Genetic Algorithm based object extraction and classification mechanism is proposed and developed for extracting the content of the videos and images. In the methodology, the object extraction problem is attacked as a classification problem. For the classification problem, a Genetic Algorithm based classifier is proposed and described. By using Normalized-cut segmentation and defining each object with the Best Representative and Discriminative Feature model which contains MPEG-7 descriptors, candidate objects are obtained. The classifier makes decisions by using these features and BRDF model. By using genetic operations of GA, the classifier improves itself in time. In addition to these, the system supports fuzziness by making multiple categorization and giving fuzzy decisions on the objects. Externally from the base model, a statistical feature importance determination method is proposed to generate BRDF model of the categories automatically.

In the thesis, a platform independent application for the proposed system is also implemented. Throughout the experiments by using the implemented application, the proposed system achieves much bettter performances compared to the other approaches on using single MPEG-7 descriptors as feature, the studies using Best (single) Representative Feature and the retrieval systems using CalTech 101 image dataset. Furthermore, the test results clearly shows the positive effect of GA model.

Although only visual descriptors are considered in this study, by using the same GA based model, an alternative audio recognition system can be accomplished. Moreover, the model can be turned into a system that provides all of audio, visual, text caption features in videos. The only thing to realize such ideas is a tool that decomposes the audio data or text captions from videos in some meaningful manner. This can be a good future direction.

Other future works can be listed as follows:

- The segment grouping method described in Section 4.5 is inefficient in time considerations when big numbers of segments are used. An improvement on the method can be performed or a more efficient model can be proposed to shorten the process times of the system.

- In the current system, to import a video to the system, keyframes should be extracted first. Then all the keyframes are given to the application. This process can be merged into a single entry point to the system so that videos are directly imported into the system.

- In the current system, 8 of MPEG-7 visual descriptors are implemented, other visual descriptors can also be implemented.

- Making queries on XM Software is costly (approximately 25 seconds totally for 8 features), this cost can be reduced. Although it is not possible to change the execution time without making changes on XM Software application, a middle layer can be adapted for situations like caching mostly used feature values or handling predictible cases. But in this case, the middle layer system should deal with the low level features which is not a desired case. This can be planned as a caching wrapper to the XM Software external from the proposed system in this study.

- To shorten the execution time, parallel machines can be used.

REFERENCES

[1] Erwin M. Bakker, Thomas S. Huang, Michael S. Lew, Nicu Sebe, and Xiang Sean Zhou, editors. *Image and Video Retrieval, Second International Conference, CIVR 2003, Urbana-Champaign, IL, USA, July 24-25, 2003, Proceedings*, volume 2728 of *Lecture Notes in Computer Science*. Springer, 2003.

[2] Ilaria Bartolini and Marco Patella. Warp: Accurate retrieval of shapes using phase of fourier descriptors and time warping distance. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(1):142–147, 2005. Member-Paolo Ciaccia.

[3] David Beasley, David R. Bull, and Ralph R. Martin. An overview of genetic algorithms: Part 1 fundamentals. *University Computing*, 15(2):58–69, 1993.

[4] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[5] Alexander C. Berg, Tamara L. Berg, and Jitendra Malik. Shape matching and object recognition using low distortion correspondences. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 26–33, Washington, DC, USA, 2005. IEEE Computer Society.

[6] M. Bober. Mpeg-7 visual shape descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):716–719, Jun 2001.

[7] B.S.Manjunath, J.-R.Ohm, V.V.Vasudevan, and A.Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703–715, Jun 2001.

[8] A. Cavallaro and T. Ebrahimi. Object-based video: extraction tools, evaluation metrics and applications. In *Proc. of SPIE Visual Communications and Image Processing*, Lugano, Switzerland, July 2003.

[9] Leszek Cieplinski. Mpeg-7 color descriptors and their applications. In *CAIP '01: Proceedings of the 9th International Conference on Computer Analysis of Images and Patterns*, pages 11–20, London, UK, 2001. Springer-Verlag.

[10] Timothee Cour, Florence Benezit, and Jianbo Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 1124–1131, Washington, DC, USA, 2005. IEEE Computer Society.

