DESIGN AND IMPLEMENTATION OF A NOVEL VISUAL ANALYSIS SYSTEM FOR IMAGE CLASSIFICATION

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

BY

ÜMİT LÜTFÜ ALTINTAKAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

OCTOBER 2013

Approval of the thesis:

DESIGN AND IMPLEMENTATION OF A NOVEL VISUAL ANALYSIS SYSTEM FOR IMAGE CLASSIFICATION

Submitted by **ÜMİT LÜTFÜ ALTINTAKAN** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

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

Prof. Dr. Adnan Yazıcı
Head of Department, **Computer Engineering**

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

Examining Committee Members:

Assoc.Prof. Dr. Pınar Karagöz
Computer Engineering Dept., METU

Prof. Dr. Adnan Yazıcı
Computer Engineering Dept., METU

Assoc.Prof. Dr. Murat Koyuncu
Computer Engineering Dept., Atılım University

Assoc.Prof. Dr. Alptekin Temizel
Informatics Institute, METU

Asst.Prof. Dr. Sinan Kalkan
Computer Engineering Dept., METU

Date: 01.10.2013

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, Lastname : Ümit Lütfü ALTINTAKAN

Signature :

ABSTRACT

DESIGN AND IMPLEMENTATION OF A NOVEL VISUAL ANALYSIS SYSTEM FOR IMAGE CLASIFFICATION

Altıntakan, Ümit Lütfü
PhD., Department of Computer Engineering
Supervisor: Prof.Dr. Adnan Yazıcı

October 2013, 117 pages

Possibilities offered by the technology to create, share and disseminate image and video data have resulted in a rapid increase in the available visual data. However, the data is useless unless it is effectively accessed, which necessitates the semantic analysis of visual data. In this dissertation, we present a novel visual analysis system along with its application to image classification problem. We aim to address the challenges in the area originated from the semantic gap, and to facilitate the research efforts in the extraction of high-level semantic information from images. Our system differs from existing works, and contributes to the area in several aspects: A complete visual analysis system in an integrated architecture, a novel fuzzy learning approach in classifier training, a unique feature weighting scheme, a probabilistic classification method, a new high-level classifier fusion, and a new bag-of-words model are some of the key contributions introduced in this dissertation. The experiments conducted on benchmark datasets have shown that our approaches can significantly improve the performance in image classification.

Keywords: Image Classification, Self-Organizing Maps, Fuzzy SVM, Classifier Fusion, Bag-Of-Words.

ÖZ

RESİM SINIFLANDIRMA İÇİN YENİ BİR GÖRSEL ANALİZ SİSTEM TASARIM VE UYGULAMASI

Altıntakan, Ümit Lütfü
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof.Dr. Adnan Yazıcı

Ekim 2013, 117 sayfa

Teknolojinin sunduğu imkânlar sayesinde resim ve video verisi üretimi, paylaşımı ve yayımı mevcut görsel verinin çok hızlı artmasını sağlamıştır. Ancak, mevcut veri, etkin erişim imkânları sunulmaması durumunda kullanılamaz olmaktadır ki, bu da görsel verinin mantıksal analizini zorunlu kılmaktadır. Bu doktora tezinde, yeni bir görsel analiz sistemi ve bunun resim verilerinin sınıflandırılması problemine uygulanması sunulmaktadır. Biz bu çalışma ile araştırma alanında mantıksal boşluktan kaynaklanan problemlere çözüm bulmak ve resimlerden yüksek-düzeyde anlamsal bilgi çıkarım çalışmalarına katkı sağlamayı amaçlamaktayız. Bizim sistemimiz mevcut çalışmalardan farklı olup, araştırma alanına pek çok açıdan katkı sağlamaktadır: Entegre edilmiş yeni bir görsel analiz sistemi, sınıflandırıcı öğretiminde yeni bir bulanık öğrenme metodu, yeni bir üst düzey füzyon ve yeni bir BOW modeli, bu tez ile ortaya konan katkılardan bazılarıdır. Ortak veri setleri üzerinde yapılan testler, bizim yaklaşımlarımızın resim sınıflandırmasında önemli performans artışı sağladığını göstermektedir.

Anahtar Kelimeler: Resim Sınıflandırma, Öz-Düzenleyici Haritalar, Bulanık Vektör Destek Makinesi, Sınıflandırıcı Füzyonu, Kelimeler Kümesi.

To my family

ACKNOWLEDGMENTS

I would like to express my inmost gratitude to my advisor, Professor Adnan Yazıcı, for his guidance, advice, encouragements and support throughout this PhD. study. I have learned greatly from him on how to be an excellent researcher, and without his supervision, I would not be able to complete this study.

I am very thankful to the members of my PhD. thesis committee, Dr. Pınar Karagöz and Dr. Murat Koyuncu, for their support and useful feedbacks throughout the work.

I would also like to thank all the members of Multimedia Research Group at CENG/METU for their technical guidance and support during the study.

Finally, I would like to thank to my family, especially to my mother and wife for all their patience, support and always encouraging me.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vi

ACKNOWLEDGMENTS viii

TABLE OF CONTENTS ix

LIST OF TABLES xii

LIST OF FIGURES xiii

CHAPTERS

1. INTRODUCTION 1
    - 1.1 Motivation 1
    - 1.2 Objective 1
    - 1.3 Challenge 2
    - 1.4 Contributions 2
    - 1.5 Publications 3
    - 1.6 Thesis Outline 4
2. BACKGROUND INFORMATION AND RELATED WORK 5
    - 2.1 Background Information 6
        - 2.1.1 MPEG-7 Overview 6
        - 2.1.2 Data Sets 8
        - 2.1.3 Evaluation Metrics in Image Classifications 9
    - 2.2 Related Work 11
        - 2.2.1 Learning Methods in Visual Analysis 11
        - 2.2.2 Existing Visual Analysis Systems 13
        - 2.2.3 Fusion Methods 14
        - 2.2.4 Summary 16
3. VISUAL ANALYSIS SYSTEM 17
    - 3.1 The System Architecture 17
    - 3.2 The Low-level Processing Layer 22
        - 3.2.1 Segmentation 22
        - 3.2.2 Annotation 23
        - 3.2.3 Feature Extraction 25
    - 3.3 The Feature Modeling Layer 26

3.3.1 Self-Organizing Maps 26

3.3.2 The construction and utilization of SOMs in visual analysis 27

4. LEARNING IN VISUAL ANALYSIS 31

4.1 SVM Theory 32

4.1.1 Optimal Hyperplane 33

4.1.2 Soft Margin SVM 36

4.1.3 The Kernel Trick 37

4.2 Fuzzy SVM 39

4.2.1 Existing Membership Functions 40

4.3 A Novel Membership Function 41

4.3.1 Membership Calculation for Low-level Visual Features 42

4.3.2 Membership Determination Strategy 43

4.4 The Construction and Utilization of Binary Classifiers 46

5. HIGH-LEVEL CLASSIFIER FUSION 49

5.1 Dempster-Shafer Theory 50

5.2 Probability Estimates in SVM Classification 52

5.3 Determination of Feature Weights 55

5.3.1 SOM Mapping and 2-d Clustering 56

5.3.2 Entropy Calculation and Weight Formulation 61

5.4 Application of the DS Combination in Classifier Fusion 62

6. AN EFFECTIVE BOW MODEL IN IMAGE CLASSIFICATION 65

6.1 Background Information 66

6.1.1 Scale-Invariant Feature Transform 66

6.1.2 The Classification Methods used in BOW 67

6.1.3 Related Work 69

6.2 System Architecture 70

6.2.1 Codebook Generation and Feature Mapping 71

6.2.2 Our BOW Approach for Image Classification 72

6.2.3 Extraction and Utilization of Distinctive Features in Image Classification 73

6.3 The Effect of Threshold in Distinctive Feature Detection 75

6.4 Codebooks Generated by SOM and k-Means 76

6.4.1 Average quantization errors of codebooks 76

6.4.2 Effect of codebook size in classifications: 76

7. EXPERIMENTS 79
7.1 The Training and Test Steps Applied in Image Classification............................... 80
7.1.1 Training Design in Image Classification 80
7.1.2 Test Design in Image Classification 82
7.2 Dataset and Evaluation Metric 84
7.2.1 Dataset 84
7.2.2 Evaluation Metric... 85
7.3 Extraction and Preparation of Low-level Features 85
7.3.1 The MPEG-7 Descriptors 86
7.3.2 Data Preparation... 86
7.4 Classification Results of Single Classifiers... 90
7.5 Combined Classification Results 95
7.6 Classification Results in BOW Model 98
7.6.1 Experimental Results and Performance Evaluation in BOW 99
7.7 Overall Evaluation and Comparison with Previous Works.................................. 100
8. CONCLUSION AND FUTURE WORK 103
REFERENCES 105
VITA 117

LIST OF TABLES

TABLES

Table 2.1: Visual features and corresponding descriptors... 7
Table 3.1: The MPEG-7 features and their usability.. 25
Table 4.1: The SOM distances and membership values for sky concept.............................. 45
Table 5.1: The classification results and probability estimates... 54
Table 5.2: Single classification decisions for a test image and feature weights.................... 62
Table 6.1: The Effect of Threshold in Classification (TP and FP rates for *car* class)........... 75
Table 6.2: Average quantization Errors in k-Means and SOM Clusters................................ 76
Table 7.1: The classes and number of images used in the experiments................................ 85
Table 7.2: Data preparation for SVM training in learning the Sky concept.......................... 89
Table 7.3: Data preparation for SOM construction 89
Table 7.4: The utilization of regions in membership calculation and entropy analysis for *sky* class. 89
Table 7.5: Individual Classification Results (SVM). 90
Table 7.6: Individual Classification Results (Fuzzy SVM)... 90
Table 7.7: Comparison of single classifiers in SVM and fuzzy SVM. 91
Table 7.8: Comparison of methods on VOC 2006 dataset using MAP................................. 92
Table 7.9: MAP scores of best single classifier and fusion in classical SVM. 96
Table 7.10: MAP scores of best single classifier and fusion in fuzzy SVM. 96
Table 7.11: Effect of feature weights on combined results.. 97
Table 7.12: Effect of fuzzy learning on combined results (W-DS)....................................... 98
Table 7.13: The classification results obtained by BOW model (AP). 100
Table 7.14: The overall classification results (AP). 101
Table 7.15: Comparison with previous works (AP).. 102

LIST OF FIGURES

FIGURES

Figure 2.1 The MPEG-7 class hierarchy. 6

Figure 2.2 Learning training data. 10

Figure 2.3 Classifier confusion matrix. 10

Figure 3.1 General architecture of the visual analysis system. 18

Figure 3.2 Functional view of the classification system. 19

Figure 3.3 Low-level processes performed in the system. 22

Figure 3.4 A sample image and regions after JSEG segmentation. 23

Figure 3.5 A sample run of the annotation tool. 24

Figure 3.6 A Sample SOM Network. 27

Figure 3.7 Construction of SOM networks. 28

Figure 3.8 SOM training for a sample class. 29

Figure 3.9 The distributions of features on SOM space with bmu-hits. 29

Figure 4.1 Classifier learning process. 31

Figure 4.2 Separating hyperlanes in a 2-d space. 33

Figure 4.3 The hyper-plane maximizing the margin in a two-dimensional space. 34

Figure 4.4 A soft-margin SVM. 36

Figure 4.5. The non-linear separable data in input space and mapping to feature space. 37

Figure 4.6 A Sample image with annotations. 45

Figure 5.1 The fusion process in visual analysis. 50

Figure 5.2 SVM classification error rate vs. distance to hyperplane. 53

Figure 5.3 A sample image and region annotations. 54

Figure 5.4 The feature weighting process. 56

Figure 5.5 A sample SOM of size 7x8. 57

Figure 5.6 The bmu hits on the SOM. 58

Figure 5.7 The bmu hits after normalization 58

Figure 5.8 The aggregated *bmu* hits 59

Figure 5.9 The first cluster and updated hits 60

Figure 5.10 Final clusters. 60

Figure 6.1 A sample image and SIFT keypoints. 66

Figure 6.2 The training steps of the classification system. 70

Figure 6.3 The test steps of the classification system. 71

Figure 6.4 The codebook generation and a sample SOM. 72

Figure 6.5 The distinctive keypoints of Figure.6.1. 74

Figure 6.6 The effect of codebooks generated by k-Means and SOM in SVM classification. 77

Figure 7.1 The training phase of our classification system. 81

Figure 7.2 The test phase of our classification system. 83

Figure 7.3 A sample MPEG-7 file in xml format. 87

Figure 7.4 Sample images with annotations. 88

Figure 7.5 Effects of DS-fusion in classical SVM (MAP). 92

Figure 7.6 Effects of DS-fusion in fuzzy SVM (MAP). 93

Figure 7.7 Effects of DS-fusion in classical SVM (F-measure). 93

Figure 7.8 Effects of DS-fusion in fuzzy SVM (F-measure). 93

Figure 7.9 F-measure results of SVM and fuzzy SVM (*car* class). 94

Figure 7.10 F-measure results of SVM and fuzzy SVM (*cow* class). 94

Figure 7.11 F-measure results of SVM and fuzzy SVM (*horse* class). 95

Figure 7.12 F-measure results of SVM and fuzzy SVM (*motorbike* class). 95

CHAPTER 1

INTRODUCTION

1.1 Motivation

Possibilities offered by the technology to create, share and disseminate image and video data have resulted in a rapid increase in the available multimedia collections. For example, the Flickr system is reported to host more than 6 billion images as of August 2011 [1], and more than 200.000 videos are uploaded to YouTube per day [2]. However, the huge amount of available multimedia data is useless unless it is effectively accessed, which necessitates semantic analysis of the visual data.

Some of the potential multimedia applications that would be enabled by the semantic analysis of visual data can be as follows:

- *Surveillance and security systems*: For example, the determination of unattended luggage in public areas, such as airports, or identifying abnormal events in surveillance videos requires semantic-enabled multimedia applications.
- *Multimedia Search and Retrieval*: Querying the massive amounts of multimedia requires accessing the results by issuing semantic keywords or high-level concepts. A sample query, for instance '*retrieve the videos of car accidents from yesterday's news videos*', can only be accessible if the available multimedia data is indexed at semantic level.
- *Medical image analysis.*
- *Object detection:* Along with the presence or absence of a particular object, detecting object boundaries is also important, and has many potential applications, such as motion tracking and event detection in videos.
- *Image Annotation:* Automatic annotation of images is a critical process to effectively retrieve and filter the visual content available in internet or personal libraries, which can only be met by providing extracting high-level semantics in images.

1.2 Objective

The aim of this thesis is to design and develop a visual analysis system for the extraction of high-level semantic information from visual data, and to contribute the research in image analysis by addressing major problems in the field. In this thesis, we are particularly interested in image classification problem, which is also known as high-level feature extraction or image indexing in the literature.

We approach the image classification as a supervised learning problem: given a set of training image along with semantic labels associated to them, we construct binary image classifiers using the low-level features extracted from the training data, and utilize these classifiers to predict the presence or absence of a learned class in test images. We aim to contribute in developing reliable and effective multimedia applications by introducing some theoretical and algorithmic basis for extracting semantic information from visual data.

1.3 Challenge

The greatest challenge in the research area is the *semantic gap* [3, 4], which can be described as 'the lack of coincidence between the information that one can extract from the digital data and the interpretation that same data has for a user in a given situation' [5]. The gap basically emphasizes the difficulty in obtaining the high-level semantic information by using the automatically extracted low-level visual features in multimedia domain.

Different lighting conditions, views and positions in image and video files along with geometrical properties are some of the challenges in the research field. Moreover, starting from the extraction of low-level visual features, the analysis visual data requires computationally intensive processes, which makes is also a problem in developing effective multimedia applications for most of the real-world problems.

1.4 Contributions

We consider the followings are some of the key contributions in this dissertation.

- *An Effective Visual Classification Approach*: We have designed a complete image analysis system within an integrated framework by introducing some novel methods in different tasks such as low-level processing, feature weighting, classifier learning, and so on. The system differs from the previous works in many ways, and contributes to the area in obtaining the high-level semantic information in visual data.
- *Fuzzy Learning in SVM*: In order to increase the learning capacity of classifiers, we perform a fuzzy training approach in the classifier learning phase. More specifically, we reduce the effect of noisy regions in images by assigning some membership degrees prior to the classifier learning in training process.
- *A New Membership Calculation Method for High-Dimensional Spaces:* The existing membership calculation methods are inapplicable to the complex visual features, which are ranging from tens to several hundreds in dimensions. We introduce a new method in evaluating the membership degrees of low-level features by using neural network approaches to map the high-dimensional visual features on 2-D output spaces.
- *Low-level Feature Modeling*: We build a number of semantic models utilizing the feature vectors of training data, and exploit them in several computations, which is a key factor in developing our visual analysis system. The computation of membership degrees in classifier training, the calculation of feature weights, and the generation of codebooks are some of the examples of them.

- *Low-level Feature Weighting:* We present a new approach in feature weighting through applying information entropy measures, which are utilized as the trust level of the classifiers during the high-level classifier fusion process.
- *Probabilistic Classifications*: We perform a probabilistic classification model in the learning phase, which enables not only the binary classification results but also the strength of the classification. The probabilistic classification results are utilized as the confidence degrees, and used in the combination of single classifiers.
- *High-level Classifier Fusion*: The outputs generated by individual classifiers are combined by applying a novel fusion method based on the Dempster Shafer combination rule [6]. This process exploits the classifiers as main information sources, and utilizes three different parameters to obtain an optimal combination: 1) the binary classification decisions, 2) the confidence degrees in classifications, and 3) the feature weights.
- *A New Codebook Generation Method In BOW*: The codebook generation is a key factor in developing effective visual classification systems that perform the Bag-of-Words (BOW) model [7]. We introduce a new method to generate the visual-words in the codebook generation phase, which produces better class vocabularies than classical approaches.
- *Utilization of Distinctive Local Features in BOW*: We also work on the utilization of scale-invariant local features [8] throughout the dissertation, and present a new method to determine the distinctive SIFT features in BOW model. This approach provides an increase in the image classification performance, and has the potential to develop efficient real-world applications by reducing the total number of local features.

1.5 Publications

The fuzzy learning approach presented in Chapter 4, and its application to image classification was presented in IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE 2012). A journal version of the approach along with its effects on the classifier fusion described in Chapter 5 was sent to the Imaging Science Journal (SCI Journal) on April 2012, and still under review.

The application of the DS theory to classifier combination, and the utilization of the probabilities in SVM classification (Chapter 5) is submitted to the 2nd Int. Conf. on Computing and Comp. Vis. (ICCCV 2013), and accepted. Finally, the use of distinctive SIFT features along with a new codebook generation method in BOW presented in Chapter 6 is sent to IEEE Trans. on Multimedia on September 2013, and under review.

1.6 Thesis Outline

The rest of the dissertation is organized as follows:

- Chapter 2 introduces the background information and related work in the literature by reviewing some of the major works done in image analysis.
- Chapter 3 presents the system architecture of our classification system: major tasks performed in the system from a functional point of view, low-level feature processes and feature modeling are explained, respectively.
- Chapter 4 focuses mainly on the classifier learning phase of the system. First, we summarize the SVM theory briefly, and then present the fuzzy extension in SVM learning. Next, some of the previous approaches for membership calculations are introduced. Finally, our membership calculation method along with its application to image classification is described in this chapter.
- Chapter 5 presents the classifier fusion process, in which we utilize the Dempster-Shafer (DS) Evidence Theory [9] in combining the classifier results. The chapter presents the basics of the DS theory, the feature weighting process, and the probabilistic classification method. Finally, the chapter concludes with the application of the DS combination rule to our classification system.
- Chapter 6 presents the use of SOM neural networks in the generation of codebooks in BOW model. We investigate the SIFT method separately in this chapter, and employ different classification techniques in BOW including Naïve Bayesian, k-NN and linear SVM classifications. We also introduce the determination of the distinctive SIFT points in this chapter.
- Chapter 7 presents the experiments conducted on benchmark datasets comprehensively. We analyze the effect of the methods introduced in this dissertation by performing a number of experiments. We examine the followings in this chapter: 1) the effect of the fuzzy learning approach in SVM, 2) different fusion methods including the DS combination, along with the effect of feature weighting, 3) the analysis of the BOW model on different classifications methods. The chapter concludes with the overall evaluation of the methods introduce in this dissertation, and compares the classification results with the state-of-art works using the same datasets.
- Chapter 8 concludes the dissertation, and presents the future work.

CHAPTER 2

BACKGROUND INFORMATION AND RELATED WORK

The importance of high-level feature extraction from visual data has drawn much attention from the research community over the years. A fundamental requirement in building effective multimedia systems for various application domains and different purposes requires extraction of high-level semantic information from visual data. This requirement can only be met by providing semantic-based models that produce high-level interpretations using a number of low-level attributes extracted from multimedia content [10].

The main focus in early approaches is basically on the content-based image retrieval (CBIR) [11], in which a user employs a sample query by entering a query image or a part of it. The general approach in CBIR systems is to access the visual data primarily using the low-level attributes extracted from the input query, and to find the similar images from the available multimedia collection that best match the extracted features. Some CBIR systems also support text-based or primitive feature-based (color, shape etc.) queries, but they provide limited or unsatisfactory results since the computations are solely performed using the signal-level attributes, which do not encapsulate high-level semantic interpretations [12, 13].

Instead of accessing the visual content using low-level attributes, one often requires more effective methods that enable high-level query searching capabilities in multimedia applications. To illustrate, for example, a semantic-enabled query can be: "*find the scenes of sport videos that include scores*". These queries often require significant amount of reasoning as well as extracting high-level semantic information included in the visual data. However, the state-of-art methods in image processing provide limited functionalities that are mainly domain-dependent, which are far from realizing the access to visual data at semantic levels. The major challenge in the area, as stated earlier, is the semantic gap between the low-level attributes that can be extracted automatically from images, and their high-level interpretations in different context [3].

Recent approaches in image analysis have been shifted from content-based methods to semantic-enabled systems, in which complex classification schemes along with various system architectures are introduced [14]. For example, different video indexing systems is presented for automatic indexing of videos in [15]. The authors mainly focus on three problems in the indexing problem: 1) *what* to index in terms of granularity, 2) *how* to index in terms of the modalities, and 3) *which* index in terms of the type of the index used for labeling. The proposed solutions to these problems are: 1) indexing can be applied to the entire document or to single frames in video files, 2) the different modalities of video can be considered, such as visual, auditory or textual, and 3) the index terms is defined by using some high-level abstractions in video, such as the names of the players or time dependent positions in soccer video. A probabilistic framework for semantic indexing video shots is presented in [16], which aims to the low-level media features to some high-level labels by relating the semantic concepts and their co-occurrences along with some temporal dependencies occur in semantic concepts.

In this chapter, we first provide some background information related for the implementation of the visual analysis system, and then present a comprehensive review of the previous works in image analysis literature.

2.1 Background Information

2.1.1 MPEG-7 Overview

MPEG, formally named *Multimedia Content Description Interface*, is developed by the Moving Pictures Expert Group (MPEG), a working group of ISO/IEC [17]. Unlike the preceding MPEG standards, the MPEG-7 aims to provide a rich set of standardized tools to describe the multimedia content [18].

The MPEG-7 standard provides a set of description tools, which consist of descriptors, description schemes, and a description definition language along with some system tools to operate on these descriptions. The *descriptors* represent the features or attributes in multimedia data such as the color, texture, textual annotation and media formats. The *description schemes* specify the structure and the relationships between the components, which can be either descriptors or other description schemes. The *description definition language* is an XML-based schema to define or extend the available descriptors and description schemes. The top-level class hierarchy of the MPEG-7 standard used to define multimedia content is shown in Figure 2.1.

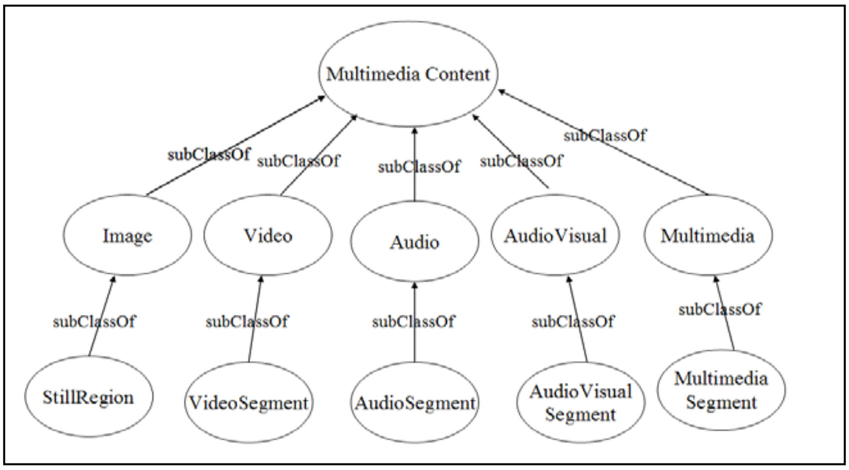


Figure 2.1 The MPEG-7 class hierarchy.

There are four different feature types in the visual part of the MPEG-7 standard, which are *color*, *texture*, *motion*, and *shape* based. Table 2.1 shows the MPEG-7 visual descriptors and the corresponding feature types accordingly.

The definitions of the features used are as follows:

- *Dominant Color Descriptor (DC):* The DC feature represents the most representative colors in an image or image region. This descriptor includes information about the representative colors, their percentages, spatial coherency of the color, and color variance.

Table 2.1: Visual features and corresponding descriptors

Type	Feature	Descriptor
Visual	Color	Dominant Color
		Scalable Color
		Color Layout
		Color Structure
		GoFGoPColor
	Texture	Homogeneous Texture
		Texture Browsing
		Edge Histogram
	Shape	Region Shape
		Contour Shape
		Shape 3D
	Motion	Camera Motion
		Motion Trajectory
		Parametric Motion
		Motion Activity

- *Color Layout Descriptor (CL):*The CL feature characterizes the spatial distribution of colors within an image [19]. The color information of image is divided into 8x8 blocks, and in each block the dominant colors are found using the YCbCr color system. Then, a discrete cosine transform is applied on the dominant color of each channel, where the coefficients of the transform are used as descriptors.
- *Color Structure Descriptor (CST)*: The CST feature is a generalization of the color histogram that captures the spatial characteristics of the color distribution in an image. A structuring element of rectangular shape (8x8) is slided over the image, and the number of positions that an element contains each particular color is recorded, and used as the descriptor.

- *Scalable Color Descriptor (SC):* The SC feature is a color histogram in hue-saturation-value (HSV) color space. In order to reduce the number of bins in the original histogram, a Haar transform encoding is used. The possible descriptor sizes are 16, 32, 64, or 128 bins [19, 20].
- *Edge Histogram Descriptor (EH)*: The EH feature represents the spatial distribution of five edges in image: four directional edges, and one non-directional edge. The amount of the vertical, horizontal, 45 degree, 135 degree and non-directional edges is calculated in 16 sub-images in an image, and at the end a 80-dim vector is generated [20].
- *Homogeneous Texture Descriptor (HT)*: The HT feature is extracted by applying a Gabor filter [20] in 6 frequencies in 5 different orientation channels, and represents a region's texture in terms of local spatial frequency statistics. At the end, the energy and energy deviation in each channel results in a 62-dim feature vector for the descriptor representation.
- *Region Shape Descriptor (RS)*: The RS feature is defined by the Angular Radial Transform [20], which captures the distribution of all pixels within an image region through decomposing the shape into a number of orthogonal 2-d basis functions.
- *Contour Shape Descriptor (CS)*: The CS feature uses Curvature Scale-Space representation, and mainly represents the characteristic shapes of an object or image region based on its contour [20].

In the implementation of our visual analysis system, we mainly work on 2-D colored images, and hence we use a subset of the descriptors shown in Table 2.1. The detailed information about the extracted MPEG-7 descriptors along with their utilization in the scope of the dissertation is given in Chapter 3 and Chapter 7, respectively.

2.1.2 Data Sets

Having a common multimedia dataset is a basic requirement to facilitate the evaluations in semantic multimedia researches since mainly for determining a common platform in evaluations. In order to compare the proposed methods, a number of datasets for different application domain and purposes are provided in image analysis. These datasets are usually divided into training and test sets, and used in semantic modeling and evaluation tasks respectively. This section briefly introduces some of the existing multimedia datasets along with the main usage areas.

The TREC Video Retrieval Evaluation (TRECVID) is organized by the National Institute of Standards and Technology, and is a major video retrieval benchmarking platform for the semantic analysis of video data [21]. The data is ranging from documentaries, films, educational material to multi-lingual broadcast news. Semantic annotation of the data is not provided by the organizers, but some researchers create data annotations and these are distributed amongst the participants. The dataset is available only to participants and renewed each year. The competitions may be changed in years.

One common benchmark dataset, which is also publicly available, is PASCAL Visual Object Classes (VOC) challenge. It has been organized annually from 2005 to present for visual object classification and detection tasks. The dataset provides a standard set of images along with standard annotations. The two major tasks in the VOC challenge are: 1) *classification*, which is the process of predicting the presence or absence of an object class in test images, and 2) *detection*, which aims to get the bounding boxes of each object in images [22].

The Caltech 101 and Caltech 256 datasets contain images of 101 and 256 image categories, respectively. They are generated primarily for the object recognition evaluations, in which only a single object is included in each image in the datasets. Since they include relatively small number of training images compared to other collections, and the image collections are not natural, these datasets are not ideal for making comparisons, in general.

Another video dataset framework to automatically detect 101 semantic concepts in video is presented in [10], which contains a multimedia archive of 85 hours of broadcast news. The aim in this work is to organize for evaluating the performances of different approaches in semantic video analysis and to give insight the intermediate steps in video indexing by providing a common framework for the evaluation and repeatability of the experiments. The authors have manually labeled the ground truths at shot level using over hundred semantic concepts, and also provide the archive with a baseline implementation methods including a visual-only, textual-only, early fusion, late fusion and combined analysis in the framework.

The LabelMe dataset [23] at MIT is similar to VOC challenge, and contains general photographs that contain multiple objects. The dataset has been formed by providing a web-based annotation interface, and encouraging the users to contribute and share their annotations. For the most part the dataset is incompletely labeled, and the users are free to choose which objects to annotate. Hence, dataset is unsuitable for strict testing and comparison.

A recently produced data set is the Lotus Hill collection [24], which contains annotations in a hierarchical decomposition of individual objects. But, only a limited number of images are available to researches and not too much work has been reported on it yet.

2.1.3 Evaluation Metrics in Image Classifications

Classifiers are trained on finite training datasets, and have to be tested against a different set of data for performance evaluations. One of the important properties in classifier learning is the generalization capacity, since learning the training data too precisely, i.e. over-fitting, results in problems during the classification of new data. Different data learning schemes in classifier training is shown in Figure 2.2.

There are mainly two general terms that are used to evaluate the classifier performances: accuracy, which is the percentage of correct classifications, and error rate, which is the percentage of misclassifications.

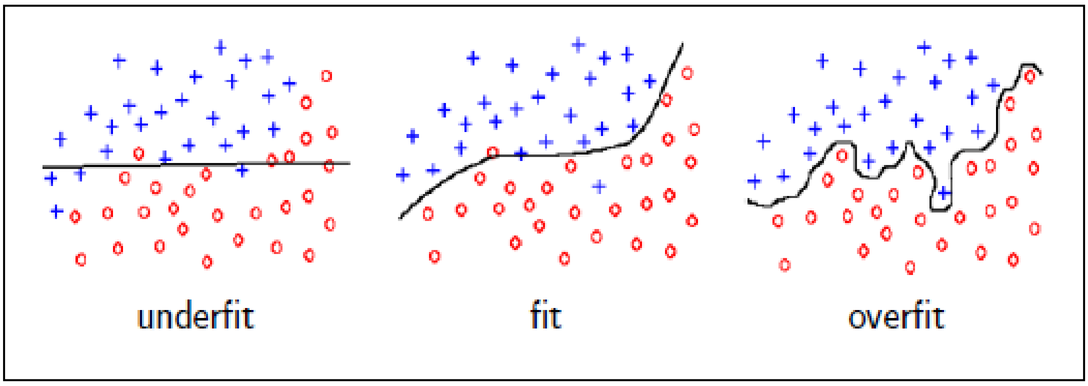
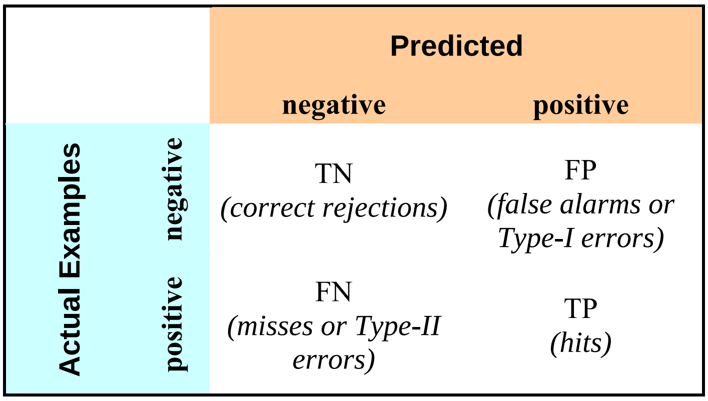


Figure 2.2 Learning training data.

However, these two terms are not useful in many classification problems since they give equal costs to correct and incorrect classification results, which is not realistic for most of the real-world problems.

Figure 2.3 depicts possible classification results in a classification problem [25]:

- *True Negative* (TN): The samples that are correctly evaluated as negative class,
- *True Positive* (TP): The samples that are correctly evaluated as positive class,
- *False Negative* (FN): The samples that are incorrectly evaluated as negative class, also alled as *misses* or *Type-II* errors.
- *False Positive* (FP): The samples that are incorrectly evaluated as positive class, also called as *false alarms* or *Type-I* errors.



		Predicted	
		negative	**positive**
Actual Examples	**negative**	TN *(correct rejections)*	FP *(false alarms or Type-I errors)*
	positive	FN *(misses or Type-II errors)*	TP *(hits)*

Figure 2.3 Classifier confusion matrix.

Using the-above mentioned, the following metrics can be used to evaluate the classifiers performances in the literature [25]:

- $Accuracy = \dfrac{TN + TP}{All}$
- $True\ positive\ rate,\ Recall,\ Sensitivity = \dfrac{TP}{FN + TP}$
- $Precision,\ Predicted\ positive\ value = \dfrac{TP}{FP + TP}$

As mentioned previously, these metrics also give equal costs to correct and incorrect predictions in evaluations, which is not realistic in real-world applications. For instance, the cost of missing an air missile (FN) in case of an actual attack is much higher than the cost of false alarm (FP).

Another problem that needs to be considered in evaluating classifiers is the rate of positive and negative samples in training set, which is also called as the unbalanced data problem. For instance, in medical diagnosis, the rate of healthy cases is much higher than the patients that carry disease. In this kind of classifications, a classifier that predicts the healthy patients with 99% accuracy can be useless, since is more significant to detect the unhealthy patients.

In order to eliminate the above-mentioned two performance evaluation problems in image analysis, another classifier evaluation metric called Receiver Operating Characteristic (ROC) is defined in the literature [26]. The ROC curve characterizes the degree of overlap of classes for a single feature and the comparison of classifiers is based on different thresholds or operating points.

The Average Precision (AP) is another classifier evaluation metric, which is is defined as the mean precision at a set of equally spaced recall levels [27]. The AP summarizes the shape of the precision/recall curve; more information about the AP is given during the comparison of classifiers in Chapter 7.

2.2 Related Work

In this section, we review some of the existing works and state-of-the-art approaches in image analysis. In order to clarify the major works done and present a better understanding of the existing methods, we review the related work into three different titles as follows: 1) the learning methods and applications in visual analysis, 2) existing frameworks in visual analysis, and 3) the fusion techniques in the area, respectively.

2.2.1 Learning Methods in Visual Analysis

In the literature, a number of machine learning techniques have been used to build high-level classifiers in the scope of visual analysis [28]. The methods can be grouped into two major categories according to the learning strategies used:

- *Discriminative methods*: The most predominant learning technique used in this category is Support Vector Machines (SVMs) [29]. The SVM is applied widely in different classification problems including image analysis has the high generalization ability along with a better performance in pattern recognition problems, and also applied extensively in image analysis [28] . Apart from the SVM; k-NN classification [30], neural networks [31, 32], decision trees [32, 33] are frequently applied methods in discriminative learning.

- *Generative models:* This approach attempts to model the probability distributions mainly using Bayesian inference. The training data is used to learn the estimates of probability distributions, and unseen data is classified according to the calculated probability distributions applying Bayes rule [34]. Some popular methods in this category are Gaussian [35], Naïve Bayes [36], Mixtures of Experts [37], and Hidden Markov Models [38].

Different classification methods, including SVM, multi-layer perceptron network, the KNN classifier, a neural network, and a fuzzy neural network in image retrieval is analyzed in [26]. Each method's performance is analyzed using precision-recall scores, and at the end, the fuzzy neural network and SVM are shown to produce better classification results against other methods. In another work, optimization of SVMs in order to enable an improved video retrieval system is presented in [39], in which several SVM models using different learning parameters, such as feature selection, instance selection, and kernel parameter settings are investigated by performing Genetic Algorithm on classifier outputs.

The radial basis function networks, SVMs, naive Bayes, and decision trees are compared on three different image collections: medical, texture and natural images in [69]. The authors investigated the classification performances of the learning methods using four semantic classes for retrieving the visual data using the content information. The first three methods produce similar performance results with different computational complexities, and the last one output the poorest results in classification tasks.

A multi-layered SOM learning system is presented using a content based image retrieval approach designed for image browsing in [74]. The authors implement a two-layered SOM network design in order to reduce the final learning error in SOM. The first layer uses the color attributes in HSV color space, and the second layer uses the weights of first layer in which a re-learning is performed to reduce the learning error in image browsing applications.

The utilization of fuzzy rules is rarely applied in semantic analysis of visual data in the literature. The learning of fuzzy rules based on SVM is proposed in [56, 57], in which the authors perform SVM learning along with a fuzzy algorithm to extract linguistic fuzzy rules using MPEG-7 color and texture descriptors in beach/urban scene classification.

A fuzzy learning system by utilizing fuzzy clustering on SVM results for improving the generalization performance in human skin color segmentation is presented in [58]. The fuzzy if-then rules are constructed by using fuzzy singletons, and applied for skin color segmentation problem in the paper. The scaled hue and saturation values are exploited in classifier training, and the approach is shown to produce better results in image segmentation problems.

A fuzzy support vector machine (FSVM) is introduced in [53], which defines a membership calculation method to classify the images that cannot be classified correctly using typical SVM learning technique. The fuzzy machine produces the same results as the normal machine in the classifiable regions, however for the incorrectly classified regions; the fuzzy SVM performs better results than the classical method. An application of the proposed method is presented in [54], in which the beast cancer detection is investigated in medical image analysis using both SVM and fuzzy SVMs.

Another classification approach utilizing fuzzy SVM is applied to high-resolution remote sensing images in [55]. The authors propose a fuzzy membership method using the scale of a sample and the distance of a training sample to its class center along with their relative positions. The fuzzy method reduces the impact of the non-critical image regions in learning the classifiers in training data.

2.2.2 Existing Visual Analysis Systems

Representation of images at region level is a widely used approach in semantic analysis of visual data since this method is close to human perception [45]. The main idea behind the region-based approach is to work on the important regions instead of processing the entire image. First, the salient areas are determined through applying an initial pre-processing step, and then the following computations are performed using only the regions that include important objects. In this section, we present some of the existing visual analysis systems and major works in image analysis.

An automatic image annotation system for CBIR applications is introduced in [31], in which the images are represented as sequences of feature vectors applying a HMM approach. Another region-based classification system, which performs a maximum probability search, is presented in [32]. The work is performed by training a set of semantic classes using HMM on the low-level features extracted from TRECVID archive.

Automatic image annotation is proposed as a solution for semantic image retrieval by some works [35-36]. A multi-layer system for annotating natural scene images is presented in [35] by utilizing the salient objects in images. The SVM is employed for learning the semantic concepts, and for finding the optimal parameters in SVM an expectation maximization method is used in the proposed system. Another work for automatic annotation of images is proposed by [36], in which both content- and concept-level annotations of natural scenes by using the salient objects and the relevant semantic labels. Given an input image, first the salient objects such as "*rock*" or "*sky*" are detected, and then a semantic concept modeling is performed to get the overall annotations related to the image scene.

A hybrid learning system for image classification by utilizing decision trees along with association rule mining is presented in [33]. In the paper, first a virtual semantic ontology is constructed to map the objects into semantic classes, and then each image is divided into several sub-images by extracting the objects inside images. The virtual code of each object is used for association mining during the evaluation phase in the proposed system architecture.

A hierarchical region-based image retrieval approach is presented by Sun and Ozawa in [40] based on wavelet transform. The retrieval starts with an initial segmentation of images in the low frequency band, and then the boundaries between the segmented regions are used to improve the region-based retrieval of images. The proposed method also presents a tradeoff between the retrieval effectiveness and the efficiency in image retrieval problems.

Semantic analysis of images through a knowledge-assisted approach utilizing different learning techniques are presented in [41]. The following four learning techniques are investigated in the paper: SVMs, Self-Organizing Maps, Genetic Algorithm and Particle Swarm Optimization [42]. The low-level descriptors extracted from the initially segmented image regions are utilized in classifier training in each method, and then the classifiers are applied to evaluate the test regions with a predefined semantic label. An evaluation framework is developed in order to compare the classifier performances in each technique performing several experiments along with combined use of models such as GA with PSO, or SVM with SOM. The results show that the individual classification performance can be increased by using combined classification schemes.

There are also some works on utilizing the *contextual knowledge* such as the size, position or relative locations of objects in an image. The context can also help to achieve reasoning on unseen data, and improve the semantic extraction process in visual analysis. For example, an approach to also include the contextual features during the image annotation is presented in [43], and each pixel in an image is assigned to one of a finite set of labels by using multi-scale conditional random fields.

In the literature, there are also some researches that make use of the semantic concept hierarchies in terms of ontologies in order to improve the image annotation task. The use of semantic ontologies in describing visual content helps in providing well-structured information, and improves the retrieval accuracy in general. A review of the recent works towards the use of semantic hierarchies and ontologies in the field of image analysis is presented in [44].

A survey related to the semantic-based image retrieval is presented by Liu et al. in [45]. The authors categorize the efforts in narrowing the semantic gap problem into five major areas: 1) the use of object ontologies in terms of high-level class definitions, 2) the use of machine learning methods to associate the low-level features by query concepts, 3) the use of relevance feedback to obtain user intention, 4) the generation of semantic templates for high-level image retrieval, and 5) the combination of evidences related to the visual data for internet-based image retrieval in web.

2.2.3 Fusion Methods

The fusion process can be applied at different levels in image classification: One general approach is to perform the fusion on low-level features, which is known as the feature fusion or early fusion approach. Another fusion strategy is classifier fusion or late fusion, which takes place on classifier outputs. In this section, we provide some basic works regarding to the combination of semantic classifiers in the scope of visual analysis.

An early fusion approach is employed utilizing different learning techniques to combine the MPEG-7 descriptors in a content based image retrieval system [46]. First, an SVM classifier is trained on a feature vector that is formed by merging the individual MPEG-7 descriptors. Secondly, a back-propagation feed-forward neural network is trained on the low-level features of two images, and then a k-nearest neighbor classifier is used to evaluate the fusion. Finally, a fuzzy neural network is applied performing fuzzy rules. The experimental results show that, the back-propagation fusion has the best performance over other methods in early fusion.

A video indexing system based on combination of low-level descriptors is presented in [47]. The effect of performing fusion at different levels, such as using low-level features or classifier results, is exploited in the paper, and a system without any fusion also is compared against a system including static feature fusion. Another work on content-based video indexing by using MPEG-7 visual features is proposed in [48]. Combination of individual classifiers within a neural network environment is presented, in which the information entropy is used to analyze the weights of classifiers trained on different features. A popular clustering algorithm, k-means, is initially applied on the low-level features, and then the features are mapped to weight vectors using the distances among the samples and cluster centers. Finally, a neural network based fusion algorithm is performed for weighting the features.

An adaptive boosting method for combining multiple classifiers is introduced in [49], in which the AdaBoost algorithm [50] is trained using the classifier errors in the initial evaluations. The proposed approach modifies the original AdaBoost so that the diversity between different classifier combinations is. The classifier combination producing higher diversity is used as kernel in the algorithm, and at the end an efficient combination is achieved.

A semantic concept extraction method employing static and dynamic feature fusion for the classification of soccer games using dimensionality reduction is presented in [51]. In order to reduce the redundancy along with the ambiguity among the low-level features, an early fusion scheme is performed in two ways. First, the features are merged into a feature vector using simple static operators such as *concatenation* and *average.* Secondly, a dimensionality reduction algorithm is applied to represent the data optimally. The model performs well in certain semantic concepts, such as *close up action*, *zoom on player,* and *center view* in soccer videos.

Another approach is used for ensemble learning of single classifiers using information entropy measures in [52]. The relationship between classification accuracy and entropy is attained by applying a genetic algorithm that uses the accuracy as cost function [53]. The majority voting is used to aggregate the individual classifier results, in which 120 ensemble classifiers are trained and the optimal 21 are selected for a final decision in the algorithm.

2.2.4 Summary

Apart from the aforementioned works, there are various approaches in achieving intelligent visual analysis systems in extracting high-level semantic interpretations in different application areas, such as scene classification, object detection and content based image retrieval are the most active ones.

Although there are several methods in different frameworks in the literature, the following steps are almost always performed in the semantic analysis of multimedia applications:

1) *Syntactic analysis:* This stage includes the representation of multimedia content, partitioning the visual content into meaningful parts such as shots, key frames, segments etc., and then a signal-level processing on them to obtain low-level features. This step can be grouped into two categories: the unimodal approaches that process different multimedia modalities independently, and the multimodal approaches that combine them in comprehensive ways [32].
2) *Classifier Learning:* This stage includes the processing of previously extracted low-level features by performing machine learning techniques to build semantic concept models. The data is usually divided into the training and test sets, in which the learning process is performed using the former set and the evaluations are applied on the latter. Different learning techniques can be applied in various architectures [28], but mostly they are far from providing unsatisfactory results in the classification of visual data [41].

3) *Domain Knowledge:* This step is performed by the approaches, which exploit domain information to utilize domain-specific representations [54] and rules [32]. Exploitation of domain knowledge usually improves the classification performance for detecting certain classes, but they have limited extensibility to other domains.

CHAPTER 3

VISUAL ANALYSIS SYSTEM

In this dissertation, we present a complete visual analysis system for the semantic analysis of visual data. Our system includes a number of novel approaches in bridging the semantic gap, and addresses some of the major challenges in the field by providing domain-independent methods. The general architecture of our system is shown in Figure 3.1. In order to enable better semantic representations of images, starting from the initial extraction of low-level features, we introduce a number of novel methods in different layers in the implementation of the system shown in Figure 3.1.

There are major differences between our visual analysis system and existing works, and some of the distinguished aspects introduced by our system can be listed as follows: First of all, an original system architecture is designed within a completely integrated framework for semantic analysis of visual data. Secondly, a fuzzy learning approach for high-dimensional feature spaces is introduced in classifier SVM training, which has not been investigated before. Thirdly, a unique feature weighting is used by examining the distributions of low-level visual features onto class-based trained SOM networks, which is mainly used to determine the confidence level of SVMs during the classifier combination process. Finally, a new high-level classifier fusion is implemented using the binary classification results obtained from the SVM classifiers along with the classification confidences. The DS theory is exploited in the combination of single classifier results, in which the basic probability assignments of information sources are computed individually in each classification process utilizing a probabilistic SVM evaluation method [55].

This chapter is organized as follows: Section 3.1 introduces the architecture of the visual analysis system in a functional point of view including the descriptions of core processes along with their relations to other tasks in the system. Section 3.2 provides the low-level computations performed on visual data, which applies both to both training and test images. Finally, Section 3.3 describes the construction and utilization of SOMs in the system with some illustrative examples.

3.1 The System Architecture

The visual analysis system depicted in Figure 3.1 includes several steps, and to give a better understanding of the system we group these steps into four core layers in a functional point of view in Figure 3.2. We refer each layer as a *module* or *process block* hereafter, the tasks performed in each layer along with the relations among them are explained in this section.

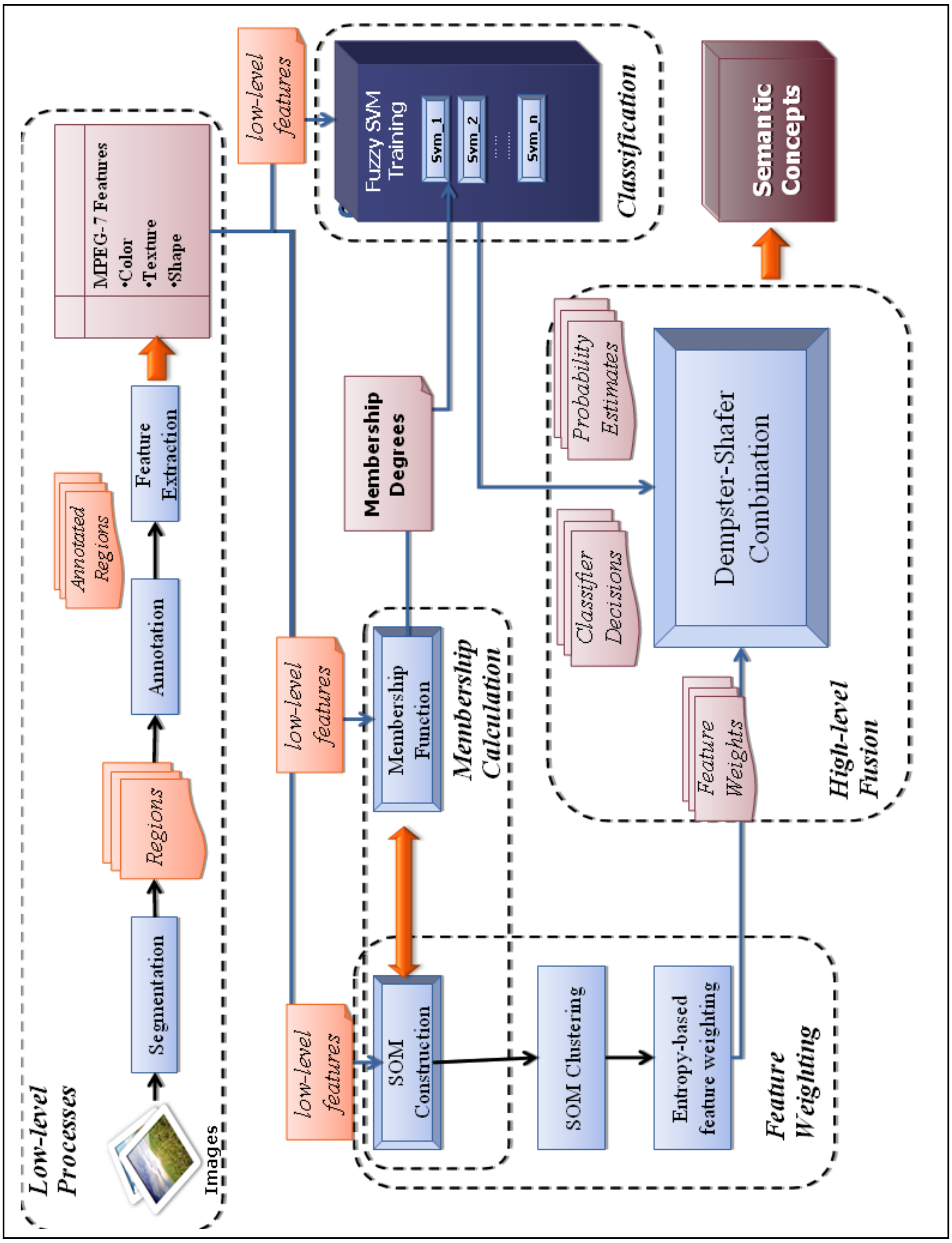


Figure 3.1 General architecture of the visual analysis system.

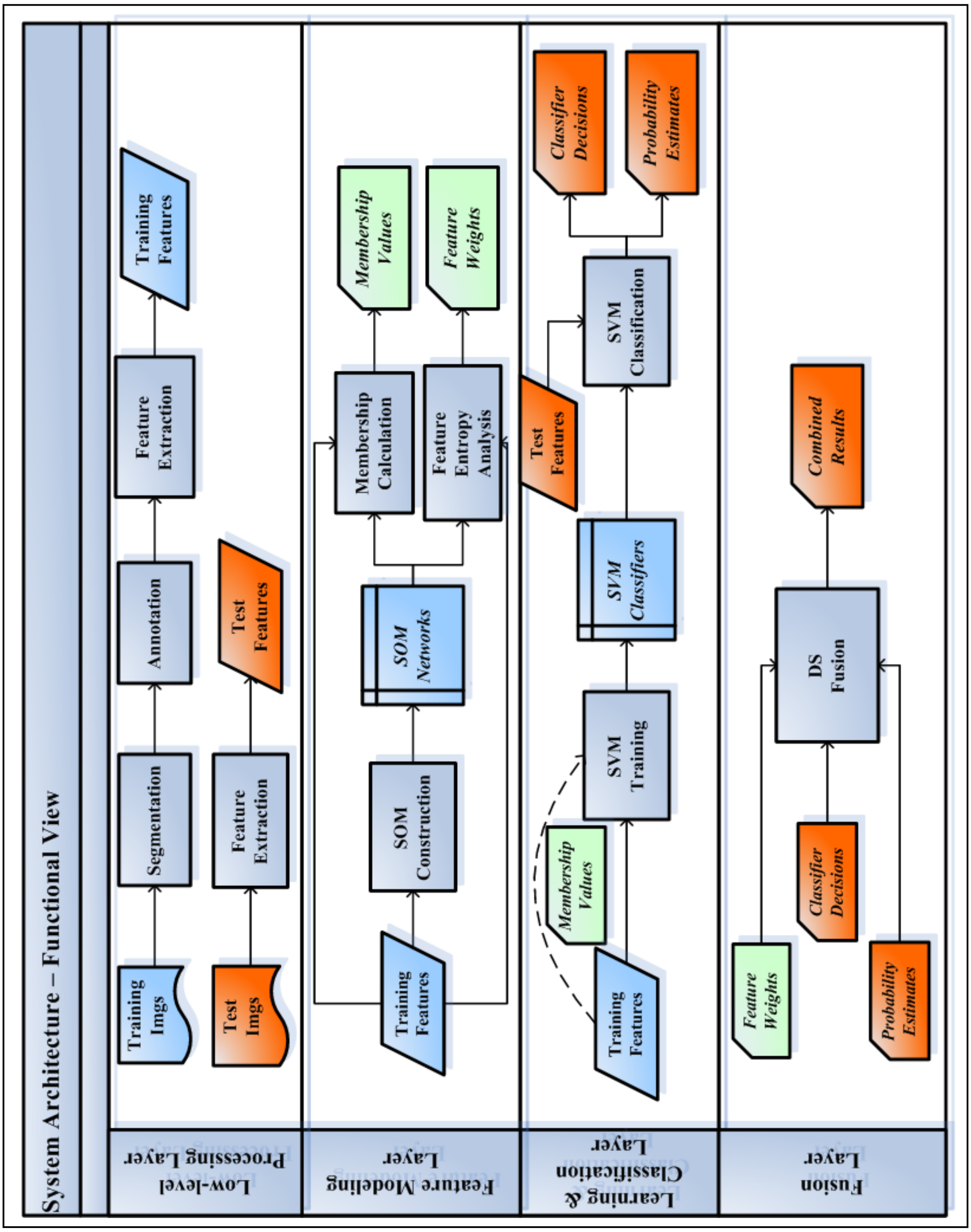


Figure 3.2 Functional view of the classification system.