[11] Timothee Cour, Florence Benezit, and Jianbo Shi. Multiscale normalized cuts segmentation toolbox for matlab version 1.0, Aug 2006.
http://www.seas.upenn.edu/~timothee [Online; accessed 08-June-2007].

[12] Ritendra Datta, Jia Li, and James Z. Wang. Content-based image retrieval: approaches and trends of the new age. In *MIR '05: Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262, New York, NY, USA, 2005. ACM.

[13] Y. Deng, B. Manjunath, C. Kenney, M. Moore, and H. Shin. An efficient color representation for image retrieval, IEEE Transactions on Image Processing, vol.10, (no.1), IEEE, Jan. 2001. p.140-147.

[14] A. Doulamis, Y. Avrithis, N. Doulamis, and S. Kollias. A genetic algorithm for efficient video content representation. In *Proceedings of IMACS/IFAC International Symposium on Soft Computing in Engineering Applications (SOFTCOM '98)*, 1998.

[15] E.Hadjidemetriou, M.D.Grossberg, and S.K.Nayar. Multiresolution histograms and their use for recognition. *Transactions on Pattern Analysis and Machine Intelligence*, 26(7):831–847, July 2004.

[16] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *cvprw*, 12:178, 2004.

[17] Li Fei-Fei, Rob Fergus, and Pietro Perona. One-shot learning of object categories. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(4):594, 2006.

[18] Payel Ghosh and Melanie Mitchell. Segmentation of medical images using a genetic algorithm. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1171–1178, New York, NY, USA, 2006. ACM.

[19] Wojciech Gradkowski. Jmatio java library v0.2, 2006.
http://www.sourceforge.net/projects/jmatio [Online; accessed 12-September-2007].

[20] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1458–1465, Washington, DC, USA, 2005. IEEE Computer Society.

[21] Alex D. Holub, Max Welling, and Pietro Perona. Combining generative models and fisher kernels for object recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 136–143, Washington, DC, USA, 2005. IEEE Computer Society.

[22] S. Hwang, E.Y. Kim, S.H. Park, and H.J. Kim. Object extraction and tracking using genetic algorithms. In *2001 IEEE Signal Processing Society ICIP 2001 conf. proc.*, volume 2, pages 383–386, Thessaloniki, Greece, 2001.

[23] I.Aravind, C.Chandra, M.Guruprasad, P.S.Dev, and R.D.S.Samuel. Implementation of image segmentation and reconstruction using genetic algorithms. *Industrial Technology, 2002. IEEE ICIT '02. 2002 IEEE International Conference on*, 2:970–975 vol.2, 11-14 Dec. 2002.

[24] Sangoh Jeong, Chee Sun Won, and Robert M. Gray. Image retrieval using color histograms generated by gauss mixture vector quantization. *Comput. Vis. Image Underst.*, 94(1-3):44–66, 2004.

[25] Eiji Kasutani and Akio Yamada. The mpeg-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval. In *ICIP (1)*, pages 674–677, 2001.

[26] Eun Yi Kim and Keechul Jung. Object detection and removal using genetic algorithms. In Zhang et al. [69], pages 411–421.

[27] Eun Yi Kim and Se Hyun Park. Automatic extraction of moving objects using distributed genetic algorithms. In Liu et al. [34], pages 262–269.

[28] E.Y. Kim, S.H. Park, and H.J. Kim. A genetic algorithm-based segmentation of markov random field modeled images. *SPLetters*, 7(11):301–303, November 2000.

[29] Whoi-Yul Kim and Yong-Sung Kim. A region-based shape descriptor using Zernike moments, Signal Processing: Image CommunicationVolume 16, Issues 1-2, , September 2000, Pages 95-102.

[30] Interaction Design Laboratory. Open video project. School of Information and Library Science, University of North Carolina at Chapel Hill.
http://www.open-video.org/ [Online; accessed 02-December-2007].

[31] Longin Jan Latecki and Rolf Lakämper. Shape similarity measure based on correspondence of visual parts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(10):1185–1190, 2000.

[32] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2169–2178, Washington, DC, USA, 2006. IEEE Computer Society.