1. *Low-level Processing Layer*: The first layer covers basically the low-level computations performed on the images in training and test collections. This layer has a number of individual steps, which are performed sequentially and applied differently on training and test phases.

The low-level computations performed on training images are as follows: 1) we apply an initial segmentation algorithm on the training images, and at the end each image is partitioned into a set of non-overlapping regions, 2) in annotation process, we assign keyword labels to each image region that contains a semantic entity, 3) the final step applied to training images is feature extraction, in which the low-level features are populated from the regions that is annotated before.

The low-level processes applied to test images are simple, we apply the feature extraction step directly on the image and the visual features of the test image are populated. As mentioned previously, these processes are applied both in training and test phases with minor differences, which are explained in Section 3.3 of the chapter.

2. *Feature Modeling Layer*: The second layer of our system analyzes the extracted visual features mainly utilizing neural networks. The tasks in this layer include the computations for modeling the low-level features extracted from the images of training set. The aim of establishing modeling in low-level feature space is to recognize the common patterns of the visual features, which are extracted from semantically similar regions from training data.

Contrary to the previous module, which both applies to training and test images; this layer deals only with the training. The SOM neural network [56] forms basis for the computations performed in this layer as follows:

- First, we group the low-level features according to their class labels, and then, perform SOM trainings by utilizing the low-level features individually. In this step, we build a separate SOM network for each image class and feature type by using the features extracted from corresponding image regions in training set.
- Once the SOMs are formed, we utilize them in the following two tasks: 1) to compute the membership degree of training data prior to the SVM training, and 2) to analyze the distribution of the low-level visual features on SOM output spaces for calculating the feature weights, which are used in the fusion process.

This module plays a key role in the implementation of the concept detection system, which forms the basis for two important contributions of this thesis:

- *Membership Calculation:* The membership detection process is used to determine the importance of the training samples during the classification task. This method improves the learning capacity of the typical SVM classifiers by decreasing the effects of noisy data in training collection. The details are explained in Chapter 4.
- *Feature Weighting:* The feature weighting process investigates the effectiveness of a low-level feature for a class. The basic idea behind weighting features is that some

features are more powerful in learning certain concepts than others. In order to determine the significance of a visual feature, we perform a feature weighting process in our visual analysis system. The computations are performed by analyzing the distributions of the low-level features on class-based trained SOMs. The best-matching-unit histograms in SOM mapping are used to calculate the weights. The details of the feature weighting approach are provided in Chapter 5.

3. *Learning and Classification Layer:* This layer has two functions that both applies to training and test data, respectively.

- *SVM Learning*: We use one of the state-of-art machine learning algorithms as the classification method in our visual analysis system. The SVM is one of the most popular learning techniques with extensive applications in various classification problems, including image analysis [6, 54, 57, 58]. Using the training features, we build binary SVM classifiers for each feature type and semantic class under the one-against-all approach [59]. At the end, each SVM is trained using a different feature types and becomes an expert of one semantic class. Hence, each SVM can classify between the positive and negative samples of one class.

  The training process applied in the dissertation differs from the previous approaches in that before starting the training process of SVM classifiers, we initially assign membership degrees to low-level features in training set. The training samples are treated according to their membership degrees, and the approach improves the overall learning capacity of the SVM by reducing the negative effects of noise data during the classification process. The theory of SVM along with a fuzzy approach in SVM learning is explained more detailed in Chapter 4.

- *Binary Classification:* Secondly, in the classification step of this layer, we input the low-level features extracted from test images, and obtain the binary classification results along with the probability estimates in each classification. In this step, each SVM is run individually according to the feature type, i.e. the SVM that is trained for that feature type. Each SVM provides two outputs at the end of the classification process: the binary decision given by the SVM, and the probability estimates, which are calculated using the value in SVM decision function. Both values are utilized in the fusion step. More information on these steps is given in Chapter 5.

4. *Fusion Layer:* The fourth and final layer in our visual analysis system is the *Fusion Layer*, which combines the results of binary classifiers into a final decision. As can be seen in Figure 3.2, the fusion process uses the following parameters: the *feature weights* obtained in *Feature Modeling Layer,* the *binary classification results* and *probability estimates* of the *Learning and Classification Layer*.

We apply a modified version of the DS combination rule [9] in the fusion step. In this combination scheme, we use the single SVM classifiers as experts or evidence providers, the feature weights as trust level of the experts, and the probability estimates in classifications as confidence levels. We exploit the individual classifier confidences along with the feature weights in each combination, and this enables an optimum combination during the combination of classifiers.

3.2 The Low-level Processing Layer

The extraction of low-level features from input images data forms the initial steps in the overall classification system. The computations performed in the *Low-level Processing Layer* are shown in Figure 3.3. The purpose of this layer is to perform low-level computations on training and test images, and extract the low-level visual features from them. These features are used to represent the color, shape and texture attributes of the images. In this section, we describe the following steps, respectively: 1) *Segmentation*, 2) *Annotation*, and 3) *Feature Extraction*.

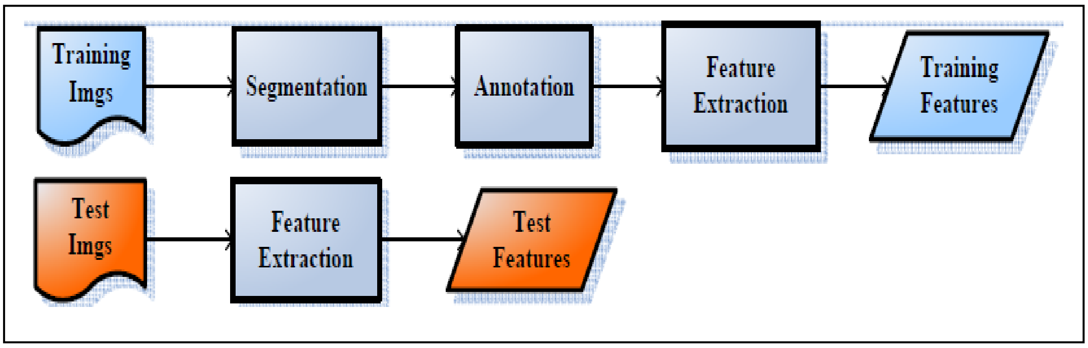


Figure 3.3 Low-level processes performed in the system.

3.2.1 Segmentation

Image segmentation is the process of portioning images into non-overlapping image regions, which is an active research area in image processing and computer vision. The aim in segmentation is to partition the image into homogeneous regions that have common characteristics, which facilitates the further applications performed on the image. The segmentation is used as a preliminary step in many image processing applications including images and videos [60], face detection[61], content-based image retrieval [44], object tracking [62], and so on.

We use the image segmentation step to identify the important image regions and to annotate the training data, if it does not include annotations. Instead of building the classifiers on the whole image data, we initially apply a segmentation process and determine the important image regions such as an object or a part of a scene. We apply the segmentation step prior to feature extraction and thus the classifier learning and low-level feature modeling layers are directly affected by the segmentation method. The segmentation process enables to eliminate the irrelevant image regions during the computations, which both reduces the size of input data in computations and gives more robust learning methods.

This initial segmentation of images helps to achieve better feature representations, which eventually improve the overall concept detection since the features are extracted locally from the segmented regions that contain perceptually dominant areas of images.

There are several image segmentation techniques that can be grouped into two main approaches: pixel-based local methods [63], and region-based global methods [64, 65]. In our classification system, we apply the JSEG segmentation algorithm [60], which carries out the segmentation task in two independent steps: color quantization, and spatial segmentation. In the first step, a color quantization is performed in color space to differentiate the regions of the image without considering any spatial distributions. Each pixel in an image is represented by its color class, and the pixels are replaced by their corresponding color classes, which forms a class-map of the image. In the second part of the segmentation process, a spatial segmentation is performed on this class-map without considering the pixels' color similarities. The class-map of each color region is called a *J-image* in the algorithm [60], which represents the edges and the textures of the input image. Final image segments are determined by performing a region growing method on the J-images. A sample image and its regions after the JSEG method is applied are shown in Figure 3.4.

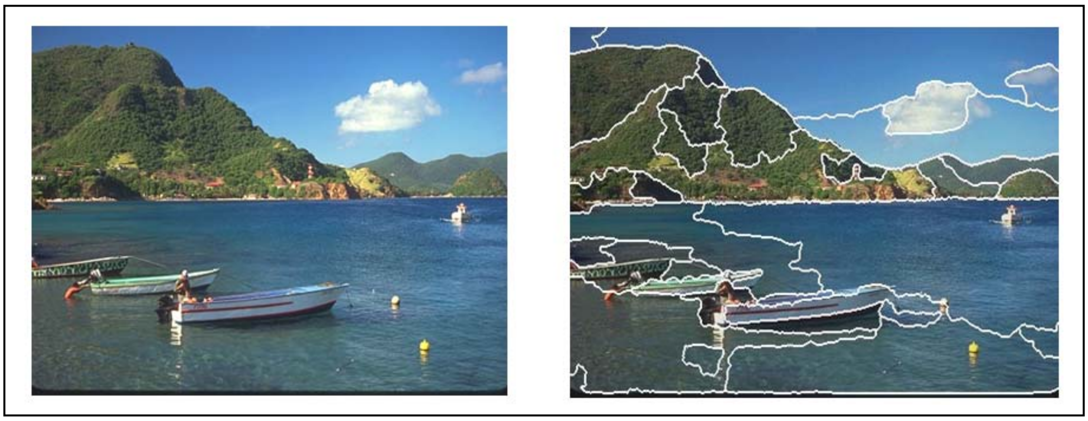


Figure 3.4 A sample image and regions after JSEG segmentation.

3.2.2 Annotation

Image annotation, specifically automatic annotation of images [66], is an active research area in several image processing applications [31, 67], which also requires the extraction of high-level semantic features from images. However, we do not interest in the automatic annotation of the images in this study. The aim of the annotation step is to assign high-level semantic descriptions to image regions and eliminate the irrelevant image parts in feature extraction process so that the quality of the extracted descriptors can be improved during the implementation of classifiers.

Annotation can be performed automatically, semi-automatically or manually in general, but the manual annotation achieves the highest accuracy concerning the descriptions of semantics. As a consequence we choose a manual annotation approach in system implementation, although it is the most time consuming type of annotation method.

There already exist some annotation tools for image processing, such as [68], which provides a semi-automatic annotation tool for image and video files. There are also MPEG-7 based annotation tools such as M-Ontomat-Annotizer [69], and IBM VideoAnnEx [70]. They both aim to support automatic video shot detection and manual tagging of objects with bounding boxes. However, the support for automatic or semi-automatic annotation of images is rather poor in these tools, and the available tools have limited usability since they do not support the annotation of image segments and bounding boxes at the same time.

In order to support the annotation and enable the labeling of the regions in training data, a manual annotation tool is developed in the scope of this dissertation. This provides to assign keyword annotations on the image segments which are generate by the previous image. The tool uses the image regions, and corresponding segmentation masks generated in the segmentation step. Then, it operates on each region independently, when a region is selected by selecting a pixel inside a region, it is automatically labeled with the keyword, which denotes the class label for the region and given as a parameter to the tool. The annotation tool provides the annotation of more than one region at the same time, and it also enables to select a bounding box to annotate a region. A sample image and annotation process is presented in Figure 3.5.

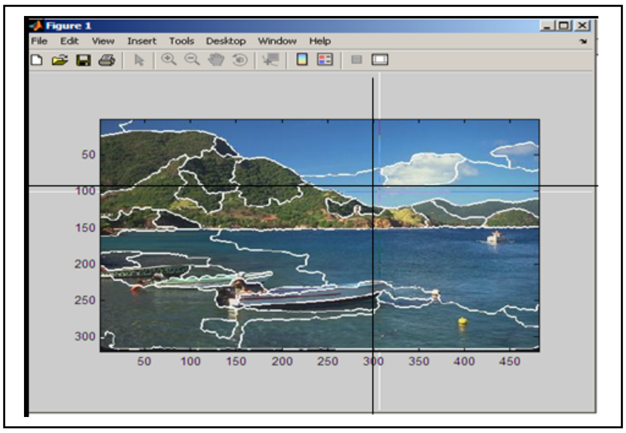


Figure 3.5 A sample run of the annotation tool.

3.2.3 Feature Extraction

The third and final step in *Low-level Processing Layer* is the extraction of low-level visual features from the annotated image regions. Feature extraction is the process of generating low-level feature descriptors, which are directly used in various tasks such as classifier training, feature weighting, and so on. Hence, the extraction and utilization of the low-level features are the most critical steps in this layer and directly affects the performance of the whole classification system. The end of the feature extraction step is a set of low-level image features, i.e. feature vectors, which represent different attributes of the input images.

The low-level features used in the classification system are selected considering the following properties:

1) The features that carry enough information to obtain high-level semantic information in visual data,
2) The features that do not require any domain-specific knowledge,
3) The features that are easy to compute, and can be applicable to large image collections efficiently,
4) The features that are most related to human perception system.

Based on these considerations, we select to use the color, shape and texture based features in the implementation of our classification system, since these are the most closely related visual features to human perception and are widely used in various image applications. However, there is not a single and best representation of these features. In order to provide standardized descriptions for multimedia data, the MPEG-7 standard [19] specifies a set of descriptors by defining the syntax and semantics of low-level visual features as mentioned in Chapter 2.

Although the MPEG-7 standard specifies different content descriptors for multimedia data, only a part of them are applicable to 2-D images. Table 3.1 lists the available feature types, and their applicability to different multimedia modalities [19].

Table 3.1: The MPEG-7 features and their usability.

Feature	Audio	Image	Video
Color	-	X	X
Shape	-	X	X
Texture	-	X	-
Motion	-	-	X
Camera Motion	-	-	X
Time	X	-	X
Audio features	-	-	X

An evaluation of the MPEG-7 visual features is presented in [17]. The author analyses the MPEG-7 descriptors from a statistical point of view to examine the quality of the MPEG-7 visual descriptors for content based image retrieval applications. In [17], analysis of eight visual descriptors on three different media collections is presented including monochrome texture-only, colored, and a set of synthetic images. The main results show that: 1) the best descriptors are *Color Layout, Dominant Color, Edge Histogram* and *Texture Browsing* considering the redundancy and sensitivity analysis of the descriptors, and the others are highly dependent on them, 2) The color histograms (*Color Structure* and *Scalable Color*) perform badly on monochrome data, and 3) The MPEG-7 descriptions can be augmented by additional descriptors for better image representations.

3.3 The Feature Modeling Layer

There are a number of motivations to apply a modeling in low-level feature space before applying the classifier learning and fusion steps. First of all, we exploit the large amount of low-level features obtained from training data, to discover the intrinsic patterns in each class. We use SOM for this task, which can be applied to view and cluster high-dimensional feature vectors on 2-D spaces.

Secondly, once we obtain the common patterns by utilizing the training features, we can evaluate the effect of each training vectors in classifier learning, since the training images may contain noise and this affects the determination of optimum decision surface in SVM learning [71]. Hence, we apply a fuzzy approach in classifier training, in which each training data can contribute differently on the construction of the decision surface.

Thirdly, by distributing the low-level features extracted from image regions in training set, we can examine the distributions of the low-level features over semantic concepts, and analyze the randomness of the distributions on different SOM spaces. We exploit this information to evaluate the relative importance of each low-level feature for image classes.

3.3.1 Self-Organizing Maps

The SOM [56, 72] is a fully connected two-layer neural network based on competitive learning for exploring and clustering high-dimensional datasets. The SOM is trained in unsupervised manner and provides an ordered map of input signals by examining the internal structure of the input signals and coordinating the connections between the units.

A sample SOM is shown in Figure 3.6, in which a lattice of nodes and the neurons are placed at them. Each neuron of SOM is fully connected to the input vectors and contains a weight vector of the same dimensionality as the input space.

The SOM runs in two modes; training and mapping. In the training phase [73], the network organizes itself using a large numbers of inputs and the map is constructed and in a competitive process. Once a SOM is built, it can be used to map new input vectors of same dimensionality [74].

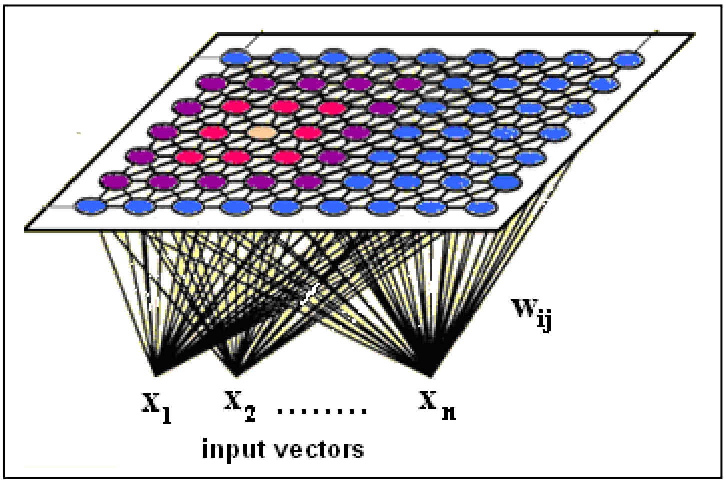


Figure 3.6 A Sample SOM Network.

During the mapping phase, the input vectors are located on the map by finding the closest SOM unit, which is also a classification of input vectors.

SOM training is a competitive and consists of the following steps:

1) *Initialization:* The weight vectors of all SOM neurons are initialized,
2) *Sampling:* A vector is chosen from the training data randomly,
3) *Similarity matching:* The SOM unit with the closest Euclidean distance to the selected vector is chosen as the best-matching-unit (*bmu*),
4) *Updating*:
    a. The neurons that are neighbor to the BMU are found by performing an exponentially decaying neighborhood function; the neighborhood radius shrinks in following iterations,
    b. Each neighboring neuron's weights, including the BMU, are adjusted to become more like to the input vector. The farther away the neighbor is from the BMU, the less its weights get altered ,
5) *Continuation*: The process is repeated several times until convergence of network. At the end; each SOM unit is assigned a weight vector of the same dimensionality of the input data.

In the mapping mode of the SOM, a new input vector is compared with every SOM unit and the closest SOM unit to the input vector, i.e. the unit with minimum Euclidean distance, is selected as the *bmu* for the input.

3.3.2 The construction and utilization of SOMs in visual analysis

In this dissertation, the SOM is primarily used as a tool to map the high-dimensional visual features onto more compact spaces, and to represent the intrinsic patterns of semantic classes by utilizing the SOM units as cluster centers.

The SOM training and mapping has two important properties in these computations, which are extensively exploited throughout this study in the classification of images [75]:

- *Topology-preserving*: The SOM mapping carries the topology information of the input space, i.e. the elements that are near in the input space locates closely in the output space, while distinct patterns are mapped on the distributed regions of the map [76].
- *Clustering of high-dimensional input sets:* The SOM can also serve as a clustering method for high-dimensional spaces [77]. As described in SOM training, the input vector produces effect on the *BMU* and its neighboring units in each iteration, and hence similar patterns produce a spatial ordering in the output map so that the units that are close to each other form a cluster on the output map.

The SOM is used for the two core tasks throughout the dissertation (Figure 3.1): the *Membership Calculation*, and *Feature Entropy Analysis*. The topology-preserving property of the SOM forms the basis of the former task, where membership degree of a training sample is computed by finding the distance of the sample to the corresponding SOM structure. The latter task makes use of the clustering property of the SOM, which exploits the distribution of feature vectors on concept-based SOMs. The details of the Membership Calculation are provided in the next section while the feature weighting process based on feature entropy is defined in Chapter 5.

For each image class and feature type we train a separate SOM as shown in Figure 3.7.

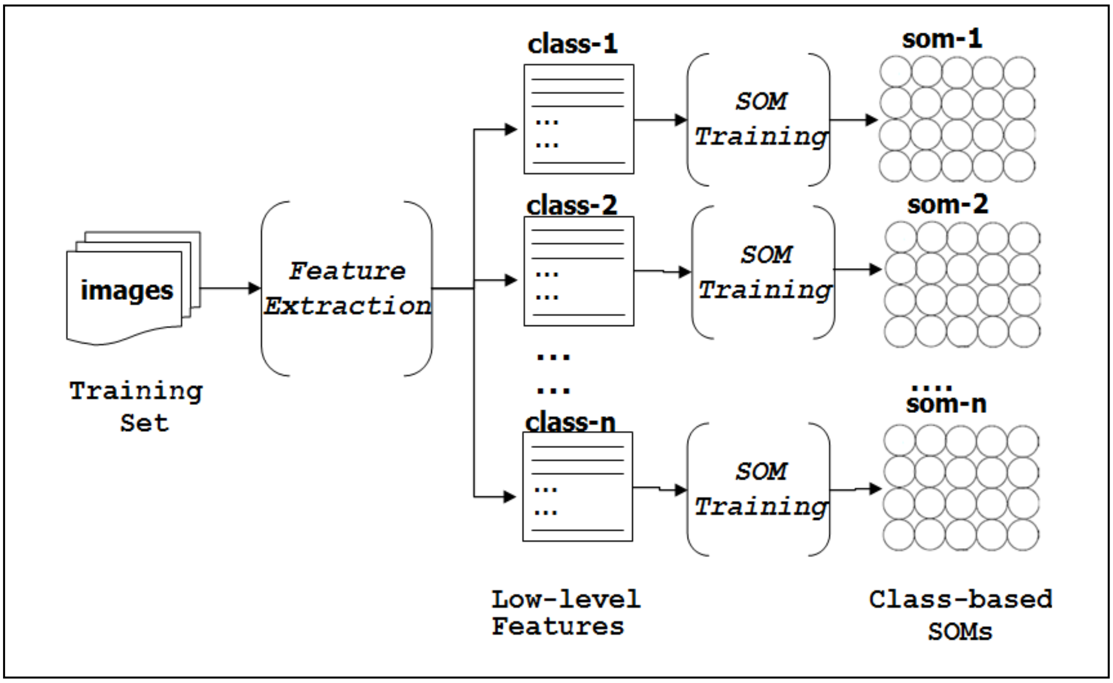


Figure 3.7 Construction of SOM networks.

Figure 3.8 illustrates a more detailed view of the SOM construction process shown in Figure 3.7, at the end each SOM unit is assigned a weight vector the same as the input feature. Figure 3.9 shows a sample mapping of the low-level features on a class-based trained SOM.

The SOM neural network forms the basis for several computations throughout the implementation of the image classification system in this dissertation. The analysis of low-level features for membership calculation and feature weighting are explained in Chapter 4 and Chapter 5 respectively. In Chapter 6, we investigate the generation of visual codebooks by performing SOM clustering, and compare its effectiveness against other methods.

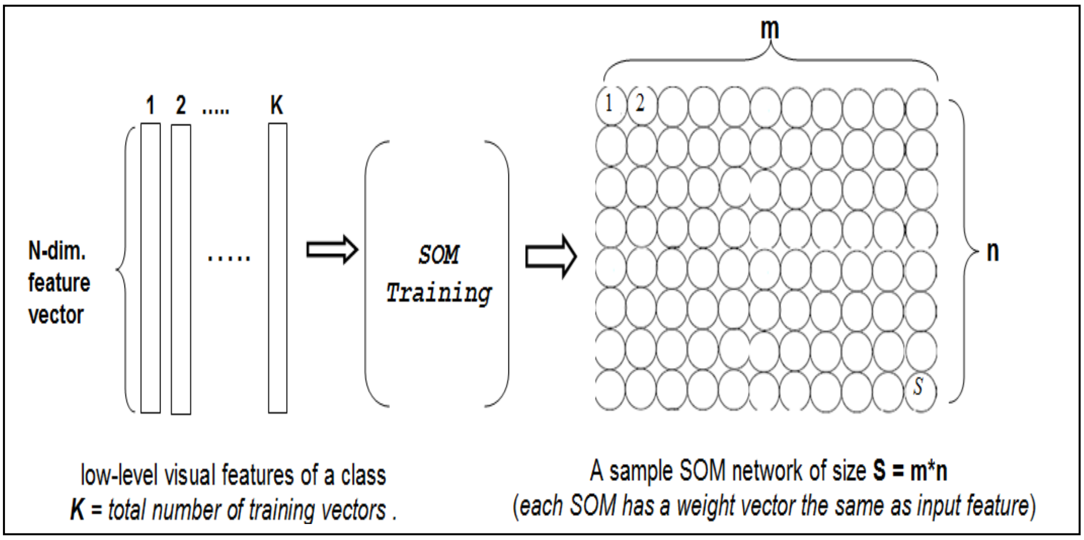
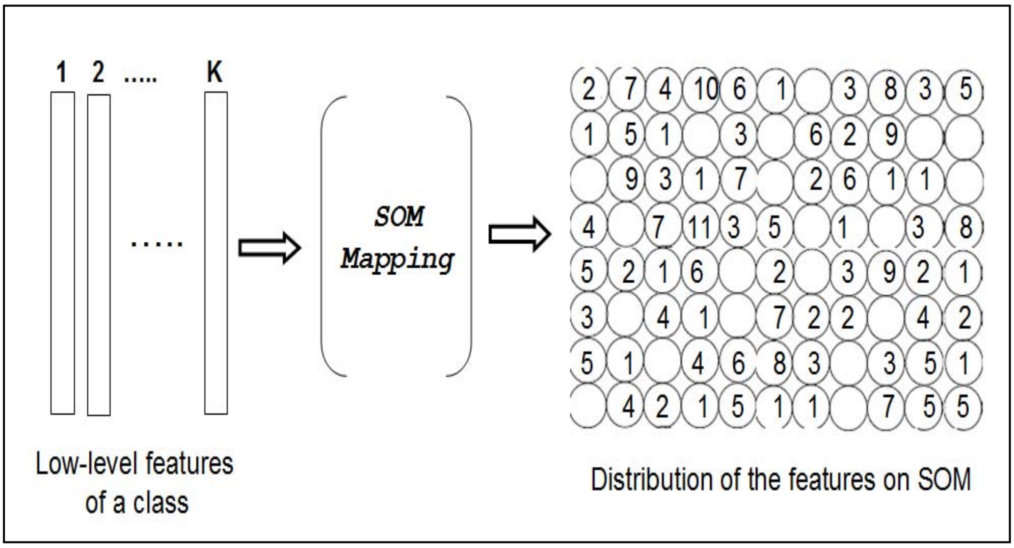

m
1 2 ..... K
N-dim. feature vector
.....
SOM Training
1
2
S
n
low-level visual features of a class
K = total number of training vectors .
A sample SOM network of size S = m*n
(each SOM has a weight vector the same as input feature)

Figure 3.8 SOM training for a sample class.


1 2 ..... K
.....
SOM Mapping
Low-level features of a class
Distribution of the features on SOM

2	7	4	10	6	1		3	8	3	5
1	5	1		3		6	2	9		
	9	3	1	7		2	6	1	1	
4		7	11	3	5		1		3	8
5	2	1	6		2		3	9	2	1
3		4	1		7	2	2		4	2
5	1		4	6	8	3		3	5	1
	4	2	1	5	1	1		7	5	5

Figure 3.9 The distributions of features on SOM space with bmu-hits.

CHAPTER 4

LEARNING IN VISUAL ANALYSIS

The learning phase of our visual analysis system includes a number of sequential processes within an integrated system architecture as described in Chapter 3. This chapter provides the tasks applied in the classifier learning phase, and includes the membership calculations of the low-level features in training set, and the construction of binary SVM classifiers.

There are two main tasks in the classifier learning process shown in Figure 4.1: 1) *Computation of Memberships*, and 2) *SVM Training*. As can be seen in the functional view of our classification system (Figure 3.2), these tasks are employed in different layers in the system: the *Feature Modeling* and *Learning Layers*, respectively.

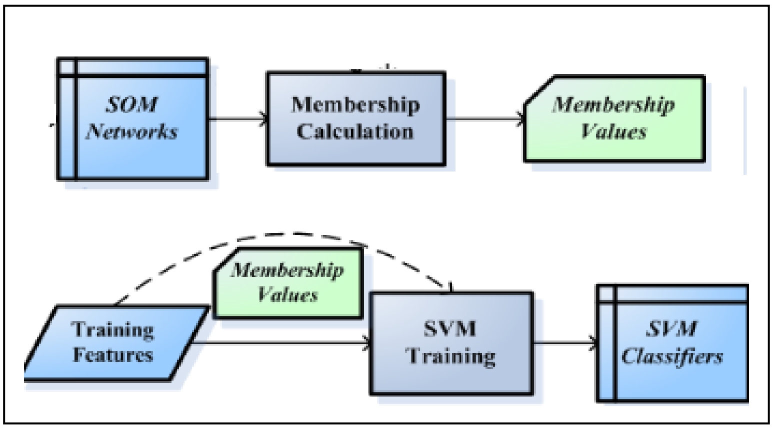


Figure 4.1 Classifier learning process.

The ultimate goal of the learning in visual analysis is to construct classifiers to evaluate the presence or absence of semantic classes in unlabeled image collections. There are different learning techniques using various frameworks ^{i}n the literature, to obtain high-level semantics from low-level visual features, as summarized in Chapter 2.

The Support Vector Machine (SVM), with high generalization ability and better performance in pattern recognition, is a state-of-art classification technique with successful applications in many real-world problems including image analysis [71, 78-80]. Although the SVM is very effective in many classification problems, the method is far from providing adequate level of accuracy in the extraction of high-level semantics from visual data due to the well-known semantic gap problem [3, 5, 81].

The SVM treats each training sample equally during the construction of its decision boundaries (e.g. support vectors), which makes a major limitation in SVM is learning especially if the training data includes noise. This drawback results in to generate poor machines for visual classification problem since the low-level features extracted from training images generally includes too much noise. In order to address this problem in SVM training, we apply a fuzzy learning approach in our classification system through assigning each training sample a fuzzy membership degree. The fuzzy approach in SVM training is first introduced by Lin et al. to improve the SVM learning [82] by assigning membership values to each training sample during the construction of SVM decision surface. This approach reduces the effects of the outliers in training data. The success of the proposed method depends highly on applying an appropriate membership function for the application domain, and in the literature, although different membership functions are presented, they are mostly designed for simple classification problems [80, 83, 84], and none are suitable for the low-level visual features that have high-dimensional feature vectors.

In this dissertation, we propose a unique approach in classifier learning process by introducing a new membership function based on SOM neural networks. The new approach improves the learning capacity of the individual classifiers by reducing the effect of noise in training images prior to the construction of classifiers. As described in Chapter 3, we build class-based individual SOMs by utilizing only the low-level features extracted from corresponding image classes. These SOMs are exploited to calculate the membership degrees of the training samples by computing the distance of each training sample to its best-matching-unit (*bmu*) on SOM.

On of the contributions of this thesis is to construct an enhanced learning approach in visual analysis or classification of semantic classes by utilizing the fuzzy learning method in the system, which has not been investigated before. Another major contribution in the field is the presentation of a new membership function to calculate the membership degrees of high-dimensional visual features. The proposed approach has a number of advantages over the existing methods in the computation of membership degrees, and our membership methods can also be utilized for applications that have the complex feature spaces.

The organization of this chapter is as follows: Section 4.1 provides the basic concepts related to the SVM Theory, which is the main learning technique used in this thesis. Section 4.2 introduces the fuzzy extension in SVM learning along with some of the current membership calculation methods. Section 4.3 presents our membership function on SOM networks with some illustrative examples. Finally in Section 4.4, we present the steps performed in training phase for building SVM classifiers in the scope of our analysis system.

4.1 SVM Theory

The SVM has two important properties, which make it an efficient learning technique in classification problems: 1) *high generalization capability*, and (2) introduction of *kernel trick*. The high generalization is achieved by searching an optimal hyperplane to discriminate the positive and negative classes during its learning process, which enables SVM to maximize the margin between the two classes. The kernel function is used when the samples of two classes can not be separated in their original forms. The mapping provided by kernel enables SVM to search an optimal hyperplane in higher dimensional spaces. The details of the approach are given in the following subsections.

4.1.1 Optimal Hyperplane

Given a data distribution of the form $\{(x_1, y_1)......(x_n, y_n)\}$ $\in$ R x $\{-1,+1\}$, and each x_i is a data point with labels $y_i \in \{-1,+1\}$: SVM tries to find a function that correctly classifies the data of the distribution, which is also applicable to unseen data patterns.

As an example data distribution is given in Figure 4.2 in which different hyperplanes can be used to correctly separate the two classes of the data.

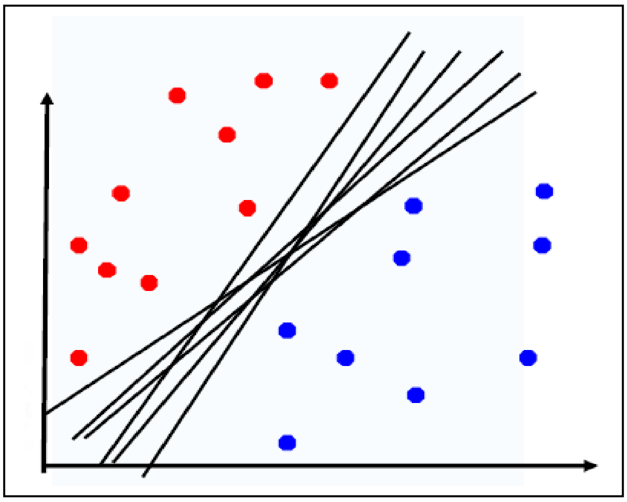


Figure 4.2 Separating hyperlanes in a 2-d space.

In order to guarantee the best generalization performance, the SVM searches for the hyperplane having the largest margin. The optimal hyperplane is unique and provides the maximum separation among the data of two classes, since the remaining hyperplanes might be closer to one class in the initial data distribution, and hence do not produce correct classifications for newly seen data.

For example, an illustration of such hyperplane is depicted in Figure 4.3, in which two parallel hyper planes are built on each side of the separating hyperplane and ***w*** represents the margin length.

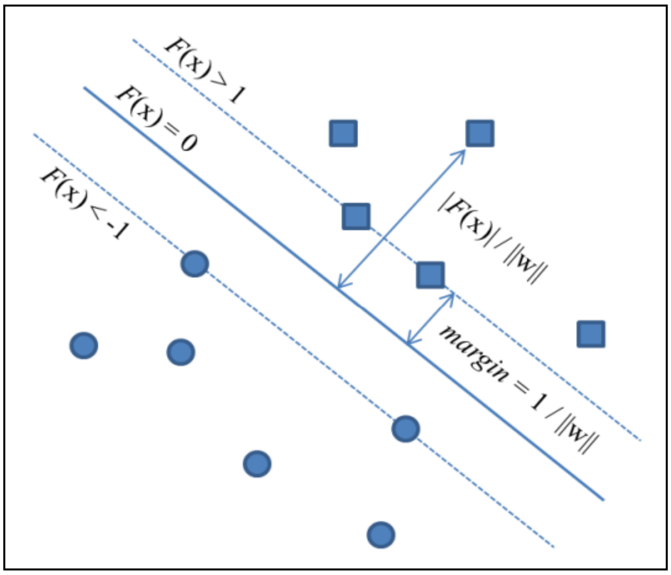


Figure 4.3 The hyper-plane maximizing the margin in a two-dimensional space.

Formally, suppose we have a training set $\{(x_1, y_1)......(x_n, y_n)\}$, where each x_i belongs to one of the classes $y_i \in \{-1, +1\}$ for $i = 1, 2, ..., n$. If this set can be linearly separable, the SVM finds the hyperplane $w.x + b = 0$ as the largest margin between the two classes. The classification is performed according to the following decision function:

$$f(x_i) = \text{sign}(w.x_i + b) = \begin{cases} 1, & \text{if } y_i = 1 \\ \text{-1}, & \text{if } y_i = -1 \end{cases} \tag{4.1}$$

Maximizing the margin requires minimizing ||w||, and thus the learning problem becomes a quadratic optimization:

$$\begin{cases} \min \frac{1}{2} \| w \|^2 \\ y_i (w^T x_i + b) \geq 1, \quad i = 1, 2, ...n \end{cases} \tag{4.2}$$

, where *w* is the weight vector and *b* is the bias term.

Given such an optimization problem, it is always possible to construct another problem, which is called as the *dual problem* having the same results as the original one.

By applying the Lagrange multipliers [85] to Equation (4.2), we can rewrite the dual problem as follows:

$$\begin{cases} \text{maximize: } Q(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{subject to: } \sum_i \alpha_i y_i = 0 \\ \qquad \alpha \geq 0 \end{cases} \tag{4.3}$$

The solution of Equation (4.3) is the form of:

$$\mathbf{w} = \Sigma \alpha_i y_i \mathbf{x_i} \qquad b = y_k - \mathbf{w^T x_k} \tag{4.4}$$

, for any $\mathbf{x_k}$ such that $\alpha_k \neq 0$.

Each non-zero α_i in Equation (4.4) indicates that the corresponding point x_i is a support vector. But, the optimal Lagrange multipliers, denoted by α_i^*, must satisfy the following condition [85].

$$\alpha_i^* \{ y_i (\mathbf{w}^* . \mathbf{x_i} - b) - 1 \} = 0 \text{ for } i = 1,2,....,m \tag{4.5}$$

Hence the optimum weight vector depends only on the support vectors, which coefficients are nonnegative. Once the nonnegative Lagrange multipliers and the corresponding support vectors are found, we can compute the bias *b* using a positive support vector x_i :

$$b^* = 1 - \mathbf{w}^* . \mathbf{x_i} \tag{4.6}$$

Finally, the classification of a new data point can be done by the following decision function:

$$F(x) = \sum_i \alpha_i y_i \mathbf{x_i} . \mathbf{x} - b \tag{4.7}$$

, where each $\mathbf{x_i}$ is support vector.

An important property of Equation (4.7) is: it only depends on the inner product between of the test points $\mathbf{x}$ and support vectors $\mathbf{x_i}$ Therefore, solving the optimization problem involves computing the inner products between all pairs of the training points, which allows the generalization capability in nonlinear problems. The application of SVM to these problems is explained in Section 4.1.3.

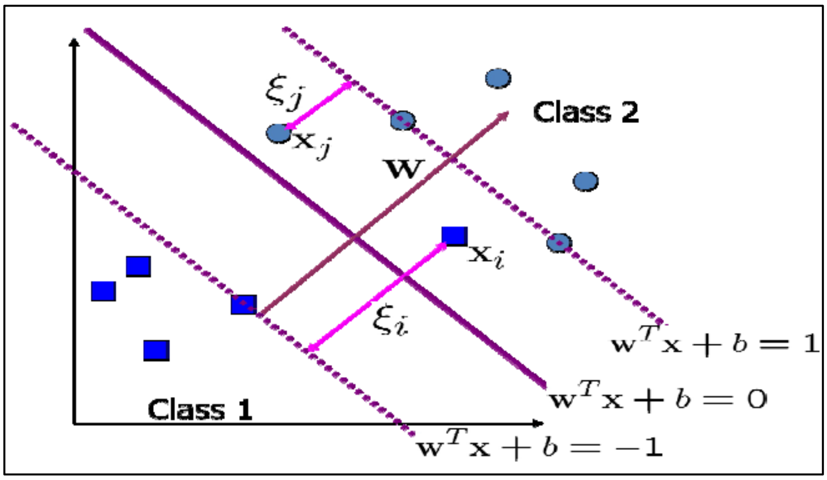
4.1.2 Soft Margin SVM

In some classification problems, the input data cannot be linearly separable unless some classification violations are allowed. As can be seen in Figure 4.4, it is not possible to correctly satisfy all data points, and some classification errors must be accepted to find a maximum margin hyperplane.

In this classification scheme, the constraints in Equation (4.2) are relaxed by introducing some slack variables, $\xi_i \geq 0, i = 1, 2, \ldots n$ which allow some classification errors for some difficult data points.

The quadratic program in Equation (4.2) becomes:

$$\begin{cases} \min \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{n} \xi_i \\ y_i (w^T x_i + b) \geq 1 - \xi_i, \quad i = 1, 2, \ldots n \end{cases} \tag{4.8}$$



ξ_j
$\mathbf{x}_j$
Class 2
$\mathbf{w}$
$\mathbf{x}_i$
ξ_i
$\mathbf{w}^T\mathbf{x} + b = 1$
Class 1
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

Figure 4.4 A soft-margin SVM.

Here, the C is a parameter to control the trade-off between the maximization of the margin and the minimization of the errors, i.e. a large C value results in narrow margins with more misclassifications, while small C values results the SVM to ignore more data points with a wider margin. The amount of total misclassification is controlled by aggregation of the individual errors in the classification, i.e. $\sum_{i=1}^{n} \xi_i$.

If we follow the similar steps as introduced in Section 4.1.1, we can obtain a similar dual problem identical to the linearly separable case, except with the introduction of an upper bound on C on α_i now.

$$\begin{cases} \text{maximize: } L(\alpha) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j \\ \text{subject to: } \sum_i \alpha_i y_i = 0 \\ \quad 0 \leq \alpha \leq C \end{cases} \tag{4.9}$$

4.1.3 The Kernel Trick

In most real world problems, the initial distribution of the training data is too complex, and it is not possible to separate the classes linearly generally. In this case, the SVM first maps the data from the input space to some higher-dimensional feature spaces, and then a linear separating hyperplane is searched in the new space. The process is shown in Figure 4.5.

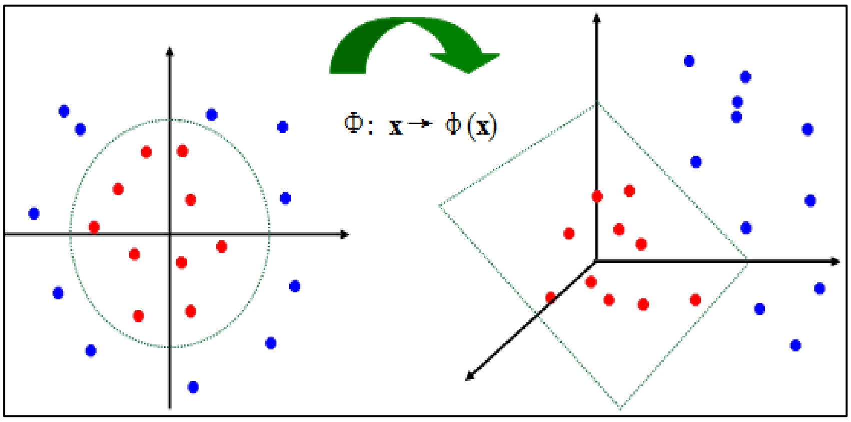


Figure 4.5. The non-linear separable data in input space and mapping to feature space.

The solution of Equation (4.8) is not practical for most real-world problems. To make it clearer, the data shown in Figure 4.5 cannot be separated even misclassifications are allowed, and the solution can only be found if the initial data points are mapped to a higher dimensional space as demonstrated in Figure 4.5.

Although the training set is not linearly separable in the original input space, it is linearly separable in the new feature space, and, the maximal separating hyperplane can be found in the new feature space. A linear classification in the high-dimensional feature space corresponds to a nonlinear classification of the original input space, and this method is called as *kernel trick.*

As stated in the previous section, the optimal hyperplane and SVM decision function utilize only the dot products among vectors in the input space. The problem for non-linear case is solved by introducing a mapping function, $\phi(x_i)$, which must satisfy the Mercer's condition [58]. Under this condition only, the product of the data points can be translated to the high dimensional domain by applying mapping to data points: $\phi(x_i)\ .\ \phi(x_j)$.

However, it is not necessary directly perform the mapping function, $\phi(x_i)$, to solve the quadratic program in Equation (4.8), we only require a kernel function $K(x_i, x_j) = \phi(x_i)\ .\ \phi(x_j)$ that computes the dot product of the data points in the new feature space. Therefore, instead of using the mapping $\phi(x_i)$ function directly on inputs, an inner product of points by applying the kernel function can be utilized, which avoids the curse of dimensionality problem in high-dimensional feature spaces.

The kernel trick plays a critical role in SVM formulation and its performance, since the inner product $\phi(x_i)\ .\ \phi(x_j)$ does not need to be evaluated in the feature space. Functions that satisfy the Mercer's theorem [71] can be used as dot products and hence can be used as kernels. This property enables the computations to be performed in the input space rather than the high-dimensional feature space.

The nonlinear separating hyperplane can now be found by using the Lagrange multipliers and the kernel method, solving the following equivalent problem:

$$\begin{cases} \min \frac{1}{2} \sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{j=1}^{n} \alpha_j \\ \sum_{i=1}^{n} y_i \alpha_i = 0 \\ 0 \leq \alpha_i \leq C, \qquad i = 1, 2, ..., n \end{cases} \tag{4.10}$$

, and the decision function becomes:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \text{b} \right) \tag{4.11}$$

There are several alternatives that can be used as the kernel function in SVM. The only requirement for the function is that it must satisfy the dot-product of the inputs in the original feature space. The most predominantly used kernels in the literature for SVM training are as follows:

- Polynomial kernel: $K(x_i\,,x_j) = (1 + x_i\,.\,x_j)^{\mathrm{d}}$
- RBF kernel : $K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2*\sigma_j^2}}$
- Sigmoid kernel : $K(x, y) = \tanh(x^T y + \theta)$

4.2 Fuzzy SVM

Although the SVM is a powerful learning technique in different classification problems, it also has some limitations: First, the SVM is very sensitive to noise, which affects the learning of the optimal decision surface inversely. Secondly, the training samples are assumed to belong to either of one class, i.e. either negative or positive class in binary classification, and in real-world problems, some data points such as outliers do not exactly belong to either of the classes. Thirdly, the effect of each training sample in SVM training is same since the SVM treats every training data point uniformly. This affects the detection of optimal hyperplane in most classification problems, since some of the training samples can be more important than the others in the classifier learning process.

In order to solve these problems, the SVM Theory is extended by Lin et al. by a new with fuzzy learning approach, which the authors called the new machine a Fuzzy SVM [82]. In the fuzzy approach, each training sample is assigned a membership value representing the attitude of the sample to one class, and this way different training points can contribute differently on the construction of the decision surface during the learning phase. The proposed approach has the potential to improve the SVM classification performance, since it decreases the effect of outliers or noise in training data once the membership degrees of training data are computed properly.

Formally, suppose that we have a set of training samples $\{(X_1, y_1, \mu_1)......(X_n, y_n, \mu_n)\}$, and each X_i belongs to one of the classes $y_i \in \{-1, +1\}$, with a fuzzy membership value $\sigma \le \mu_i \le 1$, for $i = 1, 2, ..., n$, and a small constant $\sigma > 0$. The fuzzy memberships, μ_i, associated to each training data, x_i , can be regarded as the attitude of corresponding data point to one class during the classification process. In this case, the quadratic problem of Equation (4.8) can be described as follows:

$$\begin{cases} \min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\mu_i\,\xi_i \\ y_i\,(w^T\phi(x_i) + b) \ge 1 - \xi_{i,}\ \ i = 1,2,...n \end{cases} \tag{4.12}$$

, for $\xi_i \ge 0, i = 1, 2, ...n$.

In this formulation, the term ξ_i represent a measure of the error in SVM, and $\mu_i\xi_i$ gives the total amount of error in fuzzy SVM. Hence, a smaller μ_i value would reduce the effect of error ξ_i in Equation (4.12), and decreases the effect of the corresponding input point, x_i in SVM training. The solution of Equation (4.12) is performed similarly as the classical case with slight difference [82].

4.2.1 Existing Membership Functions

In the literature, several membership functions are proposed for calculating the membership values in fuzzy SVM training. For instance, Lin et al. [82] have defined a membership method according to the distance between a sample and its class center, within their original fuzzy SVM paper. Similar approaches are also investigated by some researchers to find a suitable membership calculation method with minor modifications [83, 86, 87]. The method is as follows: 1) a clustering method is performed on the initial data, 2) the distances of training points to their closest cluster centers are located, 3) a scaling function is applied to calculate the membership values of training data directly in the input spaces [88].

To make it clearer, given a set of training samples $\{(x_1, y_1, \mu_1)......(x_n, y_n, \mu_n)\}$, and x_+ denoting the mean class $\{+\}$ and x_- denoting the class center $\{-\}$, first the radii of each class is found, for a binary classification scheme:

- Radius of + class is determined by: $r_+ = \max | x_+ - x_i|$
- Radius of – class is found by: $r_- = \max | x_- - x_i |$

And, in general, the fuzzy membership values of data samples are determined by using a function on the mean and radius values for each class, as follows:

$$\mu_i = \begin{cases} 1 - | x_+ - x_i | / (r_+ + \delta) \ , \ \mathit{if}\ y_i = 1 \\ 1 - | x_- - x_i | / (r_- + \delta) \ , \ \mathit{if}\ y_i = \text{-}1 \end{cases} \tag{4.13}$$

A different way of calculating membership degrees for fuzzy SVM training is presented in [83]. In this approach, the relations of a sample to its cluster center also considering the relations among the samples themselves by describing a fuzzy connectedness of the training samples [61]. This is defined by performing a KNN distance calculation on the training points.

Another membership calculation method is proposed in [80], in which both the correctly and incorrectly classified data points are used in the calculation. The authors focus on the data points in the mixed regions, where both data points occur, and assign them higher fuzzy membership values than the other points.

Most of the above-mentioned methods define the membership values using the distance among the training points to their class centers, in which the computations are directly performed in the original input space. The methods assume a prior knowledge with a compact data distribution, and none consider the high-dimensional feature spaces as in the case in image analysis. For instance, Lin et al. assume that the outliers in training set can be detected by computing the relative distance of samples to their class centers [87], which is true under limited conditions as explained in [84]. Hence a new way to compute the membership values of low-level visual features is required in order to apply a fuzzy learning method to classical SVM.

The detection of memberships for the high-dimensional visual features is not a trivial process since the features that are extracted from similar image regions do not always have common distributions in the low-level feature spaces. In this dissertation, we propose a unique approach to address the limitations of the previous membership calculations methods. The new approach provides an enhanced membership detection method such that unlike the previous membership functions which directly make calculations in the original feature space, we first map the very high-dimensional feature vectors into lower dimensions introducing a neural network approach, and then perform the membership calculations in the new low-dimensional spaces.

The new membership function utilizes the class-based trained SOM networks, which are introduced in Chapter 3. The main aim in using a SOM mapping for membership calculation is to analyze the high-dimensional visual features by mapping them onto more compact, 2-D discrete maps, in which the spatial locations of the neurons on SOM includes the inherent statistical patterns of the features.

4.3 A Novel Membership Function

The most critical part in fuzzy SVM is the selection of an appropriate membership function, which determines the level of importance of the training samples prior to the classifier learning process. In this section, we present the main motivations in our membership calculation approach, and provide the utilization of the method the in image classification.

To the best of author’s knowledge, the membership functions defined in the literature for fuzzy SVM training are not suitable for our problem for the following reasons:

- The image classification problem requires the analysis of complex low-level features and none of the earlier approaches deal with such high-dimensional features,
- The feature extraction task distributes the visual features sparsely in the original feature space; even they are extracted from semantically similar regions.
- Applying a clustering method on visual features in their original spaces does not produce the desired clusters due to the semantic gap problem,

Since the visual features do not have a common pattern distribution in low-level space, calculation of membership values in these spaces often produce poor results. Thus, a suitable membership detection function must be defined for the high-dimensional visual features to be able to perform fuzzy SVM learning in our system.

4.3.1 Membership Calculation for Low-level Visual Features

In this dissertation, we introduce a novel membership function utilizing the SOM neural networks. In this approach, we use the SOMs to map the low-level feature of semantic classes considering each SOM unit as a cluster center for the class. In other words, the mappings of low-level features onto the output layers of concept-based constructed SOMs can be viewed as a special clustering on the visual features. We use the distance of each training sample in this mapping as a similarity measure of the sample to the corresponding class, since each SOM is are built individually by utilizing the low-level features of that class only. Hence, the distances input vectors to their corresponding *bmus* is used as a measure of the (dis)similarity among the images to semantic classes. The membership detection algorithm is as follows:

The method is applied as follows:

1) For each SOM structure; the mean and maximum distances among the SOM units are found, since each SOM is of different size and weight vectors.
2) Each input vector is mapped to a corresponding SOM network. The SOM is determined according to the semantic class and type of the feature the SOM trained in the construction phase. This mapping computes the minimum distance between an input vector to the closest SOM unit.
3) A normalization is performed on the calculated distance, which on two factors: a) the average distance or quantization error of SOM units, b) the binary sign of the input vector for the class, i.e. the input vector may be either a positive sample or not. This normalization is important in determining the membership values, since we either use the average distance among SOM unit or maximum distance according to the label of the feature vector.
4) The membership value of the input is determined by applying a RBF on the normalized SOM distance of the low-level feature.

Algorithm 4.1 The calculation of membership values of low-level features.

```
Input: A set of low-level features, and semantic annotations of the form $\{(X_1,C_1)......(X_N,C_N)\}$, where each $X_i$ is assigned a semantic class $C_i \in \{Set\_of\_Concepts\}$ for $i = 1,2,...N$ in the classification system.
Output : The membership degrees of input features.
1: Calculate the SOM distance for each input vector.
        $dist_{i,j} = distance(X_i, S_j)$
2: Compute the average and maximum distances in each SOM.
```

```
        $mean\_qerr_j = mean\_distance(S_j);$
        $max\_qerr_j = \max\_distance(S_j);$
    3: Form the normalized SOM distances according to the label of the input vector.
        $\text{if}(\ C_i == C_j\ )\quad \text{D\_norm} = \begin{cases} 0 & ,\ \text{if } \text{dist}_{i,j} <= \text{mean\_qerr}_j \\ \dfrac{\| \text{dist}_{i,j} - \text{mean\_qerr}_j \|}{\text{mean\_qerr}_j} & ,\ \text{otherwise} \end{cases}$
        $\text{else}\quad \text{D\_norm} = \begin{cases} 0 & ,\ \text{if } \text{dist}_{i,j} >= \text{max\_qerr}_j \\ \dfrac{\| \text{dist}_{i,j} - \text{max\_qerr}_j \|}{\text{max\_qerr}_j} & ,\ \text{otherwise} \end{cases}$
    , where $C_i$ is the annotation class of the input vector and $C_j$ is the SOM that's used in membership calculation.
    4: Finalize the membership through applying a Gaussian function.
        $\mu_{Xi} = e^{-\frac{\|D\_norm\|^2}{2*\sigma_j^2}}$
```

4.3.2 Membership Determination Strategy

We propose a heuristic function in the calculation of memberships in Algorithm 4.1 by making an assumptions as the closer a positive training sample's feature vector to corresponding SOM (e.g. the *bmu*), the more representative it for that class, and its contribution is increased during the SVM learning phase. This assumption holds in our system since we build the SOMs by utilizing only the positively labeled training images and each SOM is trained individually for certain classes in our visual analysis system.