[33] C.Y. Lin, B.L. Tseng, and J.R. Smith. Ibm mpeg-7 annotation tool version 1.5.1, 2003.
http://www.alphaworks.ibm.com/tech/videoannex [Online; accessed 15-April-2007].

[34] Jiming Liu, Yiu ming Cheung, and Hujun Yin, editors. *Intelligent Data Engineering and Automated Learning, 4th International Conference, IDEAL 2003, Hong Kong, China, March 21-23, 2003, Revised Papers*, volume 2690 of *Lecture Notes in Computer Science*. Springer, 2003.

[35] Stéphane Marchand-Maillet, Eric Bruno, Andreas Nürnberger, and Marcin Detyniecki, editors. *Adaptive Multimedia Retrieval: User, Context, and Feedback, 4th International Workshop, AMR 2006, Geneva, Switzerland, July 27-28, 2006, Revised Selected Papers*, volume 4398 of *Lecture Notes in Computer Science*. Springer, 2007.

[36] Adam Marczyk. Genetic algorithms and evolutionary computation, April 2004.
http://www.talkorigins.org/faqs/genalg/genalg.html [Online; accessed 02-January-2008].

[37] Manuel J. Marin-Jimenez and Nicolas Perez de la Blanca. Empirical study of multi-scale filter banks for object categorization. In *ICPR '06: Proceedings of the 18th International*

Conference on Pattern Recognition, pages 578–581, Washington, DC, USA, 2006. IEEE Computer Society.

[38] José Martínez. Mpeg-7 overview (version 10). Requirements ISO/IEC JTC1 /SC29 /WG11 N6828, International Organisation For Standardisation, Oct 2003. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm/ [Online; accessed 10-April-2007].

[39] Dean S. Messing, Peter van Beek, and James H. Errico. The mpeg-7 colour structure descriptor: image description using colour and local spatial information. In *ICIP (1)*, pages 670–673, 2001.

[40] Moving Picture Experts Group (MPEG). Mpeg-7 frequently asked questions. Technical Report ISO/IEC JTC1/SC29/WG11 N, International Organisation For Standardisation, Mar 2000. http://www.chiariglione.org/mpeg/faq/mp7.htm [Online; accessed 10-April-2007].

[41] Moving Picture Experts Group (MPEG). Mpeg-7 reference software experimentation model, 2003. http://standards.iso.org/ittf/PubliclyAvailableStandards/c035364_ISO_IEC_15938-6(E)_Reference_Software.zip [Online; accessed 01-April-2007].

[42] Moving Picture Experts Group (MPEG). Overview of mpeg-7 part-6 reference sw: Experimentation model (xm). ISG ISO/IEC JTC1/SC29/WG11 N7544, International Organisation For Standardisation, Nice, France, Oct 2005. http://www.chiariglione.org/mpeg/technologies/mp07-rsw/index.htm [Online; accessed 10-April-2007].

[43] Jim Mutch and David G. Lowe. Multiclass object recognition with sparse, localized features. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 11–18, Washington, DC, USA, 2006. IEEE Computer Society.

[44] P. Ndjiki-Nya, J. Restat, T. Meiers, and J. R Ohm. Subjective evaluation of the mpeg-7 retrieval accuracy measure (anmrr). Technical Report M6029, 2000.

[45] Il Seok Oh, Jin-Seon Lee, and Byung Ro Moon. Hybrid genetic algorithms for feature selection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(11):1424–1437, 2004.

[46] J.-R. Ohm, B. Makai, and A. Smolic. Results of VCE-3 on Scalable Color Optimization, ISO/IEC JTC1/SC29/WG11, M6560, La Baule, France, October 2000.

[47] J.-R. Ohm, B. Makai, and A. Smolic. Results of CE CT5 on Scalable Representation of Color Histograms, ISO/IEC JTC1/SC29/WG11, M6285, Beijing, China, July 2000.

[48] Dong Kwon Park, Yoon Seok Jeon, and Chee Sun Won. Efficient use of local edge histogram descriptor. In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia*, pages 51–54, New York, NY, USA, 2000. ACM.

[49] Hang Joon Kim; Eun Yi Kim; Jin Wook Kim; Se Hyun Park. Mrf model based image segmentation using hierarchical distributed genetic algorithm. *Electronics Letters*, 34(25):2394–2395, 10 Dec 1998.