Suppose that $D(x_i, bmu_j)$ is a distance measure between the data point x_i and SOM unit bmu_j in feature space. In order to differentiate between a noisy and a representative sample in training set, we focus on two measures: the distance between the sample and corresponding bmu, and the sign of the sample as follows:

- If x_i is a positive instance, i.e. the annotation of x_i is same as the SOM's concept, we use the pre-calculated mean of SOM in normalization, where if $D(x_i, bmu_j)$ is less than or equal to the mean it is considered to be zero (Algorithm 4.1).
- For negative instances, we take the maximum distance between any two of the SOM units as normalization factor and use in normalization, in case $D(x_i, bmu_j)$ is greater than this value, the sample is considered to be a good example of negative class (Algorithm 4.1).

The normalization applied to the SOM distance is necessary, since each SOM is trained using different number of training features, which determines the size of the network. Additionally, since the SOMs are trained independent of each other, they also differ in dimensionalities, and different weight vectors.

After performing the above-mentioned normalization, we employ a Gaussian function on the normalized SOM distance:

$$K(x_i, x_j) = e^{-\frac{\|x_i - x_j\|^2}{2*\sigma_j^2}} \tag{4.15}$$

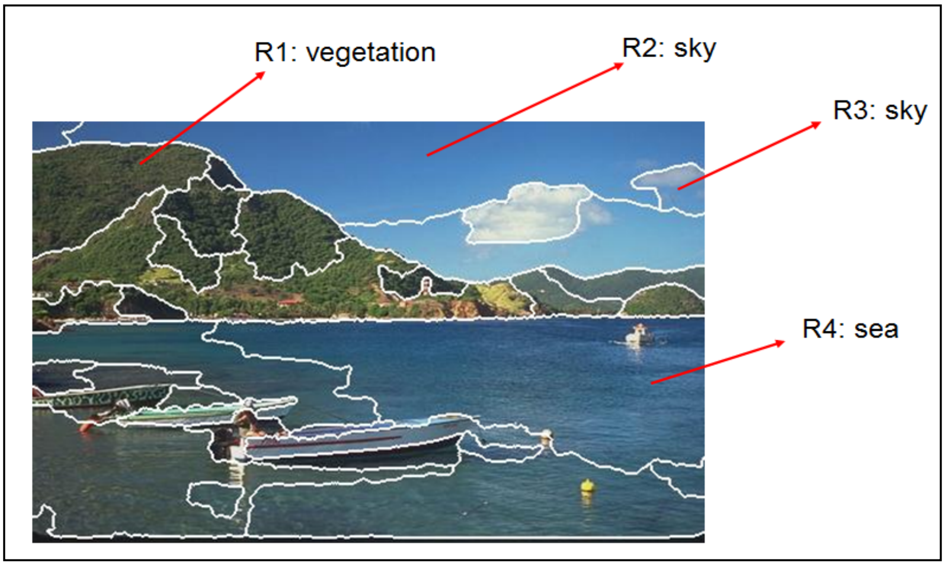
Now, we check the following two situations:

- For a positive training sample x_i, the closer the point x_i to a SOM, the more representative it is for the class: the membership of x_i is increased by the RBF. Otherwise, it is evaluated less important by RBF, since the normalized distance is low and the RBF reduces the weight in that case.
- For a negative instance, the opposite performs: the higher a data points distance to a SOM, the more membership value is given to the point, since the point is a good negative sample for the class now, and the RBF increases the point's membership degree accordingly.

To summarize, the proposed membership function provided in Algorithm 4.1 degrades the effect of noisy samples in negative and positive regions in training data, and results in an improvement in the classification performance of the classical SVM.

As a sample image with region annotations is shown in Figure 4.6. The membership values in Table 4.1 are calculated by using the SOM trained by the low-level features extracted from the images that contain at least one *sky* object. Although the regions R2 and R3 are both positive samples for the *sky* class, they have different membership values as can be seen in Table 4.1. The membership of R2 is calculated higher than that of R3 since the former region is a better representation of the class in the sense that R3 includes some white cloudy parts in it.

Analyzing the negative regions in Figure 4.6, the distance of the features extracted from R1 to the SOM is greater than that of R4, which makes R1 a better negative sample than R4. Additionally, R4 has similar color attributes (e.g. heavy blue) as *sky* regions, and the memberships calculated by CSD and EH descriptors are also different since the CSD is a color-based feature type.



33

Figure 4.6 A Sample image with annotations.

Table 4.1: The SOM distances and membership values for sky concept.

Region	Descriptor	SOM Distance	Membership Value
R1	CSD	8.7	0.91
R2	CSD	4.3	0.94
R3	CSD	5.6	0.67
R4	CSD	6.3	0.45
R1	EH	7.2	0.83
R2	EH	3.2	1.00
R3	EH	4.9	0.76
R4	EH	6.9	0.71

The advantages of the use of SOM networks in feature analysis can be listed as follows:

1) The computations are performed in a 2-D space, which provides also computational efficiency in the evaluations.
2) The SOM is sensitive to frequent input patterns, and although the extracted visual features do not have common patterns in their original spaces, the SOM construction approach enables to catch the common input patterns on training data.
3) The SOM runs fully unsupervised manner, which means that a prior learning or labeling of data is not needed to form the clusters and can be built without using of information of image classes.

4) The SOM is used as a clustering method that produces two-dimensional clusters on its output space, which preserves the topology of the features.

5) The relationships between the low-level features extracted from image regions can also be visualized, and some training samples may be grouped closer to each other than they would in a one-dimensional cluster analysis.

4.4 The Construction and Utilization of Binary Classifiers

Once the membership values are assigned to the training samples, we start building binary SVM classifiers in our classification system as shown in Figure 3.2. The classifier learning includes constructing several binary SVMs by using the low-level features of the training data along with the memberships associated to each training vector. In this step, we run the SVM learning algorithm under the one-against-all approach [89], and at the end the support vectors for each semantic concept are computed individually.

In the SVM implementation, the following steps are performed to obtain binary classifiers:

1) *Data Preprocessing*: The training data should be represented in separate vector sets since the SVM requires vectors in same dimensionalities. Therefore, we grouped the low-level feature vectors extracted from training images into different groups. Then, for each semantic class we transform the form the required training vectors and create individual SVM training data packages by adjusting the class labels to each sample accordingly.

2) *Scaling*: In order to avoid the numerical difficulties in calculations, and the attributes in greater numeric ranges dominating those in smaller numeric ranges, a scaling must be performed before the actual SVM training process. This enables effective kernel computations on the high-dimensional feature vectors without causing numerical problems.

3) *Kernel Selection*: We select to use RBF kernel in SVM training. The main reason in selecting this kernel is the capability of the RBF to nonlinearly map the input vectors to higher dimensional spaces, which is a required step in image classification.

4) *Parameter Search using Cross Validation*: The parameter selection is usually applied prior to the classification task, to provide better classification rates that are not known beforehand. We perform a parameter search using the 5-fold cross-validation technique [90] before the actual classification starts as follows: The training set is divided into 5 subsets of equal size, and each time we test a subset using the classifier trained on the remaining 4 subsets using different parameters. At the end, the accuracy is determined according to the percentage of the correctly classified data for each case and the optimum kernel parameters are reached.

5) *Use of Membership Degrees:* In order to use the membership degrees assigned to each training sample, we a second weight file that contains the membership degrees of the training samples. Since the classifiers are used to learn different concepts, the same training sample might have different membership values in each classification process.

6) *SVM Training*: After preparing the required input packages for each classification task, the classifier training process is performed, and separate binary SVM

classifiers are built for each class and descriptor type. At the end, each SVM is capable of discriminating one semantic class from the remaining classes, and provides a binary classification result on test data.

We also employ a probabilistic learning model in SVM in order to approximate the classification confidences in the scope of our classification system. The approach is explained in Chapter 6, and the implementation of the binary SVMs performing both classical learning and fuzzy learning along with their performance results is described in Chapter 7.

CHAPTER 5

HIGH-LEVEL CLASSIFIER FUSION

Fusion of individual classifier results is a crucial task in multi-classification systems since the classification performance can be improved by applying an optimum combination method. The fusion can be applied in different levels of abstraction ranging from data or feature level to decision-level depending on the architecture of the system. In our visual classification system, we build several binary SVMs, which are individually trained using different feature types for each image class. In the Fusion Layer of our classification system (Figure 3.2), we perform a novel fusion method exploiting the binary SVM results along with the classification confidences and feature weights. In this chapter we describe our approach in classifier combination and design of high-level fusion process.

Different combination techniques can be applied for classifier fusion. One general approach to combine the outputs of classifiers is majority voting, which is commonly applied in handwritten text recognition [58]. The voting approach does not require any learning process before the fusion, in which the final decision is determined according to the class label most represented by the classifiers.

There are fusion methods that require an additional training, such as the weighted average combiner. In this technique, learning is applied prior to the fusion process, example works can be found in [91, 92]. Another approach in high-level fusion is the combination of weak classifiers that are trained on different parts of the problem independently. The purpose in this approach is to build a strong classifier from several single or weak classifiers trained on same learning techniques with different parameters or cost functions.

Third classes of ensembles develop the combiner during the training of the individual classifiers, for example, AdaBoost.

In order to provide an optimal merging of the results obtained by binary SVM classifiers, and to improve the overall system performance in visual analysis, we implement a new classifier fusion framework for in this dissertation. Our fusion method is based on Demspter-Shafer Evidence Theory [93], in which we utilize three distinct information sources or evidences during the combination process as follows: 1) the individual SVM classifications, 2) the degree of confidence in each classification, and 3) the weight of low-level features for each class.

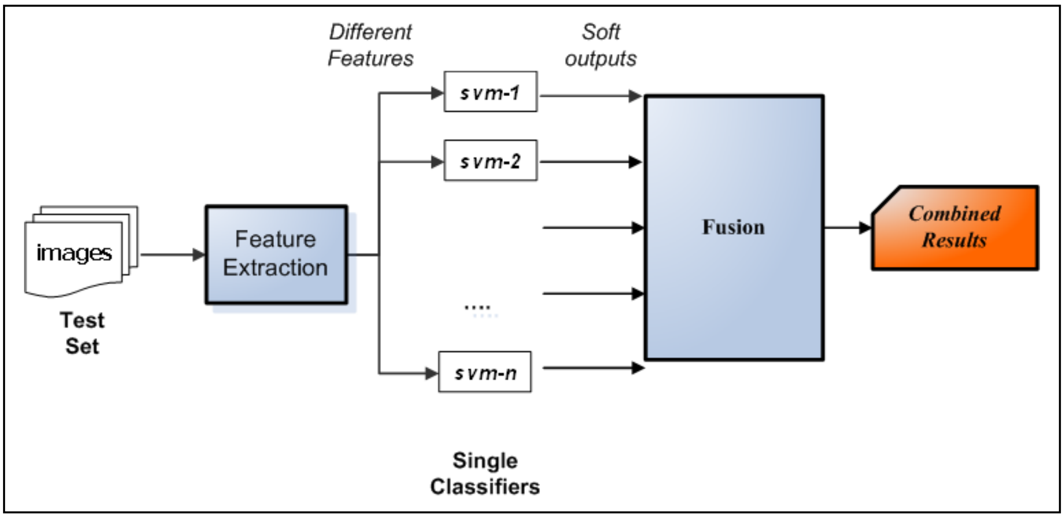


Figure 5.1 The fusion process in visual analysis.

The first parameter in our fusion approach is binary classification results obtained by several SVMs. Since we train one SVM for each feature type and image class, for a single test image, a number of classification results are obtained from the SVMs, with different decisions on the label of the image since the SVMs are trained separately using different feature sets. The second information used in the fusion process is the classification confidences, which are computed by a probabilistic approach using the absolute value of SVM decision function. The third and final input to our classification system is feature weights, which represent the importance or effectiveness of low-level features on image classes.

We have explained the SVM classification process in Chapter 4. In this chapter, we present the remaining two steps, namely, the determination of classification confidences and the computation of the effectiveness of low-level features in the scope of our classification system. The rest of this chapter is organized as follows: Section 5.1 explains the basics of Demspter-Shafer Theory, Section 5.2 introduces the topology-preserving property in SOM clustering along with the their utilization in feature weighting. Section 5.3 gives information about the determination of classification estimates in SVM using a probabilistic approach. Finally, Section 5.4 presents the fusion process of the entire system and the application of a modified DS combination rule in the scope of the thesis.

5.1 Dempster-Shafer Theory

In this dissertation we use the DS Evidence Theory to exploit the information obtained in the visual classification system in for an optimal combination of individual SVM classifiers. The Dempster-Shafer (DS) Evidence Theory [93] was introduced as a method for representing uncertainty. The DS, contrary to the Bayes probability, associates evidence to multiple events.

There are three important functions used in the DS Theory: 1) the basic probability assignment (*bpa*), 2) the Belief (*Bel*), and 3) the Plausibility (*Pl*) as follows:

- *Basic probability assignment (bpa):* The value of *BPA*, or *mass* of a set A, is represented as $m(A)$, which represents a particular element belongs to the set A but any particular subset of the set A [94]. In other words, the *BPA* represents the degree of evidence, in which the element in question belongs exactly to the set A.
- *Belief (Bel):* The measure $bel(A)$ is the degree of evidence that the element in question belongs to set A as well as to the various subsets of A [94]. Belief can be interpreted as the total amount of support given to A.
- *Plausibility (Pl):* The quantity $pl(A)$ shows the total degree of evidence that the element in question belongs to set A or any of its subsets [94]. The plausibility can be interpreted as the maximum support that can be given to the set A.

The DS can combines multiple beliefs by using their *bpa*'s, once the belief functions are defined on the same frame of discernment and based on independent evidences. Given two *bpa*'s, the DS combines all the supportive propositions as follows:

$$m_i \oplus m_j(A) = \frac{\sum_{E_k \cap E_{k'} = A} m_i(E_k).m_j(E_{k'})}{1 - \sum_{E_k \cap E_{k'} = \varnothing} m_i(E_k).m_j(E_{k'})} \tag{5.1}$$

As mentioned before, there are two basic assumptions in order to apply the DS combination rule to a specific problem, these are: 1) the sources or information provides must be defined in the same frame of discernment, and 2) they must be independent of each other. Considering our classification system, the information sources that will be used in DS combination are the binary SVM classifiers, and the two rules hold since the classifiers are trained independent of each other and defined in the same application domain.

The DS rule in combining multiple evidences can also be interpreted as a generalization of the Bayes rule [95]. The DS combination rule mainly focuses on the agreement between the different sources, and ignores all the conflicting evidence by applying a normalization factor. This approach is criticized by some researchers [96], especially in the case of strict conflictions in evidences, the normalization applied in Equation 5.1 can cause dramatic flaws and hence incorrect results by aggregating only the common on information sources. To address this problem, a modified DS combination is proposed in the literature to represent the degree of the conflict in the combination.

There is another problem in the DS combination, which fails to differentiate among different sources since the information providers may be of different importance to the problem. However, the DS combination in Equation 5.1 trusts each source equally, which can be suitable if the observations have similar accuracies or confidences. This approach is not realistic in many real-world problems including image classification, since the individual classifiers have different confidences in the evaluations.

In order to address this problem, a modified DS combination method is presented in [9], which assigns weights to the information sources, and apply the following DS combination:

$$m_i \oplus m_j(A) = \frac{\sum_{E_k \cap E_{k'} = A} wi\, m_i(E_k).wj\, m_j(E_{k'})}{1 - \sum_{E_k \cap E_{k'} = \varnothing} wi\, m_i(E_k).wj\, m_j(E_{k'})} \qquad (5.2)$$

In Equation (5.2), each evidence or information provides is assigned a weight factor representing the importance or the trust in the evidence. The determination of the weights for the sources is domain dependent: some works use the previous results obtained by the observer's decision, and so on.

In order to implement the DS combination in our visual classification system, the following input parameters should be defined first: 1) the information sources of our classification system, 2) the calculation of *bpas*, in the context of image analysis, and 3) the weights that will is assigned in the combination to achieve an optimal combination.

The first parameter is the individual binary SVMs generated in the training phase for each semantic class. As described in Chapter 4, each of SVM classifier is an expert of one single image class, and their classification results are used as the primary evidence source in the fusion. The second parameter used in DS combination is the detection of classification confidences in SVM evaluations, in which we apply a probabilistic classification approach, which is described in Section 5.2. The third area parameter to be fielded is the feature weighting method, which is based on the entropy of the distribution of low-level features on different semantic classes. Section 5.3 explains the detailed information in this process.

5.2 Probability Estimates in SVM Classification

In a binary classification problem, the SVM evaluates a point according to the sign of the following decision function:

$$F(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b \qquad (5.3)$$

The decision function in the above equation carries two kinds of information:

1) The sign representing the class of the classification, i.e. whether a positive class or not, and,
2) The absolute value representing the distance to the hyperplane; this value is related shown to represent the strength in the decision, i.e. the farther a sample point locates away from the decision boundaries the stronger the decision is [97].

The distance of the points to the SVM separating hyperplane is shown to be related with the confidence of the decision in SVM classification [97], and can be used to represent the probability approximation in classification. Several experiments was performed on various of handwritten symbols for classification in [98], and the authors show the classification error decreases exponentially with the distance from the SVM hyperplane, which is shown in Figure 5.2.

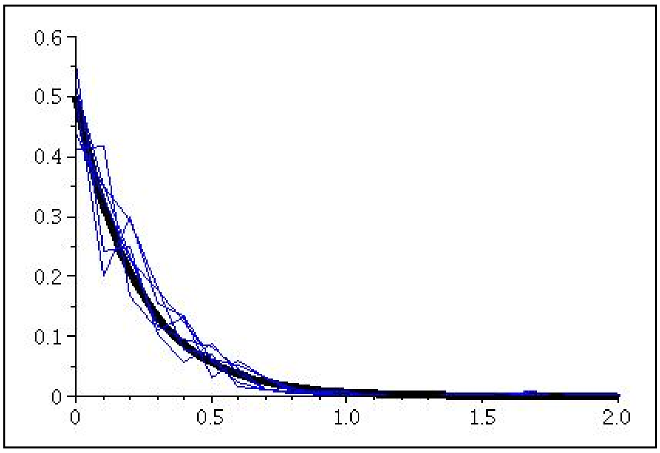


Figure 5.2 SVM classification error rate vs. distance to hyperplane.

The determination of the classifier confidences is approached by performing the classification output in a probabilistic approach proposed in [55]. This approach is useful to obtain the classification results in terms of probably estimates, since the output of the SVM is a distance from the separating hyperplane and can not be directly used as a probability value. Platt [55] introduced a parametric model to fit the posterior class probabilities with the adaptation of an additional sigmoid function in Equation 5.4, during the classification task. The method trains the SVM with the parameters of the sigmoid function, and then maps the outputs of the SVM into probabilistic estimates.

$$\Pr\,(y=1|\,f)\approx P_{A,B}(f)=\frac{1}{1+e\,(Af+B)} \tag{5.4}$$

, where $f=F(x)$ in Equation 5.3.

Instead of estimating the class-conditional probabilities, Platt uses a parametric model that fits the posterior $P(y=1\,|\,f)$ directly, and the parameters of the model are adapted to give the best probabilistic results [55].

In the classifier fusion, we employ the Platt's approach in SVM training and train the parameters in Equation (5.4) so the concept classifiers also produced the posterior probability values in classification task. The classifier probabilities are used as the *BPA* or *mass* value in DS combination, as mentioned in Section 5.1.

A test sample image and annotations applied on the regions are shown in Figure 5.3. The binary classifications results of the annotated regions applying different concept classifiers that are trained on *Color Structure* descriptor is provided in Table 5.1.

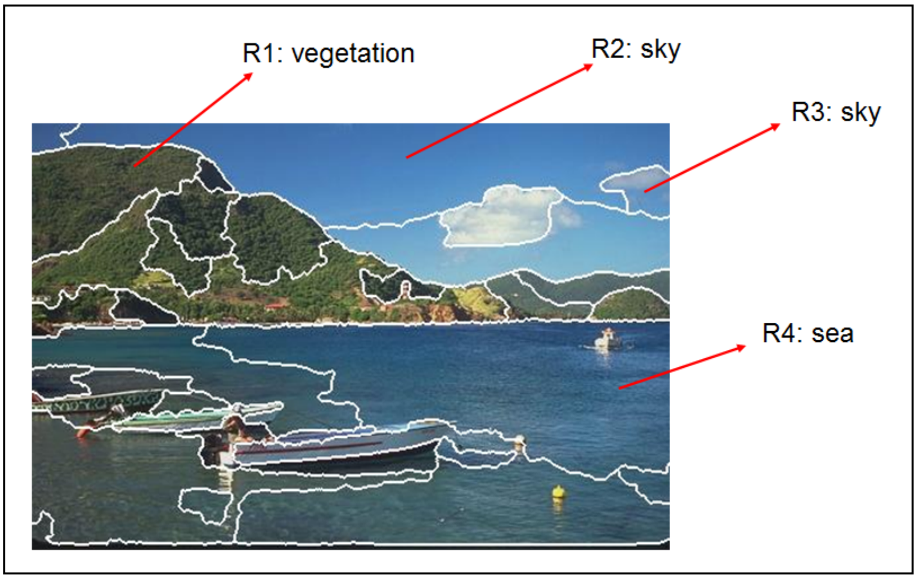


Figure 5.3 A sample image and region annotations.

As can be seen in the classification results, the classifier of *Sky* produces positive results for both of the regions: *R2* and *R3*. However, the classification probabilities or the confidence in each decision are not the same; the region *R1* produces higher result than the region *R2*, since the former region is closer to an ideal *Sky* than the latter. Also the results of negative classes differ, as can be seen in the classification results of the regions *R1* and *R4* in Table 5.1.

Table 5.1: The classification results and probability estimates

Region	Classifier	Class	Probability
R1	Sky	-	0.93
R2	Sky	+	0.89
R3	Sky	+	0.57
R4	Sky	-	0.62

5.3 Determination of Feature Weights

When the evaluation results of individual concept classifiers are compared, their accuracies are not the same, which indicates that some low-level descriptors are more effective in learning certain concepts than the others. For example, the color-based descriptors are usually more discriminative for the concepts like *Sky*, *Snow*, or *Vegetation* than the region-based features. In order to exploit this information in the classifier combination task, the relative importance of the low-level features on different concepts should be investigated. We propose a probabilistic feature weighting approach using in the dissertation, which is used to model the quality of low-level features for semantic concepts based on information entropy measures.

The proposed approach is based on analyzing the distributions of the low-level features extracted from the annotated image regions in the training set. Similar to the membership calculation method introduced in Chapter 4, the same SOM structures are utilized in feature weighting task. As stated in the previous chapter, the SOMs are built independently for each concept and descriptor pair by exploiting the positive feature vectors of each semantic class in train set. The basic idea in applying SOM mapping during the weighting process is to examine the distributions of the low-level features over semantic concepts, and analyze the randomness of feature distributions on different SOM output layers. The descriptors whose distributions are relatively non uniform are observed to produce more successful classification performances than the randomly distributed descriptors, and higher weights are given to non uniform feature distribution in the feature weighting process accordingly.

The SOM mapping performed in feature weighting process differs from the mapping applied in the membership detection task in that in the latter we map all of the samples in the training set, i.e. both the positive and negative feature vector for a concept, while in the former we map only the positive feature vectors, and perform the computations after the whole mapping is completed. In the implementation of weighting method, each training vector's *bmu* is located on the SOM, and the total numbers of *bmu* hits are counted to form the *bmu* hit histograms. These histograms are used as the posterior probability of a neuron to be the *bmu* for a new vector, and the feature weights are determined by calculating the entropy of this probability distribution during the feature weighting process.

The more uniformly or randomly a concept distributes a low-level feature the less distinctive the feature for the concept and higher entropy value is produced in this situation. After the entropy values are determined for different descriptors; the basic assumption in feature weighting is that lower entropy corresponds to better performance, since it represents a non-random distribution.

The steps included in the feature weighting are shown in Figure 5.4. After the SOM mapping is finished, we apply an initial clustering algorithm on the SOM neurons considering the topology preserving property of the SOM. Then we start the calculation of entropy in feature distributions, to identify the uniformity of the distribution as mentioned before.

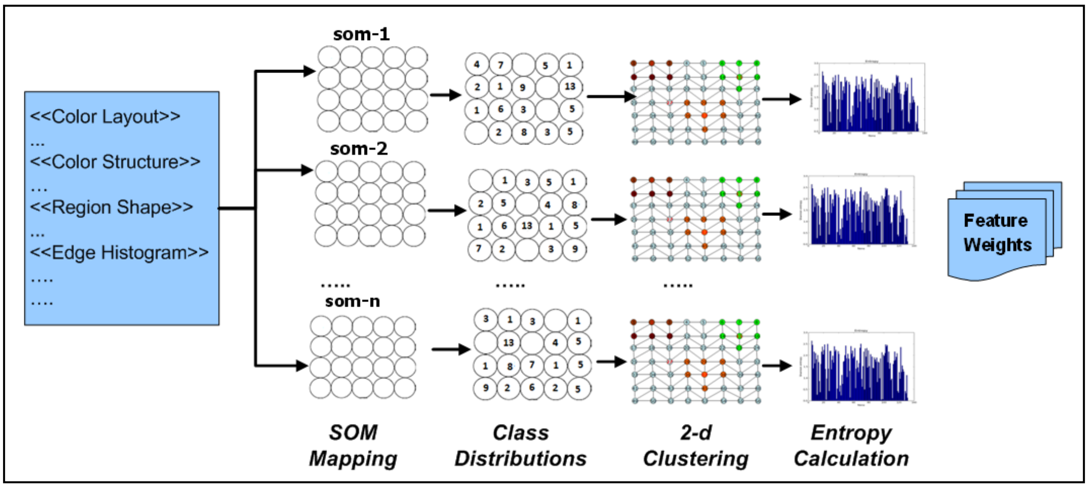


Figure 5.4 The feature weighting process.

5.3.1 SOM Mapping and 2-d Clustering

The neighboring SOM units have closer weights and the SOM maps the input vectors such that the similar patterns are mapped to the neighbor units on SOM. To exploit the topology information provided by the SOM, we apply a clustering algorithm on the initial hits of the neurons.

As described in Chapter 4, we train individual SOMs for each semantic class using only the positive samples for each concept. For each low-level feature in training set, we first map them on corresponding SOMs and obtain a mapping of the low-level features for semantic concepts, and then count the *bmu* hits after the mapping. Algorithm 5.1 presents the SOM mapping and clustering method applied in feature weighting process. We provide a sample scenario to clarify the inner steps of Algorithm 5.1 in this section.

Algorithm 5.1 SOM Clustering.

```
Input : An initial SOM structure
Output : A number of clusters and bmu hits.
1: Construct SOM from the train data
      $sM = som\_make(sD)$
2: Locate the bmu's for each training sample and calculate the SOM distances.
      $[bmus, qerr] = som\_bmus(sM, sD)$
3: Form the final bmu hits.
      $[hits] = som\_hits(sM, sD)$
4:  Normalize the bmu hits.
```

```
    $[hits] = normalize\ [hits]$
            ,where SOM_distance $\le$ $SOM_{avg\_qerr}$ for each hit.
5: Perform clustering on the SOM
    $for\ i = 1\ to\ n$
    $\forall x \in sM[],$
        $agg\_hit_x = hit_x + \sum hit_y\ ,\ where\ neigh(x,y) = 1$
        $sort\ neurons\ by\ descending\ agg\_hits,$
        $cluster\_center = index(1),$
        $if\ \ hit(cluster\_center) \le hit\_threshold\ \ stop$
        $else$
            $Cluster_i = [cl\_center, neighbours(cluster\_center)]$
            $\forall x \in Cluster_i\ ,\ hit_x = 0$
6: Merge neighbor clusters
    $if\ Cluster_i \cap Cluster_j \neq \varnothing$
            $remove\ Cluster_i,\ Cluster_j$
            $add\ Cluster_i \cup Cluster_j$
```

Figure 5.5 presents a sample SOM of size 7x8, which is trained using a small dataset. The circles represent the SOM neurons on the map, and the numbers in each circle represents the id of the neuron.

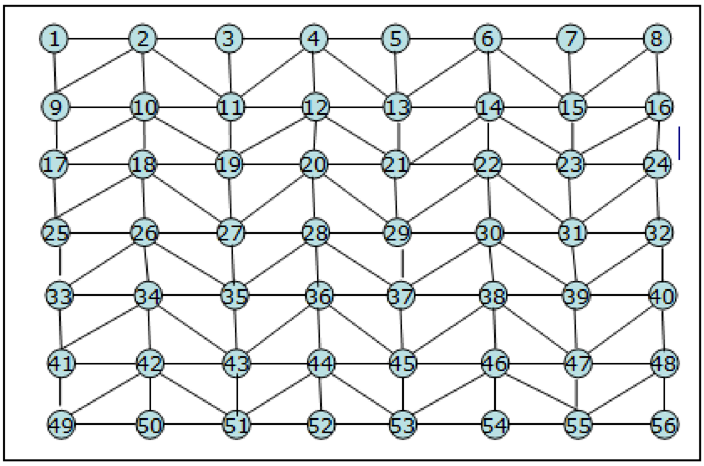


Figure 5.5 A sample SOM of size 7x8.

Figure 5.6 shows the *bmu* hits after the positive samples in training set are mapped on the same SOM of Figure 5.5. The numbers above the neurons represent the total hit count of each unit.

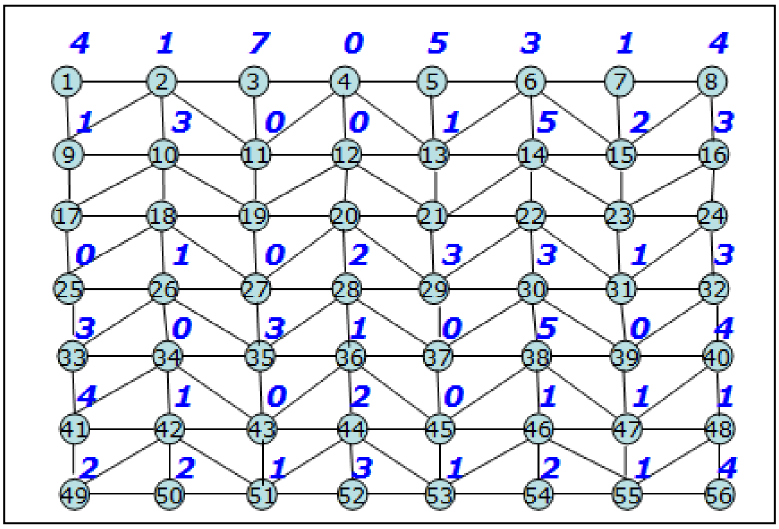


Figure 5.6 The bmu hits on the SOM.

We initially perform normalization on the initial *bmu* hits to eliminate the noisy hits on the initial hit histogram. We find the average distance between the input vectors and SOM neurons, and the hits that have bigger values than the average distance are deleted. Figure 5.7 shows the *bmu* hits after the normalization, the changes are presented in red color on the map.

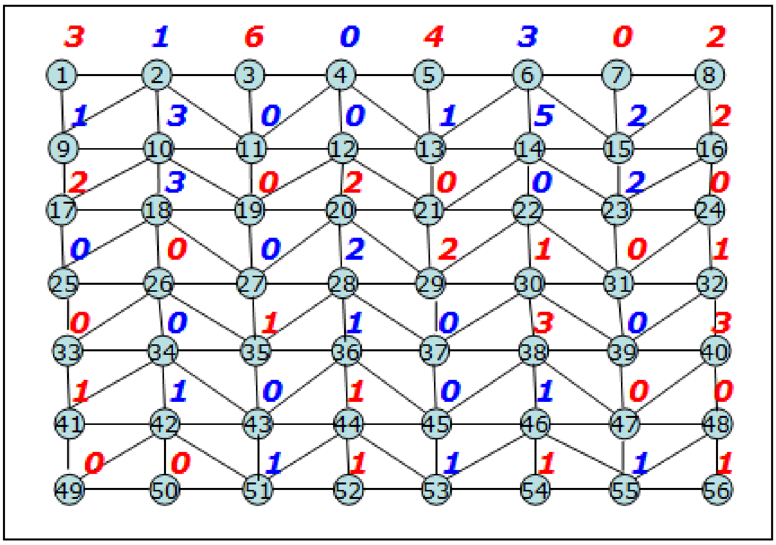


Figure 5.7 The bmu hits after normalization

Then we aggregate the hits of each SOM unit with its neighbors. The results are shown in Figure 5.8. At this step we sort all the aggregated hits and chose the largest unit as a cluster center. As can be seen in Figure 5.8, the unit 15 has the largest total hit and thus selected as the center of first cluster in the iteration of line 5 of the Algorithm 5.1.

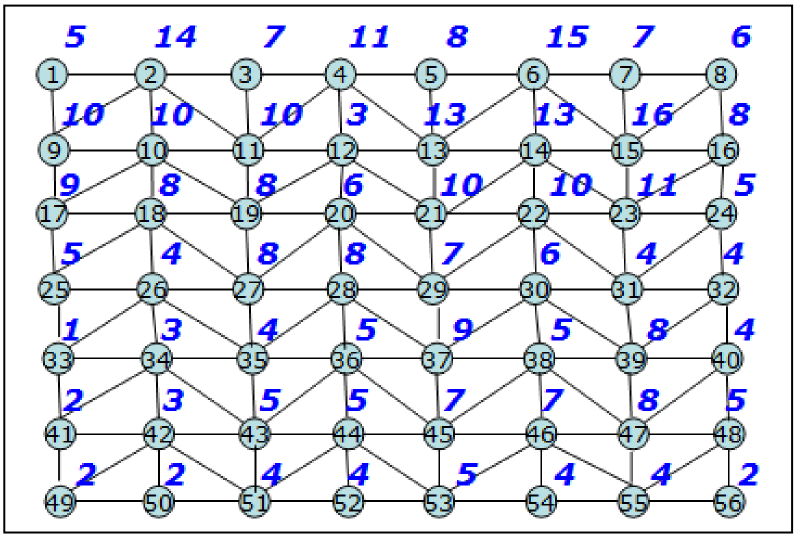


Figure 5.8 The aggregated *bmu* hits

After determining the cluster center, we collect each neighboring unit in the cluster and change corresponding hit cunts to 0. The first cluster shown in Figure 5.8 includes the following neurons: (15, 6, 7, 8, 14, 16, 23), and after the first iteration is finished in the Algorithm 5.1, the new *bmu* hits are formed as shown in Figure 5.9.

When clusters are formed, we apply a merging process on the clusters such that two clusters with at least one common unit are merged. The final form of the clusters found by applying our 2-d clustering is shown in Figure 5.10.

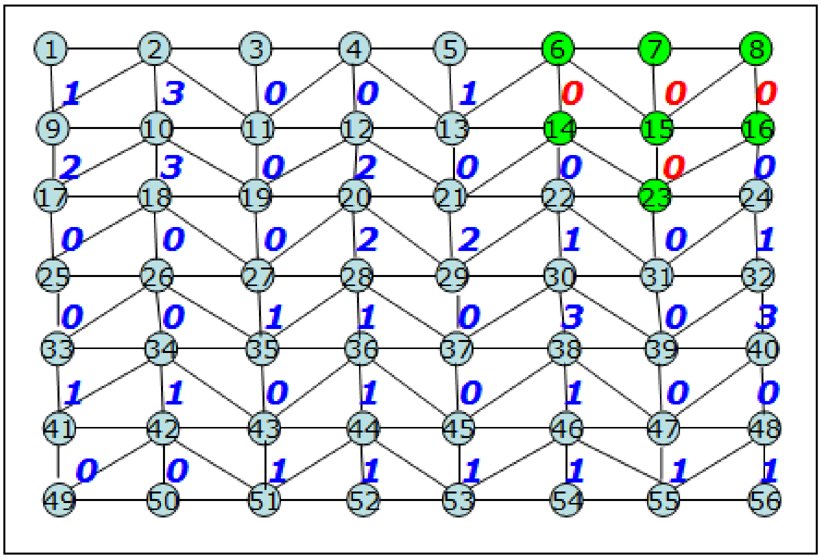


Figure 5.9 The first cluster and updated hits

When clusters are formed, we apply a merging process on the clusters such that two clusters with at least one common unit are merged. The final form of the clusters found by applying our 2-d clustering is shown in Figure 5.10.

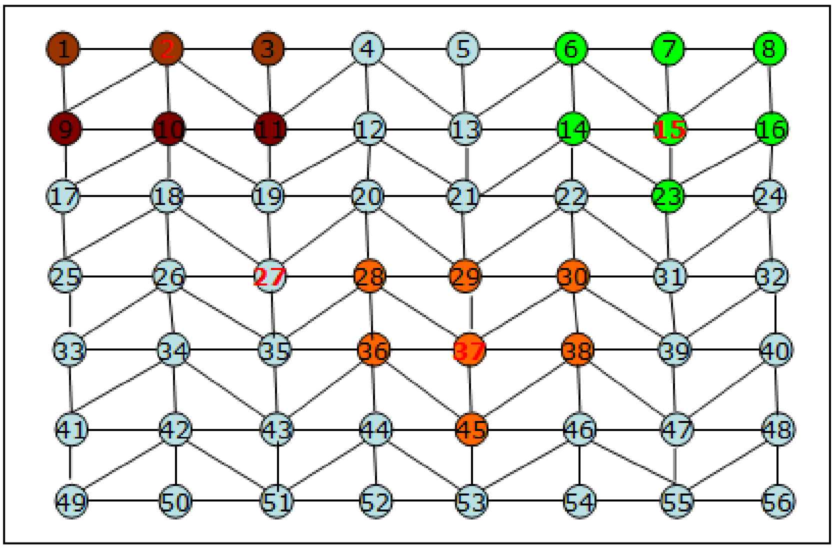


Figure 5.10 Final clusters.

5.3.2 Entropy Calculation and Weight Formulation

Entropy is defined differently in different contexts [99]. Throughout the dissertation, the term entropy is used as a metric of information measure, which is defined in the Shannon's Information Theory [100].

The entropy of a random variable is defined in terms of its probability distribution and can be used as a measure of randomness or uncertainty. Given a probability distribution, the entropy is defined as follows:

$$H(p_1, p_2, ..., p_n) = \sum_{i=1}^{n} p_i \log(\frac{1}{p_i})), \ where\ p_1 + p_2 + ..., p_n = 1 \tag{5.5}$$

Entropy is a commonly used measure to determine the randomness of a distribution. A good model is such that the distribution is concentrated on a few clusters, which result low entropy value [57]. The authors examines different clustering methods for multimedia concepts and the interpretations of in terms of the probability distributions, and evaluate the models using a similar method proposed in the dissertation by applying entropy-based calculations.

In our situation, we treat the SOM neurons as cluster centers, and the *bmu* hits as the probability density function, which represents the posterior probability of each cluster being the best match for any vector in the training set. Hence, after determining the *bmu* hits and performing the clustering Algorithm 5.1 on the map, we follow the following steps to calculate the feature weights:

1) *Perplexity Measure:* Perplexity is a more illustrative measure than entropy, which is commonly used in speech recognition, and defined as: $PPL = 2^H$.

Instead of making calculations based on entropy measure, we use the perplexity PPL in weight calculations. The maximum entropy (H_{max}) is achieved when the distribution is uniform, i.e. similar *bmu* hits. We first calculate the normalized perplexity of the *bmu* distribution as follows:

$$\overline{PPL} = \frac{\mathrm{PPL}}{PPL_{max}} = \frac{2^H}{2^{H_{max}}} \tag{5.6}$$

, which is nonnegative and less than or equal to 1, in all cases.

2) *Feature weighting formula:* A small value of $\overline{PPL}$ corresponds to a more concentrated, or non-uniform distribution; and hence the relative weight of the corresponding feature should be increased. To represent the weight, we apply a Gaussian function on the normalized Perplexity ($\overline{PPL}$) value, and set the weight of the i_{th} feature for the j_{th} concept as $w_{i,j}$:

$$w_{i,j} = e^{-\frac{\overline{PPL}^2}{2*\sigma_j^2}} \tag{5.7}$$

, where σ_j is the radius of the SOM that is trained on the j_{th} concept.

The magnitude of the entropy reflects the effectiveness of the feature for the concept, and the output generated by a classifier that is trained on that feature is treated accordingly to the weight of that low-level feature in the fusion process, as described in Section 5.1.

5.4 Application of the DS Combination in Classifier Fusion

In this section we explain the application of DS Theory in the fusion phase of the concept detection system. As described previously in this chapter, we introduce a new technique in high-level classifier fusion through introducing the DS Theory in concept detection process.

We illustrate a sample DS combination in this section. Table 5.2 shows the results of two classifiers on three images including the probabilities in each case. Before explaining the DS combination rule for this example, we first describe the fields in Table 5.2:

- The classification results, which denote the evidence provided by each classifier, and used in combining multiple evidences accordingly,
- The classifier confidences represent the basic probabilities assignments (e.g. mass functions) assigned to evidences,
- The weights are used as the trust level of classifiers in the weighted DS combination.

These values are fed into the weighted Dempster-Shafer combination applying the Equation (5.2).

Table 5.2: Single classification decisions for a test image and feature weights.

Classifier	Decision	Classification Conf.	Feature Weight
svm1	+	0.53	0.18
svm2	-	0.89	0.12
svm3	+	0.71	0.39
svm4	-	0.94	0.31

Considering the evaluations in Table 5.2:

$$m_1(+) = 0.53,\ m_1(-) = 0.47 \tag{5.8}$$
$$m_2(+) = 0.11,\ m_2(-) = 0.89$$
$$m_3(+) = 0.71,\ m_3(+) = 0.29$$
$$m_4(+) = 0.06,\ m_4(-) = 0.94$$

We combine the evidences of positive classifiers by applying Equation (5.1):

$$m_{13}(+) = 0.53 \times 0.71 = 0.376 \tag{5.9}$$
$$m_{13}(\varnothing) = 0.53 \times 0.29 + 0.46 \times 0.71 = 0.48$$

Since there is conflict in the combination, we need to normalize the computed values by $1 - m_{13}(\varnothing)$, which gives a combined mass of positive class $m_{13}(+) = 0.72$.

Similarly, by applying Equation (5.1) to combine the *bpa* of negative class:

$$m_{24}(-) = 0.89 \times 0.94 = 0.837 \tag{5.10}$$
$$m_{24}(\varnothing) = 0.89 \times 0.06 + 0.11 \times 0.94 = 0.157$$

Similarly, due to the conflict in the combination, we apply the normalization to finalize the combined decision on negative class, which results the combined mass $m_{24}(-) = 0.99$. Since the total belief in negative class is higher than positive class, the image is classified as negative by the DS combination.

CHAPTER 6

AN EFFECTIVE BOW MODEL IN IMAGE CLASSIFICATION

In this chapter, we present our work on SIFT method along with the utilization of BOW model in image classification. We introduce the use of SOM and distinctive SIFT features in BOW. There are two major contributions in this: First, we improve the classification performance by establishing a unique feature selection method along with the utilization of SOM neural network in codebook generation phase of the BOW model. The SOM maps high-dimensional SIFT features on 2-D output space in a topology-preserving way, which builds more representative codebooks than typical clustering methods, such as the k-Means clustering method. Second, we improve the classification performance and increase the scalability by the use of distinctive features, i.e. reducing the number of image features by using only the distinctive features. To the best of our knowledge, the utilization of SOM both in codebook generation and in distinctive feature detection have not been investigated in image classification before, although several BOW models have been proposed for classification of visual data in the literature [101-103].

A number of low-level features have been proposed both locally and globally to effectively represent the visual properties of an image. Among them, SIFT [104] is a popular local feature extraction method with successful applications in different fields of image analysis [7, 105-107]. The main advantage of the SIFT [104] and its variants [108, 109], is the stability of the extracted descriptors under different orientation and scale changes, which enables reliable matching between the local image points extracted from the same objects or scenes. However, there are two main limitations in effectively utilizing the SIFT features:

1) The SIFT generates a large number of local features, and for most of the real-world problems it becomes infeasible to find a correct match to a single point against a large database of features [110],
2) The SIFT does not differentiate the keypoints in interested regions, such as object boundaries, from the points in the background, and most of the keypoints arise in the background locations [8].

The first problem is related to the SIFT method in which it locates several hundreds of keypoints from a single image during the feature extraction process, which results in a large number of features to be analyzed even for small training sets. For example, an average of 1500 keypoints is populated from a single image, and if we consider a training set of 200 images, then it results in 300.000 SIFTS features to be processed in computations.

Many real-world problems require the analysis of large image sets [26, 111], which makes the SIFT method infeasible due to the high computational costs. The common approach to solve this problem is to utilize the bag-of-words (BOW) model, which is widely used in document retrieval problems [106].

The second problem is about the determination of significant keypoints. Although the SIFT method locates scale-invariant keypoints in an image, it does not provide any information about the importance of a keypoint in the classification problem [109]. Moreover, most of the keypoints arise from background regions (Fig. 1) and inversely affect the classification problem. Hence, it would be useful to eliminate the irrelevant keypoints and utilize the remaining keypoints in the classification process.

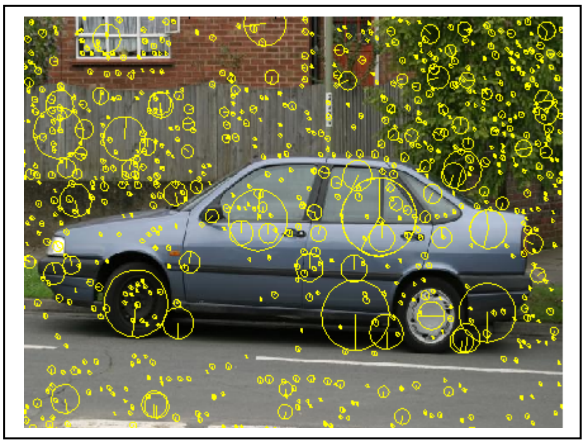


Figure 6.1 A sample image and SIFT keypoints.

The rest of the chapter is organized as follows: Section 6.1 introduces the background information related to SIFT method and present previous works in BOW model. Section 6.2 presents information about the architecture of the BOW model we developed, and the use of SOMs in codebook generation, the classification methods and the extraction of distinctive features. Section 6.3 presents the effect of threshold in distinctive feature detection, , and finally, Section 6.4 discuss the effect of different codebook sizes generated by SOM and k-Means clustering methods.

6.1 Background Information

6.1.1 Scale-Invariant Feature Transform

The SIFT was initially proposed by Lowe in 1999 as a local feature extraction method [104]. The SIFT method first detects the stable image points under different location and scale changes, and then generates the local features around these points.

The steps performed in SIFT feature extraction are as follows:

1) *Scale-space extrema detection*: The image is searched for all scales and locations, and a Difference-of-Gaussian (DoG) function is applied to identify the scale invariant locations

in the image. The scale space of an image is defined by the convolution function in Equation 6.1.

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \qquad (6.1)$$

, where $G(x, y, \sigma)$ is a variable-scale Gaussian function (Equation 6.2).

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \qquad (6.2)$$

The local extrema is detected by comparing each pixel with its neighbors by applying the following DoG function in scale-space (Equation 6.3).

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \qquad (6.3)$$

2) *Keypoint localization*: Each candidate keypoint is examined for stability through a quadratic fitting function, and the points that have low contrast or locate on the edges are discarded.

3) *Orientation assignment*: Once the keypoints are determined, one or more orientations are assigned to them based on their image gradients. The further computations are performed on this assignment for each key-point, which provides invariance to scale and orientation changes.

4) *Descriptor Representation*: The SIFT descriptor is formed by computing the local image gradients at each key-point location and finally a 128 bin descriptor is created [8].

The features extracted by SIFT are highly robust to occlusion, clutter and slight view changes, which give distinctiveness in the sense that a new feature can be matched with high probability against a large number of features [8].

6.1.2 The Classification Methods used in BOW

In this section, we provide the basics of the classification techniques employed in the paper. The utilization of each method for-based image classification system is explained in Section 6.2 in detail.

1) *The Naive Bayesian Classifier*: The Naive Bayesian (NB) classifier is a probabilistic learning method based on Bayes Theorem and maximum-posteriori hypothesis [112].

Formally, suppose that we have k classes, $(C_1, C_2, ..., C_k)$ and a sample vector $X = \{x_1, ..., x_n\}$, with n attributes. The NB classifier assigns a sample to the class having the highest posterior probability, i.e. the sample X is assigned to class C_i if and only if the following holds:

$$P(C_i \mid X) > P(C_j \mid X), \quad 1 \leq j \leq k, j \neq i \tag{6.4}$$

If we apply Bayes theorem to Equation 6.4:

$$P(C_i \mid X) = \frac{P(X \mid C_i)\, P(C_i)}{P(X)} \tag{6.5}$$

The denominator, $P(X)$ in Equation 6.5 is the same for all classes, and the class a-prior probabilities, $P(C_i)$ are unknown and assumed to be identical in the evaluation phase. Hence, the only remaining parameter that needs to be maximized in Equation 6.5 is $P(X \mid C_i)$, which is the posterior probability of the attributes given a class.

Since the attributes are conditionally independent of each other, given a class (e.g. the naïve assumption), $P(X \mid C_i)$ can be reformulated as:

$$P(X \mid C_i) \approx \prod_{k=1}^{n} P(x_k \mid C_i) \tag{6.6}$$

Finally, the NB assigns a test sample applying the maximum a posterior rule:

$$C_{map} = \text{argmax} \prod_{k=1}^{n} P(x_k \mid C_i) \tag{6.7}$$

2) *The KNN Classifier*: The k-Nearest Neighbor (KNN) is a supervised classification method that simply stores the training samples in its learning phase. During the evaluation of a new sample, the KNN first computes the similarity between the sample to its k-nearest neighbors from training data, and then the sample is assigned to the class that occurs most frequently in the k-nearest neighbors [113]. Although, the KNN is a simple classification method without a model construction or parameter estimation phases, its performance highly depends on the local information and sensitive to noise in the training data.

3) *The SVM Classifier:* The SVM is a state-of-art classification technique with extensive applications in many real-world problems [21]. During the learning phase, the SVM searches for a separating hyperplane with the largest margin between the positive and negative samples in the training set. If the samples cannot be separated in the input space, the SVM maps the input samples to higher dimensional spaces and searches the optimal hyperplane in these spaces [21].

6.1.3 Related Work

This section provides a brief overview of the existing BOW approaches in the context of image analysis. The BOW model was first introduced to visual data in 2003 [114], and since then it has been used in different fields of image analysis including image classification, object detection, content based retrieval and so on.

There are three major tasks in BOW approach [107]: 1) keypoint detection and feature extraction, 2) codebook generation, and 3) image representation. One of the most common local interest point detection technique is Difference-of-Gaussian (DoG) [8], which is also used in SIFT method to locate the scale-invariant keypoints in an image. The SIFT method is the most widely used feature extraction algorithm in BOW [115].

The general approach in codebook generation is k-Means clustering [7], and several variations of the k-Means algorithm are explored to address the limitations of the original method [116, 117]. The size of the codebooks is also examined in the BOW model, and different numbers of visual-words are studied in the literature. Typically codebooks of 1000-words are recommended as a reasonable dictionary size in image analysis applications [102].

In the literature, there are several works to build more informative codebooks in the BOW approach [103]. For example, a general framework is proposed in [118] to improve the descriptive power of visual vocabulary for large-scale image applications, and in [119] the class labels of images are exploited during the vocabulary construction process to form more representative visual words.

Another BOW framework is presented in [120] considering the spatial relations between the words and introducing so called "visual sentences" that allows ordered reading of the visual words as in the case of text. In a recent work [121], the use of BOW model along with weighted visual features is presented so that each image feature is associated to a weight that shows the relevance of the feature to an image during the image representation phase of the BOW model.

Although the BOW model is adapted from the text retrieval domain, and many applications are successfully implemented in different fields of image analysis, there are major difficulties in exploiting BOW to its full potential in image domain. For instance, one difference between the textual and visual words is that the similarity of the visual-words cannot be easily determined. For example, the words "*hard*", "*harder*", and "*hardly*" have all the same root, and can be represented by a single word in text retrieval applications. However, there is not such an analogy for visual-words, and hence the elimination of irrelevant features is important in effectively applying the BOW model to images.

Another difficulty is the size and complexity of visual features are much more than those of text documents, and applying a clustering method on the visual features generally results in loosing the descriptiveness of the visual-words [118]. This problem can be solved by building large codebooks, but this is limited by computational the performance of clustering methods. Hence, obtaining a reduced set of image features, without losing information, is crucial in effectively applying the BOW model to image analysis, which is utilized in this study by the use of distinctive features.

6.2 System Architecture

Our system architecture is composed of the training and test phases and described below:

1) The training phase of our classification system is shown in Fig. 2. There are three main steps in the training phase: Feature Extraction, Codebook Generation and Feature Mapping. In feature extraction step, the SIFT method is applied to the images of training set and the 128-D SIFT descriptors are extracted. During the codebook generation, one codebook is formed for each image class individually by training a separate SOM as can be seen in Fig. 6.2. In this representation, each SOM unit represents a visual-word of the codebook. The last step in the training phase is the feature mapping, where we map the entire image features of each class to its codebook.