[50] Euripides G. M. Petrakis, Aristeidis Diplaros, and Evangelos Milios. Matching and retrieval of distorted and occluded shapes using dynamic programming. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(11):1501–1516, 2002.

[51] P.Kanungo, P.K.Nanda, A.Ghosh, and U.C.Samal. Classification of objects and background using parallel genetic algorithm based clustering. In *Proceedings of the IEEE-International Conference on Signal and Image Processing.* BCET, Hubli, India, 2006.

[52] P.Kanungo, P.K.Nanda, and U.C.Samal. Image segmentation using thresholding and genetic algorithm. In *Proceedings of the Conference on Soft Computing Technique for Engineering Applications, SCT 2006.* NIT, Rourkela, India, 2006.

[53] I. Emre Sahin. Online text categorization using genetic algorithms. Technical Report BU-CE-0704, Computer Engineering, Bilkent University, Turkey, 2007. http://www.cs.bilkent.edu.tr/tech-reports/2007/BU-CE-0704.pdf.

[54] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[55] Thomas Serre, Lior Wolf, and Tomaso Poggio. Object recognition with features inspired by visual cortex. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 994–1000, Washington, DC, USA, 2005. IEEE Computer Society.

[56] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[57] Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000.

[58] D. Swets and J. Weng. SHOSLIF-O: SHOSLIF for object recognition and image retrieval (phase II), Tech. Rep. CPS 95-39, Michigan State University, Department of Computer Science, A714 Wells Hall, East Lansing, Michigan 48824, October 1995.

[59] Daniel L. Swets, B. Punch, and Juyang Weng. Genetic algorithms for object recognition in a complex scene. In *Proceedings of International Conference on Image Processing, (Washington, D.C.)*, volume 2, pages 595–598, October 1995.

[60] Wen-Bing Tao, Jin-Wen Tian, and Jian Liu. Image segmentation by three-level thresholding based on maximum fuzzy entropy and genetic algorithm. *Pattern Recogn. Lett.*, 24(16):3069–3078, 2003.

[61] Mutlu Uysal and Fatos T. Yarman-Vural. Selection of the best representative feature and membership assignment for content-based fuzzy image database. In Bakker et al. [1], pages 141–151.

[62] Gang Wang, Ye Zhang, and Li Fei-Fei. Using dependent regions for object categorization in a generative framework. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1597–1604, Washington, DC, USA, 2006. IEEE Computer Society.

[63] Wikipedia. Crossover (genetic algorithm) – Wikipedia, the free encyclopedia, 2007. http://en.wikipedia.org/wiki/Crossover_(genetic_algorithm) [Online; accessed 02-January-2008].

[64] Wikipedia. Genetic algorithm – Wikipedia, the free encyclopedia, 2007. http://en.wikipedia.org/wiki/Genetic_algorithms [Online; accessed 02-January-2008].

[65] Wikipedia. Evolution – Wikipedia, the free encyclopedia, 2008. http://en.wikipedia.org/wiki/Evolution [Online; accessed 02-January-2008].

[66] Peng Wu, Yong Man Ro, Chee Sun Won, and Yanglim Choi. Texture descriptors in mpeg-7. In *Computer Analysis of Images and Patterns: 9th International Conference, CAIP 2001 Warsaw, Poland, September 5-7, 2001, Proceedings*, Heidelberg, Berlin, 2001. Springer.

[67] Yakup Yildirim and Adnan Yazici. Ontology-supported video modeling and retrieval. In Marchand-Maillet et al. [35], pages 28–41.

[68] Yakup Yildirim, Turgay Yilmaz, and Adnan Yazici. Ontology-supported object and event extraction with a genetic algorithms approach for object classification. In *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 202–209, New York, NY, USA, 2007. ACM.

[69] Chengqi Zhang, Hans W. Guesgen, and Wai-Kiang Yeap, editors. *PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9-13, 2004, Proceedings*, volume 3157 of *Lecture Notes in Computer Science*. Springer, 2004.

[70] Hao Zhang, Alexander C. Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2126–2136, Washington, DC, USA, 2006. IEEE Computer Society.