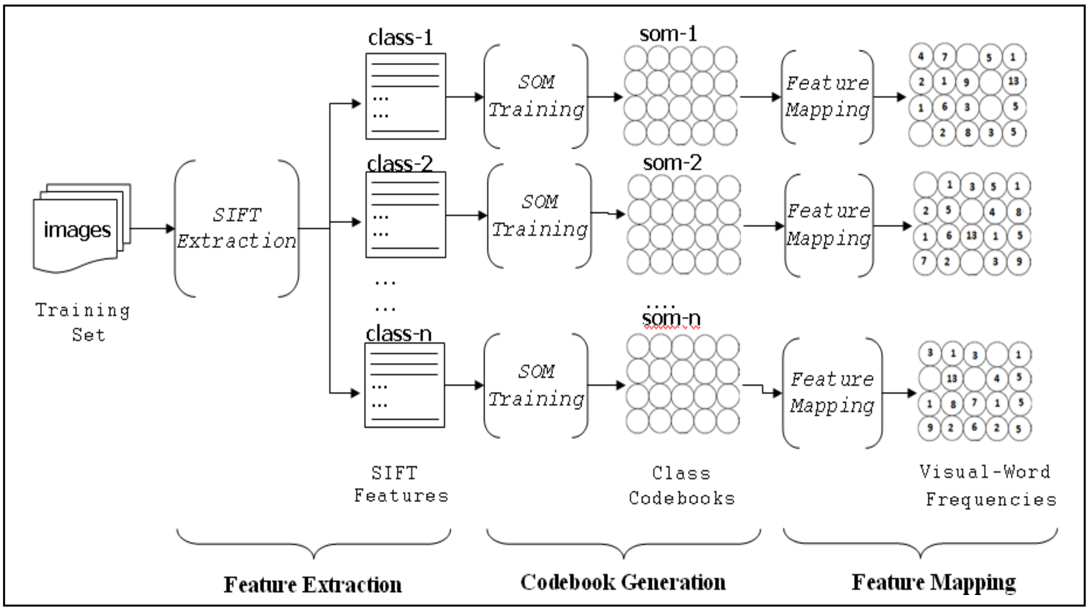


Figure 6.2 The training steps of the classification system.

2) The test phase shown in Fig. 3 also consists of three steps, Feature Extraction, Distinctive Feature Detection, and Classification. First, a similar feature extraction is performed on test images except that the extraction is performed to the whole image in the test phase and the extraction of object boundaries is done in the training phase. Second, we perform the distinctive feature detection method to determine the significant keypoints in images by analyzing the distances of local keypoints to different codebooks. Third, we perform three different classification methods, NB, KNN and SVM, to assign the class labels of unknown images using the distinctive features.

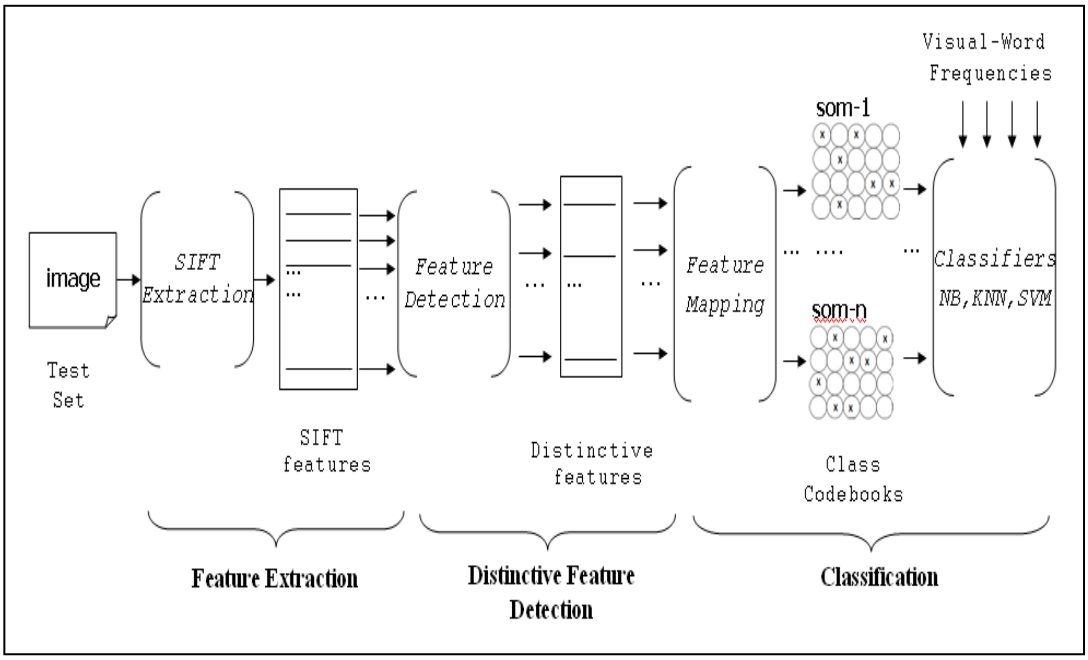


Figure 6.3 The test steps of the classification system.

6.2.1 Codebook Generation and Feature Mapping

The codebook generation is one of the most important steps in the BOW model. The predominant approach in codebook generation is k-Means clustering [7]. However, there are some problems in the k-Means method; the cluster centers are set in high density regions, the outliers can significantly affect the cluster centers, and so on. We select to use the SOM clustering in the generation of class codebooks since it provides a topology-preserving mapping of the input vectors, where the vectors that are close in the input space are mapped to closer locations on the SOM output space [122].

Figure 6.4 shows a more detailed view of the codebook generation process shown in Fig. 2. Each SOM unit is assigned a 128-bin weight vector and represents a visual-word for the class.

The size of the codebook is determined using Equation 6.8.

$$S = 5 \times \sqrt{K} \tag{6.8}$$

, where K is the size of the descriptors used in SOM training.

Once the class codebooks are generated, the SIFT features of each class is mapped to the corresponding SOM structure in the feature mapping step as shown in Figure 6.2. This mapping provides two computations:

- *Visual-word frequencies*: The count of each visual-word selected as bmu forms the visual-word frequencies, which are used as the probability estimates of the visual-words during the Bayesian classification.
- *Image Representations*: All training images are represented in terms of visual-word histograms, and then used in SVM training.

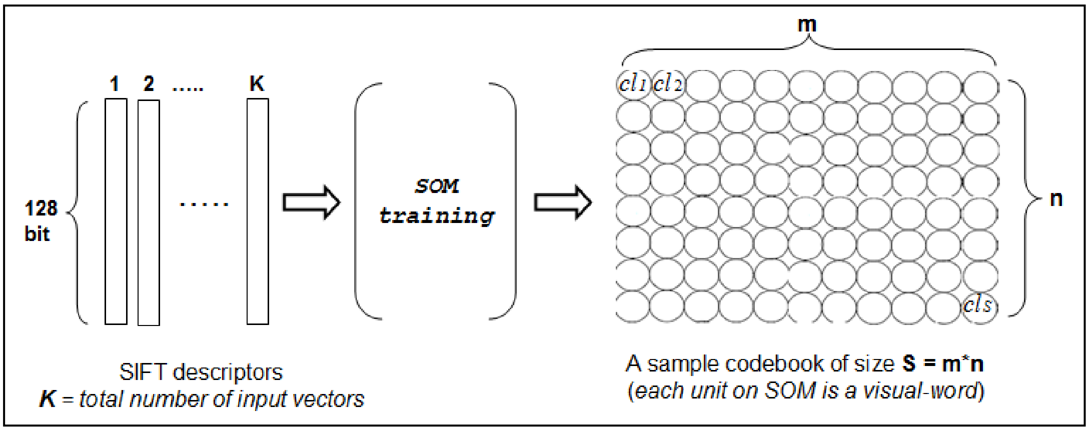

1
2
.....
K
128 bit
.....
SOM training
m
cl_1
cl_2
cl_S
n
SIFT descriptors
K = total number of input vectors
A sample codebook of size **S = m*n**
(*each unit on SOM is a visual-word*)

Figure 6.4 The codebook generation and a sample SOM.

6.2.2 Our BOW Approach for Image Classification

In order to classify a new image, we first map the local features of the image to the class codebooks by finding the closest visual-words from each codebook. Then, three different classification techniques are performed on test images as shown in Fig. 3.

1) *NB Classifier:* Before applying the NB classifier, we need to identify the attributes and classes along with the determination of the class conditional probabilities as described in Section II. In our classification framework, the SIFT features extracted from a test image form the attribute set $\{x_1,....,x_n\}$, and the codebooks represent the classes $(C_1,C_2,...,C_k)$.

The naive assumption holds since the SIFT features are conditionally independent from each other for each image class. The only parameter that needs to be maximized in (5) is $P(X \mid C_i)$, which can be restated as the probability of the distribution of the SIFT features on class codebooks. This probability is computed from the visual-word frequencies as described in the previous section.

2) *KNN Classifier*: The KNN classifier does not build a model in the training phase, and it only requires a distance matrix. This matrix is formed by mapping the local features of a test image to all codebooks as shown in Fig. 3.

More specifically, we first map the features of test images to all SOMs, and then normalize the distance between a keypoint and visual-words using the average quantization error of each SOM. This normalization is necessary since the codebooks are built separately using a different number of features in the training phase. Once the distance matrix of a test image is computed, the KNN evaluates the matrix and assigns the most common class to the test image.

3) *SVM Classifier:* The SVM classifier first builds a learning model using the representation of images in the training set, and then applies this model in classifying test images. We build binary SVMs using linear kernels under the "one-against-all" approach [89]. At the end, one SVM is formed for each image class capable of classifying whether a test image contains an instance of a class or not

6.2.3 Extraction and Utilization of Distinctive Features in Image Classification

Lowe in [8] shows that the SIFT features are highly robust in finding a correct match to a new keypoint against a large database of features that includes the features extracted from the images containing the same objects or scenes. Moreover, the incorrect matches originated from the background or cluttered image regions can be filtered out by applying a threshold on the ratio of the distances between SIFT key-points. A threshold value of 0.8 eliminates approximately 95% of incorrect matches while discarding only 5% of correct ones [8]. But this method cannot be used directly in the BOW model, since the SIFT features are represented by the visual-words in codebooks. Hence, a new way of detecting the distinctive features is required for the BOW model to eliminate the keypoints in background regions.

Inspired by the distinctive keypoint matching approach in [8], we design a new distinctive feature detection method exploiting the distances of local features to different codebooks of image classes. Instead of setting up a global threshold, we compare the distance of a feature to the first- and second-closest visual-words from two distinct codebooks, and retain a keypoint only if the ratio of the two distances is smaller than a predetermined threshold value.

The distinctive feature detection method is presented in Algorithm 1. Given a set of local features and a number SOMs representing class codebooks, first each feature is mapped to every codebook and the first- and second-closest visual-words from two distinct codebooks are found. Then, the distance between the feature and these words are compared: If the ratio of the distances is less than a threshold value, the feature is identified as distinctive, otherwise it is assumed to be a background point and discarded.

The proposed method performs well in identifying the significant keypoints in an image since the visual codebooks are formed by training a separate SOM using the features of multiple images of the same class individually. For a local image feature, the point is either selected as distinctive or eliminated as follows:

- *A distinctive feature* would have remarkably smaller distance to the first closest visual-word than the second closest visual-word from a different codebook, since the first codebook is formed using the features of the same objects, and the match is correct. On the other hand, the second codebook is a formed using the features of different objects, which results in an incorrect match of the input feature.

- *A background feature* may produce a number of false matches with similar distances, since both codebook matches are incorrect, and hence produces similar distances in Algorithm 1.

Algorithm 6.1: Distinctive feature detection method.

```
Input: Set of codebooks $\{C_1,C_2.....,C_N\}$ for N image classes, set of SIFT descriptors $\{d_1,d_2.....,d_K\}$ extracted from a test image.
Output: Set of distinctive image features.
1: For each local descriptor $d_j \in \{d_1,d_2.....,d_K\}$, calculate the distance of $d_j$ to all class codebooks.
   $dist_{i,j} = \text{SOM\_Map}(C_i, d_j)$
2: Locate the first- and second-closest visual-words to $d_j$ from two different codebooks.
   $v\text{-}word_1 = \text{index}(\min(dist_{i,j}))$, where $v\text{-}word_1 \in C_m$
   $v\text{-}word_2 = \text{index}(\min(dist_{i,j}))$, where $v\text{-}word_2 \in C_n$ and $C_m \neq C_n$.
3: Select $d_j$ as distinctive if the distance ratio to the first- and second-closest codebooks is under the threshold value of the closest class.
   if $(dist_{m,j} / dist_{n,j}) \leq threshold\text{-}C_m$, $d_j$ is distinctive.
   else discard $d_j$.
```

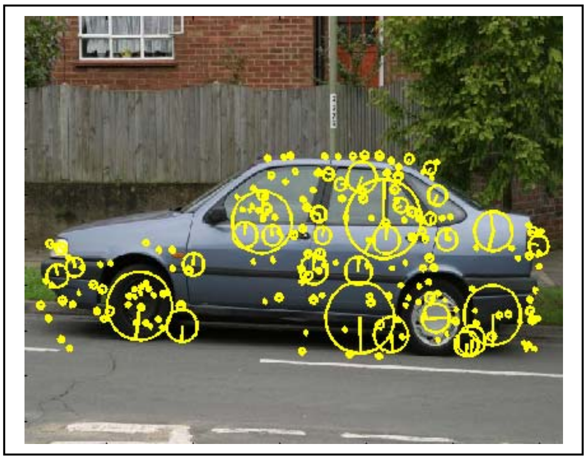


Figure 6.5 The distinctive keypoints of Figure.6.1.

Figure 6.5 shows the keypoints after applying the distinctive feature detection method to Figure 6.1.

6.3 The Effect of Threshold in Distinctive Feature Detection

The success of the distinctive feature detection method depends on the selection of an optimal threshold value, since large threshold values result in many background points to be selected and small threshold values cause to eliminate some significant keypoints. Therefore, finding an optimal threshold value for each class is crucial in computing the distinctive image features.

In order to investigate the effect of the threshold value in distinctive feature detection, we perform several experiments using KNN in classifying *car* objects. The true positive (TP) and false positive (FP) counts for 961 *car* objects is given in Table II. We select the KNN classifier in these experiments since it is more sensitive to individual image features than NB and SVM are.

Table 6.1: The Effect of Threshold in Classification (TP and FP rates for *car* class).

Threshold	**Unclassified**	**Classified**	**3-NN**		**5-NN**		**7-NN**	
			TP	*FP*	*TP*	*FP*	*TP*	*FP*
0.80	505	456	419	37	427	29	439	17
0.85	293	668	597	71	617	51	622	46
0.88	**246**	**715**	**633**	**82**	**655**	**60**	**691**	**24**
0.95	127	834	563	271	674	160	693	141
1	0	961	387	574	463	498	521	440

In this experiment, we use the images containing car objects, and utilize all available codebooks generated previously in distinctive feature detection method (Algorithm 6.1). For each threshold value shown in Table II, we run Algorithm 6.1 and use the remaining features for the classification process. We use 3 different k values in the KNN classifier: 3, 5, and 7, respectively, and if an image has less than 3 distinctive keypoints, it is labeled as an unclassified image.

First, the threshold value of 0.8, which is proposed by Lowe [8] for an optimum keypoint matching, results in 53% of the test images, nearly all features are eliminated by Algorithm 1. The main reason in this result is that the computations in [8] are directly performed using SIFT features, and we compare the distance of keypoints to visual-words, which are not the exact SIFT features but their closest matches in codebooks.

Secondly, increasing the threshold values decreases the number of unclassified images, but also increases the TP and FP rates at the same time. The optimum threshold value for the *car* class is found 0.88, which results in a TP rate of 72% when k is 7. The optimum threshold values are found similarly prior to applying the distinctive feature detection method.

6.4 Codebooks Generated by SOM and k-Means

We also investigate the effect of SOM and k-Means methods in codebook generation with varying codebook sizes by running a number of experiments. The *k* in k-Means is chosen the same as the size of the SOM. Table III shows some statistics about the classes used in this experiment and the size of the codebooks calculated by applying Equation 6.8.

6.4.1 Average quantization errors of codebooks

Table 6.2 presents the average quantization errors of the codebooks generated by both methods. These are the average distances between the SIFT features to the codebooks for each image class, which are obtained by mapping all of the features in the training set to the corresponding codebooks. As can be seen in Table 6.2, the input features are closer to the cluster centers generated by SOM compared to k-Means, which results in less quantization errors. The reason behind this result is the application of the neighborhood function in SOM, which forces to update the cluster center along with its neighbors.

Table 6.2: Average quantization Errors in k-Means and SOM Clusters

Classes	k-Means	SOM
car	15.23	9.71
cat	17.91	9.83
cow	23.67	9.87
motorbike	27.15	9.97
person	19.28	9.88
Average	20.65	9.85

6.4.2 Effect of codebook size in classifications:

We run several experiments using all SIFT features and only distinctive features to investigate the effect of the size of the codebooks, and compare the SOM clustering with k-Means clustering in the context of our classification system.

First, we employ SVM classifier using all features, and present the result for classifying the *motorbike* class (Figure 6.6). Increasing the size of codebooks improves the SVM classification performance in both clustering methods. However, the improvement in SOM is higher than that in k-Means, and for the codebooks of size 500-words and less, both methods give similar performance results.

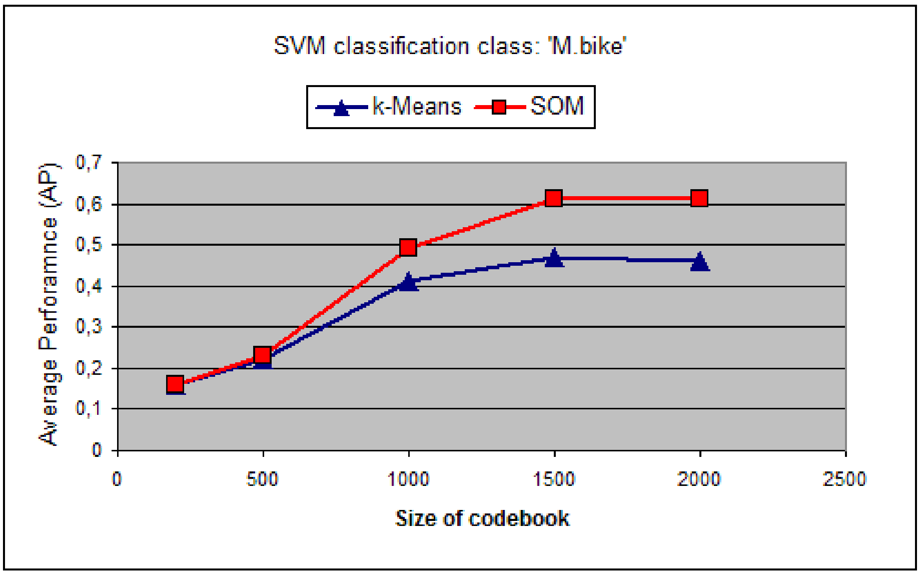


Figure 6.6 The effect of codebooks generated by k-Means and SOM in SVM classification.

Secondly, we evaluate the effect of different codebooks using the KNN classifier, with k = 5, using distinctive features. The result in classifying the person class is shown in Figure 6.7. The performance of KNN also increases with large codebooks, and similarly, both SOM and k-Means produce similar performance results for small codebooks.

Finally, we test the NB classifier using distinctive features to classify the car objects in Figure 6.8. Since NB uses the word-frequencies that are calculated in the training phase, this experiment gives more information on the effect of the codebooks in the classification process than the previous tests performed in this section. When average precisions are compared, a 1000-word codebook generated by the SOM performs better than a double-sized codebook formed by k-Means. Hence, the SOM method generates more informative clusters than the k-Means, which is one of the major contributions of this paper in image analysis.

To sum up, increasing the codebook size has a positive impact on the classification performance to some degree (e.g. 2000-words). The SOM builds more informative codebooks than k-Means does in general, and for small-sized codebooks, both methods perform similar results, and the impact of the clustering method is not so significant in the classification performance.

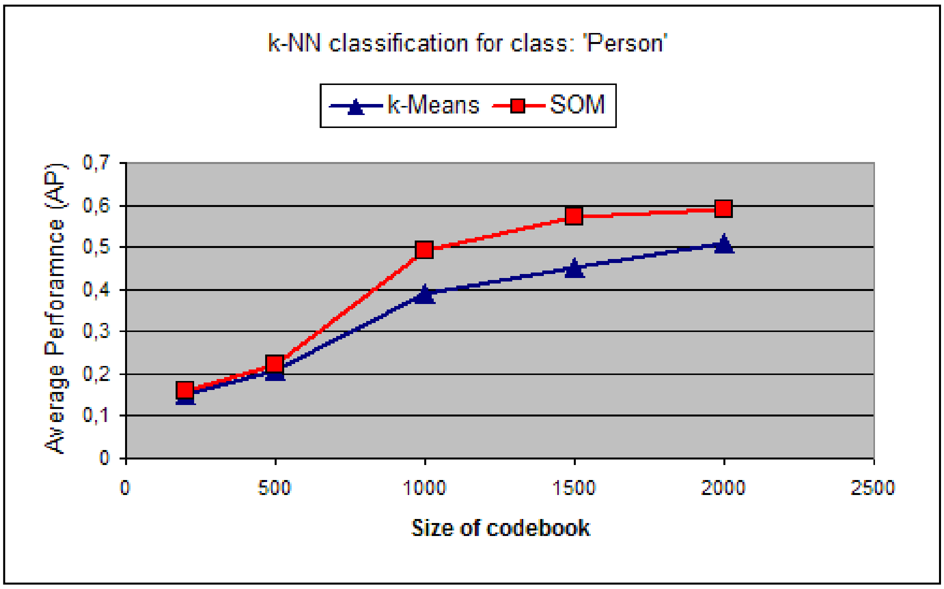


Figure 6.7 The effect of codebooks generated by k-Means and SOM in KNN classification.

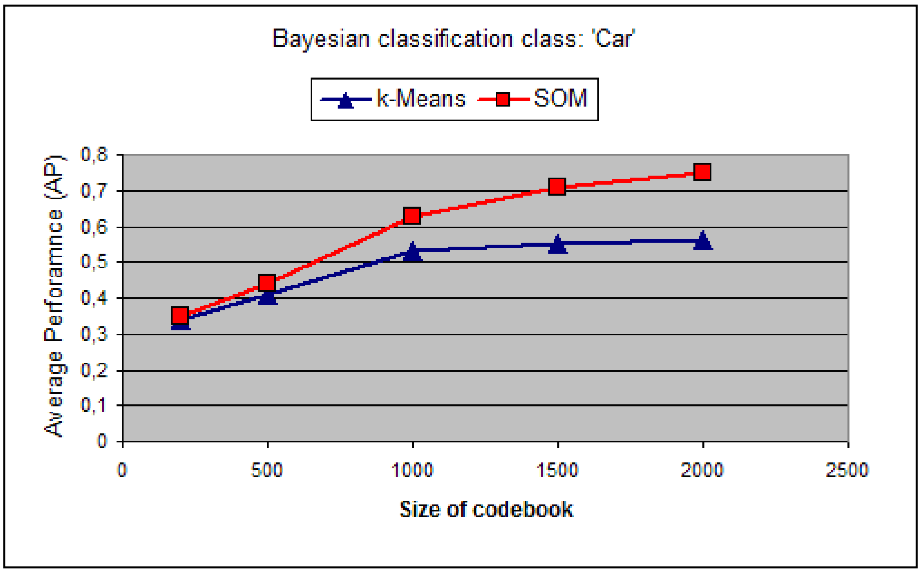


Figure 6.8 The effect of the codebooks generated by k-Means and SOM in NB classification.

CHAPTER 7

EXPERIMENTS

In this chapter, we present the results of the experiments conducted on benchmark datasets to demonstrate the effectiveness of the visual analysis system in high-level semantic extraction introduced throughout this dissertation. We also express major contributions achieved by the methods introduced by low-level feature modeling and its effects on classifier learning, feature weighting and fusion as described in previous chapters.

In order to give a better understanding of the potential applications and methods in bridging the semantic gap challenge in visual analysis, we performed several experiments on a benchmark image collection [123]. During the implementation of image classification system, we first run the training steps and build several SVM classifiers. Then, we perform several test sessions using different classifiers and evaluate them on test images. Finally, we perform the high-level classifier fusion approach to obtain the combined classifier decision.

We compare the results of the approaches applied in this dissertation and their effects on classifier performance results in several ways:

- First, we evaluate the performance of single classifiers trained by different MPEG-7 features. In this scheme, we run both classical and fuzzy SVM learning algorithms introduced in Chapter 4. This provides an initial demonstration of both the learning capacity of both SVMs along with the discrimination of MPEG-7 descriptors utilized in the dissertation.
- Secondly, after having all individual classifications, we try different fusion methods such as majority voting and weighted majority weighting along with the DS fusion introduced in Chapter 5. This enables both the comparisons of different fusion methods and their effects on traditional and fuzzy learning methods. We also utilize the low-level feature weight in this section to assess the method and its effects in high-level fusion.
- Thirdly, we run a different test session solely applying the methods introduced in Chapter 6. These are the utilization of SOM-based visual codebooks and distinctive SIFT features in BOW model. We also evaluate the effectiveness of SOM in codebook generation by generating a number of codebooks both with SOM and k-means clustering methods. In the experiments, we also evaluate Bayesian and KNN classifier along with the SVM.
- Finally, we run experiments to compare the results obtained in different classification schemes, specifically we take the best single SVM and fuzzy SVM from first runs, different fusion results and BOW approach in a single view. We also compare our

results against the top-performed works that use the same dataset in the literature.

The organization of the rest of this chapter is as follows:

- Section 7.1 present the training and test phases of the classification system by describing the steps performed in each phase,
- Section 7.2 gives information about the data set used during the experiments including the object classes, and the evaluation metrics used in this set.
- Section 7.3 explains the selected MPEG-7 descriptors along with the basic tasks applied in the extraction and preparation of the low-level features in the system.
- Section 7.4 presents the results of individually trained binary classifier applied on the test data and compares the effect of fuzzy learning in the classification system,
- Section 7.5 present the results obtained in the fusion step of the system, in which we apply different classifier combinations on individual classifiers trained by SVM and fuzzy SVM,
- Section 7.6 is related to the experiments for evaluation of our approach in BOW model, in which the visual codebooks are generated by training separate SOMs. We demonstrate the results obtained by three different classification techniques: NB, KNN and SVM, both with and without the use of distinctive SIFT features.
- Finally, Section 7.7 summarizes the entire tests in the scope of image classification, present the major improvements achieved in the dissertation. We also give comparisons of our methods to the top-performing published works in the literature.

7.1 The Training and Test Steps Applied in Image Classification

In this section we give a detailed view on the implementation of the tasks performed in training and test data sets. Our image classification system is based on supervised learning, and hence the whole process is performed in two different modes, training and test, respectively. During the training phase, we work on the images in training data collection using the available class annotations, and in the test phase we run different experiments to evaluate the performance of our classification system on test data.

As can be seen in the functional view of the overall system in Figure 3.2, we first run the tasks in the training phase, and form the structures that are used in high-level feature extraction process, such as the SOM networks, the SVM classifiers, feature weights, and so on. Then, these models and structures are utilized in the Test phase of the system to detect the high-level semantic information in unlabeled test data. In the following sections, we explain the design of the training and test phases, respectively.

7.1.1 Training Design in Image Classification

The training phase of the system is shown in Figure 7.1. The main tasks can be grouped into four layers: 1) Feature extraction, 2) SOM construction, 3) Feature mapping, and 4) Classifier learning.

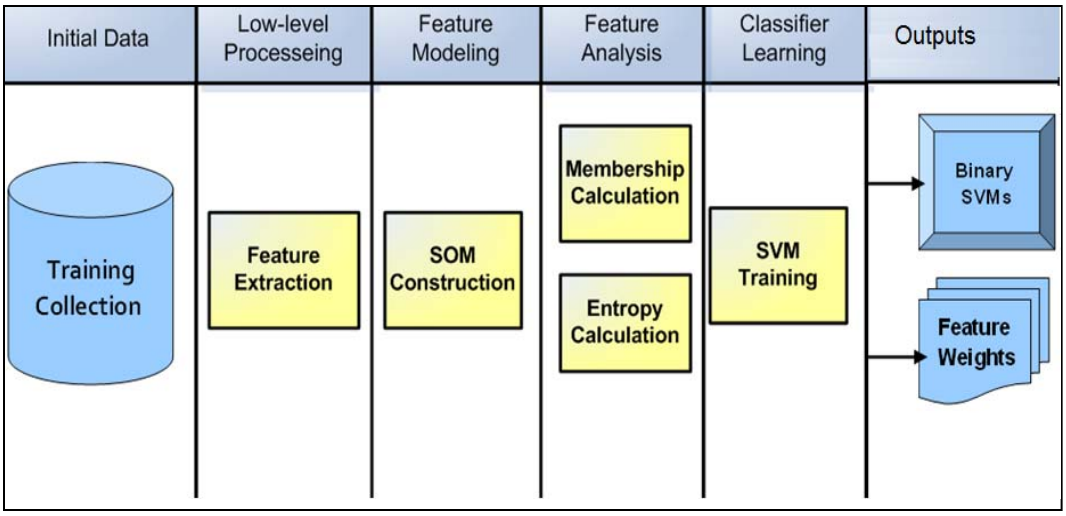


Figure 7.1 The training phase of our classification system.

The training phase of the proposed concept detection system is designed as a multi-layered sequential process, and each task corresponds to a layer described in Section 3.1. Figure 7.1 shows the individual steps performed in training, where starting from a set of training images and target concepts, several binary concept classifiers and feature weights are built for each concept. The steps performed in the training phase are presented in Algorithm 7.2.

The training phase starts with s*egmentation* process. In this step; the training images are divided into homogenous regions followed by the *annotation* task, in which the important image regions are assigned using keywords. After the annotation step, *feature extraction* is performed on the labeled image regions and a number of low-level visual features are extracted locally. Then, the required data files are prepared, and individual SOMs are constructed for each concept using the positive samples of the training set in the *Feature Mapping* layer. Next, we calculate the membership degrees of the training vectors utilizing the class-based trained SOMs by mapping the low-level features on them to determine the *bmu* hits for each image class and feature type followed by the clustering algorithm presented Algorithm 5.1. The final step in the training process is the *classifier learning* tasks, in which we train a number of binary classifiers applying both the classical and fuzzy learning methods in SVM. Another task in the final step is to analyze the entropy of the feature distributions on SOM clusters, and determine the low-level feature weights as described in Chapter 5.

```
Algorithm 7.1: Implementation of the training phase
Input : A set of training images and semantic concepts
Output : A set of binary classifiers, and feature weights for each semantic concept
1: Perform segmentation on the input images
    $[img\_regions] = \mathrm{Segment}(train\_images)$
2: Assign keyword annotation to the regions
    $[annotated\_regions] = \mathrm{Annotate}(img\_regions, concepts)$
3: Extract low-level features from the annotated regions locally
    $[low\_level\_features] = \mathrm{Extract}(annotated\_regions)$
4: Construct SOMs
    $[soms] = \mathrm{Som\_Make}(low\_level\_features^{+}, concepts)$
5: Calculate the Memberships
    $[memberships] = \mathrm{Calculate\_\ Membership}(low\_level\_features, soms)$
6: Apply SOM clustering
    $[clusters] = \mathrm{2d\_SOM\_Cluster}(low\_level\_features^{+}, soms)$
7: Train SVMs and Fuzzy SVMs
    $[svms] = \mathrm{SVM\_Train}(low\_level\_features)$
    $[fsvms] = \mathrm{Fuzzy\_SVM\_Train}(low\_level\_features, memberships)$
8: Determine feature weights
    $[feature\_weights] = \mathrm{Entropy\_Calculate}(clusters,\ concepts)$
```

7.1.2 Test Design in Image Classification

Figure 7.2 shows the steps performed on test data. The test phase can be viewed in three main steps: 1) Feature extraction, 2) Classification, and 3) High-level fusion. Fusion step is the most critical part in test phase, since the final decision related to a test image is given in this step, and we utilize a novel fusion approach in the scope of our classification framework.

The first step of the test is *low-level feature extraction*, where the same extraction process similar to training is applied on test regions. After the features are obtained from test regions, the corresponding concept classifiers are evaluated and individual classification results are generated. One important point in classifier evaluation that differs from the previous works is that we perform a probabilistic classification model in this step. The classifiers return both the sign of the evaluation and the strength of the classification at the same time. This value represents the classifier's degree of confidence and later used in fusion step. The third and last step of test phase is high-level classifier fusion, where the single classifier results are merged to produce the final results related to a test region. The fusion step is performed by applying a modified DS combination rule, where the feature weights determined in training phase, the single classification results and their confidences are merged.

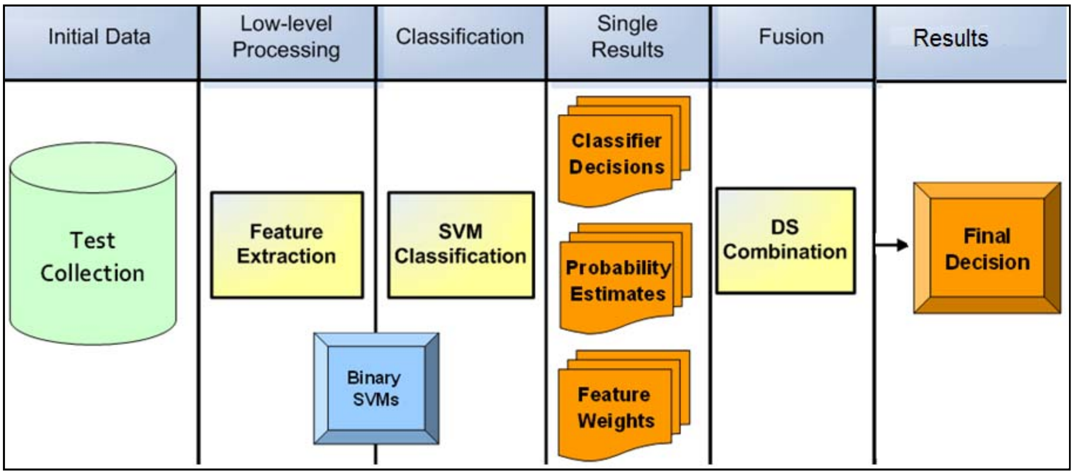


Figure 7.2 The test phase of our classification system.

Algorithm 7.2 depicts the tasks performed in the test of the system. The test process is much simpler than the training, and a set of test images are examined to determine the high-level concepts.

Algorithm 7.2: Implementation of the test phase

```
Input : A set of test images and target concepts
Output : High-level semantic concepts detected in the images
1: Extraction of the low-level features from images.
    $[low\_level\_features] = \text{Extract}(img\_regions)$
2: Perform classification using both SVMs and Fuzzy SVMs
    $[decision\_svms, confidence\_svms] = \text{Classify}(svms, low\_level\_features)$
    $[decision\_fsvms, confidence\_fsvms] = \text{Classify}(fsvms, low\_level\_features)$
3: Apply high-level fusion on classifier outputs.
    $[final\_decision\_svms] =$
        $\text{DS\_Combination}(decision\_svms, confidence\_svms, feature\_weights)$
    $[final\_decision\_fsvms] =$
        $\text{DS\_Combination}(decision\_fsvms, confidence\_fsvms, feature\_weights)$
```

7.2 Dataset and Evaluation Metric

In this section, we first provide information about the data set used during the implementation training and test phases together with the image classes used. Then, we describe MPEG-7 features selected as the low-level features in the implementation of our system along with classifier evaluation metrics.

7.2.1 Dataset

In order to show the effectiveness of our approaches in obtaining high-level semantic information from visual data, we conduct experiments on PASCAL Visual Object Classes (VOC) 2007/2008 collections [123], which are benchmark datasets for the classification and detection of visual objects by providing a standard set of images with annotations.

We select the VOC dataset in our evaluations due to the following reasons:

- The VOC dataset is publicly available to researchers and contains enough number of images for implementing the models presented in this dissertation,
- A large number of natural images are collected from the web, in which the images are categorized into 20 semantic classes, and each class has more than 200 images in training and test collections,
- The dataset also provides the object annotations publicly available,
- There major objective of our thesis is enabling better classifications on images, which is one of the main challenges in organizing the VOC dataset,
- The dataset contains some challenging images, since the images are not are not produced for image analysis purposes,
- There are many works presented on this dataset, a standard evaluation methodology provided by the dataset makes it possible to make comparison with published works.

The VOC dataset is formed by annotating user photographs collected from the Flickr photo-sharing web-site. This contains over 10.000 images in three collections: train, validation and test, and categorized into 20 semantic classes. There are nearly 24.600 object annotations in the classes as shown in Table7.1.

We used 10 out of the 20 available classes in our implementations, and build binary classifiers for each class individually. Table 7.1 presents the classes that are used in our experiments along with the total number of samples in training and test sets.

The VOC images include a wide range of viewing conditions, such as lighting, pose, and so on. An image can include more than one object classes, and each class in an image is annotated using the following attributes:

- **Class:** one of the following semantic labels: airplane, bird, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, sofa, train, and tv/monitor.
- **A bounding box:** The bounding box in an image that includes a specific object.

Table 7.1: The classes and number of images used in the experiments.

	TRAIN		TEST	
Classes	image	object	image	object
airplane	238	306	204	285
boat	181	290	172	263
car	713	1250	721	1201
cat	337	376	322	358
cow	141	259	127	244
horse	287	362	274	348
motorbike	245	339	222	325
person	2008	4690	2007	4528
sofa	229	248	223	239
tv/monitor	256	324	229	308
Total	4635	8444	4501	8099

7.2.2 Evaluation Metric

Up to 2005 and 2006 VOC challenges, the evaluation measure for the classification performance was ROC curve. Since then, it is changed to AP. For a given class, the performance of a classifier is evaluated as follows [ ]:

- The precision/recall curve is computed from classifier outputs,
- Recall is defined as the proportion of all positive examples ranked above a given threshold,
- Precision is used as the proportion of all examples above that threshold from the positive class,
- The AP summarizes the shape of the precision/recall curve, and is defined as the mean precision at eleven equally spaced recall points: (0,0.1.. ,1).

The precision at each recall level is *interpolated* by taking the maximum precision measured for a method. This AP-based evaluation provides the following advantages over the previously used ROC/AUC curve:

- The sensitivity in classifier evaluations is improved,
- The AP or MAP gives better interpretability in visual analysis, since the aim is achieving the true positives as early as possible,
- The evaluation also enables increased visibility of performances for low recall classes [ ].

7.3 Extraction and Preparation of Low-level Features

In the extraction low-level visual features for classification learning tasks, we select to implement the visual part of the MPEG-7 standard. Since, the MPEG-7 provides a standard set of low-level visual features, and used in a large amount of published work in the literature. In this section, we briefly introduce the MPEG-7 descriptors, and their preparation in the scope of our classification system.

7.3.1 The MPEG-7 Descriptors

We select 5 out of the 8 visual descriptors from the MPEG-7 standard as follows:

- Three color features, *Color Layout Descriptor* (CLD), *Color Structure Descriptor* (CSD), and *Scalable Color Descriptor* (SCD),
- One texture feature, *Edge Histogram Descriptor* (EHD), and
- One shape based feature, *Region Shape Descriptor* (RSD).

The reason for choosing them as low-level features is that they provide the most independently formed features, which provide high discriminations in image analysis. The comprehensive analysis and statistical comparison of MPEG-7 descriptors can be found in [17].

The feature extraction task is implemented by employing the MPEG-7 XM reference implementation [18], and at the end of this process we extract 5 different low-level features independently for each image or image region. We use the following parameters during the feature extraction:

- The NumberOfYCoeff and NumberOfCCoeff parameters of *CLD* are quantized to 6 and 3 bins, respectively,
- The *CSD* is quantized to 256 bins (e.g. *ColorQuantSize*=256),
- The coefficients in *SCD* is set to 256 (e.g. *NumberOfCoefficients*=256).

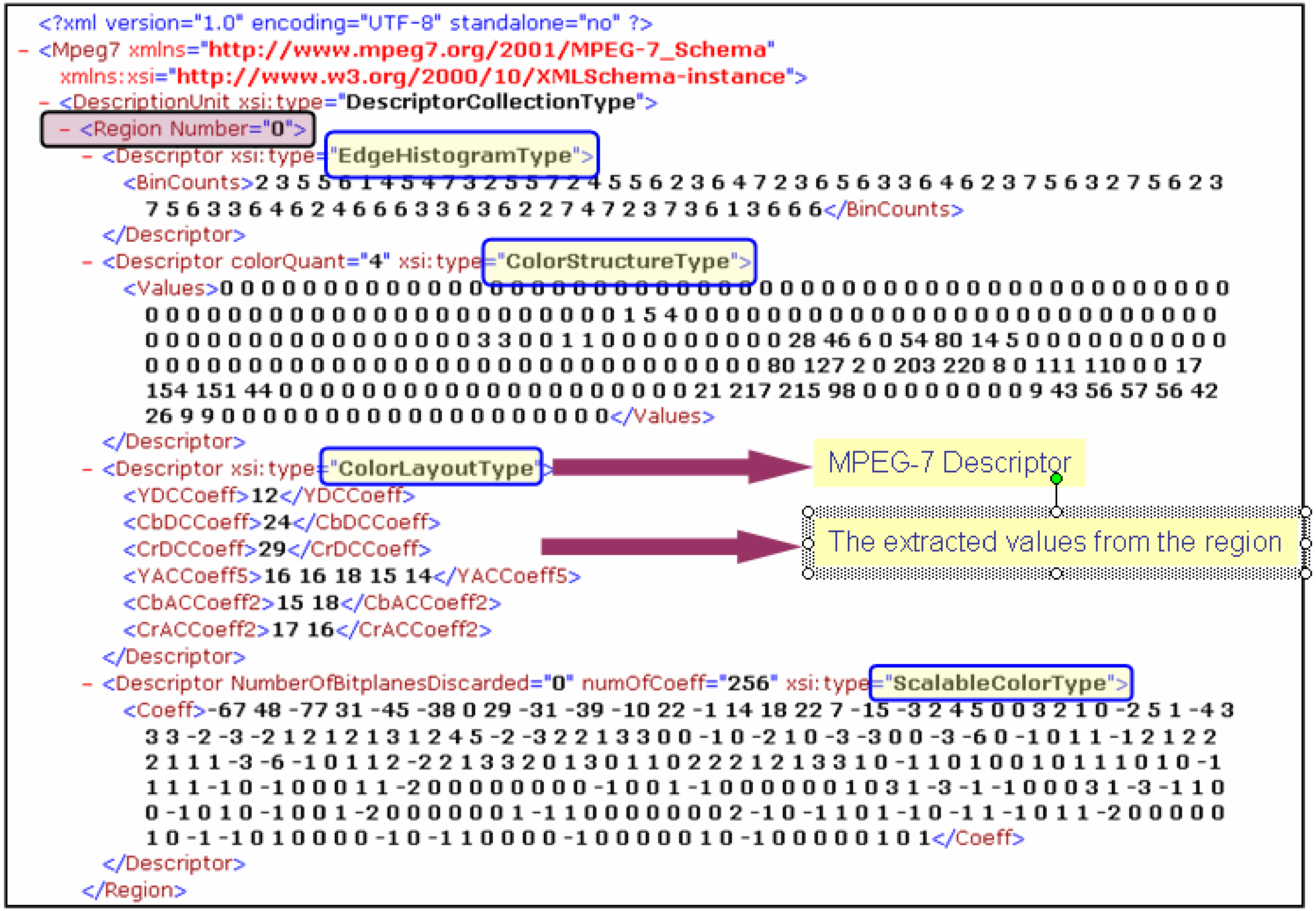
An example XML file generated by applying the feature extraction process to a sample image is shown in Figure 7.3.

7.3.2 Data Preparation

The low-level visual features of training data are used in the following tasks, as can be seen in Figure 3.1: 1) In SOM construction, membership calculation and feature entropy analysis tasks of *Feature Modeling Layer*, and 2) SVM training in *Learning Layer*. Once the low-level features are obtained from the annotated image regions, a number of pre-processing tasks are performed on them in order to prepare the required feature vectors, which can be used in different computations in the scope of our visual analysis system.

During the implementation of the concept detection system, the low-level features extracted from the training set are used in the following computations:

1) *SOM Training*: A separate SOM is constructed for each concept and descriptor type utilizing the positive samples as defined in Section 4.3.
2) *Membership Calculation*: The membership calculation is performed on each training vector to determine the membership values as explained in Section 4.4.



```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
- <Mpeg7 xmlns="http://www.mpeg7.org/2001/MPEG-7_Schema"
   xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance">
 - <DescriptionUnit xsi:type="DescriptorCollectionType">
  - <Region Number="0">
   - <Descriptor xsi:type="EdgeHistogramType">
      <BinCounts>2 3 5 5 6 1 4 5 4 7 3 2 5 5 7 2 4 5 5 6 2 3 6 4 7 2 3 6 5 6 3 3 6 4 6 2 3 7 5 6 3 2 7 5 6 2 3
       7 5 6 3 3 6 4 6 2 4 6 6 6 3 3 6 3 6 2 2 7 4 7 2 3 7 3 6 1 3 6 6 6</BinCounts>
     </Descriptor>
   - <Descriptor colorQuant="4" xsi:type="ColorStructureType">
      <Values>0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
       0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 5 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
       0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 3 0 0 1 1 0 0 0 0 0 0 0 0 0 0 28 46 6 0 54 80 14 5 0 0 0 0 0 0 0 0 0 0
       0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 80 127 2 0 203 220 8 0 111 110 0 0 17
       154 151 44 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 21 217 215 98 0 0 0 0 0 0 0 0 9 43 56 57 56 42
       26 9 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0</Values>
     </Descriptor>
   - <Descriptor xsi:type="ColorLayoutType">
      <YDCCoeff>12</YDCCoeff>
      <CbDCCoeff>24</CbDCCoeff>
      <CrDCCoeff>29</CrDCCoeff>
      <YACCoeff5>16 16 18 15 14</YACCoeff5>
      <CbACCoeff2>15 18</CbACCoeff2>
      <CrACCoeff2>17 16</CrACCoeff2>
     </Descriptor>
   - <Descriptor NumberOfBitplanesDiscarded="0" numOfCoeff="256" xsi:type="ScalableColorType">
      <Coeff>-67 48 -77 31 -45 -38 0 29 -31 -39 -10 22 -1 14 18 22 7 -15 -3 2 4 5 0 0 3 2 1 0 -2 5 1 -4 3
       3 3 -2 -3 -2 1 2 1 2 1 3 1 2 4 5 -2 -3 2 2 1 3 3 0 0 -1 0 -2 1 0 -3 -3 0 0 -3 -6 0 -1 0 1 1 -1 2 1 2 2
       2 1 1 1 -3 -6 -1 0 1 1 2 -2 2 1 3 3 2 0 1 3 0 1 1 0 2 2 2 1 2 1 3 3 1 0 -1 1 0 1 0 0 1 0 1 1 1 0 1 0 -1
       1 1 1 -1 0 -1 0 0 0 1 1 -2 0 0 0 0 0 0 0 0 -1 0 0 1 -1 0 0 0 0 0 0 1 0 3 1 -3 -1 -1 0 0 0 3 1 -3 -1 1 0
       0 -1 0 1 0 -1 0 0 1 -2 0 0 0 0 0 0 1 -1 1 0 0 0 0 0 0 0 2 -1 0 -1 1 0 1 -1 0 -1 1 -1 0 1 1 -2 0 0 0 0 0
       1 0 -1 -1 0 1 0 0 0 0 -1 0 -1 1 0 0 0 0 -1 0 0 0 0 0 1 0 -1 0 0 0 0 0 1 0 1</Coeff>
     </Descriptor>
    </Region>
```

Figure 7.3 A sample MPEG-7 file in xml format.

3) *Classical SVM training*: We have also trained independent SVMs in a similar manner with fuzzy training without considering the memberships.
4) *Fuzzy SVM Training*: We have trained fuzzy SVMs for each high-level concept and descriptor type using all the training samples and the computed membership values, as described in Section 4.5.

We illustrate a sample data preparation example and provide two sample images and corresponding region annotations in Figure 7.4. Table 7.2 and Table 7.3 show the utilization of the regions along with the class labels of the regions shown in Figure 7.4 in SOM construction and SVM training tasks, respectively. Also the utilization of regions in membership calculation and SOM clustering processes are provided in Table 7.4.

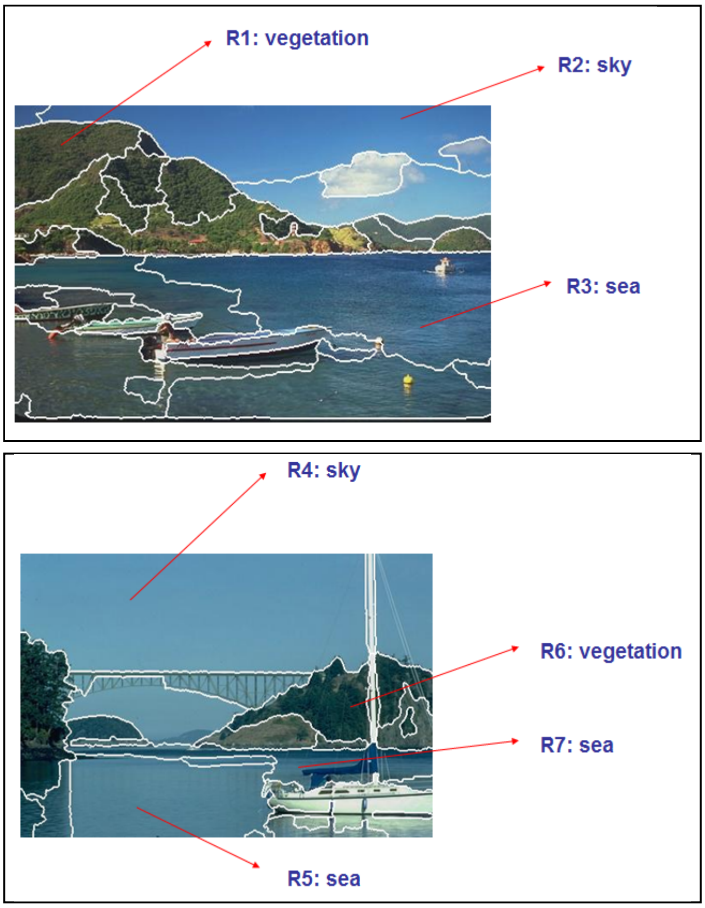


Figure 7.4 Sample images with annotations.

Table 7.2: Data preparation for SVM training in learning the Sky concept.

Region	Used?	Class
R1	Yes	-
R2	Yes	+
R3	Yes	-
R4	Yes	+
R5	Yes	-
R6	Yes	-
R7	Yes	-

Table 7.3: Data preparation for SOM construction .

Class	Regions Used	Class
Sky	[R2,R4]	+
Sea	[R3,R5,R7]	+
Vegetation	[R1, R6]	+

Table 7.4: The utilization of regions in membership calculation and entropy analysis for *sky* class.

Region	Used? (Membership)	Used? (Entropy Analysis)
R1	Yes	No
R2	Yes	Yes
R3	Yes	No
R4	Yes	Yes
R5	Yes	No
R6	Yes	No
R7	Yes	No

The low-level features extracted from the test images are used solely used as input to test the SVM and fuzzy SVM classifiers in the classification system.

7.4 Classification Results of Single Classifiers

In this section we provide the classification results of individual SVM classifiers trained by distinct visual features, and investigate the effect of fuzzy learning in our classification system.

Table 7.5 and Table 7.6 present the AP results obtained from individual classifiers trained by SVM and fuzzy SVM techniques. Before presenting the test results, we exclude the classifiers trained on the RS descriptor in the comparisons since the descriptor usually generates poor classification results in both SVM and fuzzy SVM schemes, in which it returns less than 1% of the positives in some classes such as *cat* and *horse* classes.

Table 7.5: Individual Classification Results (SVM).

Method	**Classes**	**MPEG-7 Features**					
		CL	CST	EH	RS	SC	*Average*
SVM	airplane	0.49	0.47	**0.54**	0.07	0.36	0.39
	boat	0.45	**0.49**	0.31	0.12	0.41	0.36
	car	0.52	0.55	**0.61**	0.16	0.52	0.47
	cat	0.28	**0.45**	0.34	0.00	0.36	0.29
	cow	0.24	0.29	0.25	0.01	**0.27**	0.21
	horse	0.22	**0.46**	0.42	0.00	0.42	0.30
	motorbike	0.01	**0.39**	0.47	0.00	0.37	0.25
	person	0.53	0.64	**0.69**	0.32	0.63	0.56
	sofa	**0.27**	0.14	0.25	0.00	0.22	0.18
	tv/monitor	0.30	0.26	0.24	0.00	**0.32**	0.22

Table 7.6: Individual Classification Results (Fuzzy SVM).

Method	**Classes**	**MPEG-7 Features**					
		CL	CST	EH	RS	SC	*Average*
FSVM	airplane	0.49	0.48	**0.56**	0.19	0.39	0.42
	boat	0.47	**0.48**	0.33	0.18	0.44	0.38
	car	0.54	0.57	**0.61**	0.25	0.54	0.50
	cat	0.29	**0.46**	0.38	0.00	0.4	0.31
	cow	0.27	**0.30**	0.29	0.02	0.28	0.23
	horse	0.29	**0.52**	0.45	0.00	0.44	0.34
	motorbike	0.00	0.43	**0.51**	0.00	0.39	0.27
	person	0.56	0.69	**0.72**	0.41	0.65	0.61
	sofa	**0.29**	0.18	0.27	0.00	0.25	0.20
	tv/monitor	0.31	0.26	0.28	0.00	**0.35**	0.24

When we compare the individual classification results shown in Table 7.5 and Table 7.6, the texture- and color- based features outperform shape-based feature (e.g. RS) both in classical SVM and Fuzzy SVMs. For some classes, such as *boat*, the color features perform better than the texture feature (e.g. EH), and for some classes, such as *car* or *person*, the texture feature produces the best classification scores.

Table 7.7 shows the mean AP scores (MAP) obtained in the experiments by SVM and Fuzzy SVM classifiers. For all image classes, the fuzzy approach has improved the classification performance; the highest improvement is achieved in *horse* class by 12.7%.

Table 7.7: Comparison of single classifiers in SVM and fuzzy SVM.

Classes	**MAP SVM**	**MAP FSVM**	**Improvement (%)**
airplane	0.39	0.42	8.94
boat	0.36	0.38	6.72
car	0.47	0.50	6.40
cat	0.29	0.31	6.56
cow	0.21	0.23	**10.73**
horse	0.30	0.34	**12.69**
motorbike	0.25	0.27	7.48
person	0.56	0.61	7.71
sofa	0.18	0.20	**11.17**
tv/monitor	0.22	0.24	8.04
Average	0.32	0.35	8.64

We also present the initial experimental results to analyze the effect of our fuzzy learning approach with a different membership function in [124]. We use a smaller dataset for these experiments, the VOC 2006 collection, which includes images taken with the purpose of capturing certain large objects and enables a more appropriate evaluation of fuzzy learning approach in image classification.

In the evaluation, we present the classifier results using Precision, Recall and F-score basis. Table 7.8 shows the results obtained by SVM and fuzzy SVMs learning approaches using the membership function in [124][ ].

Fuzzy SVM has increased the recall rates up to 40% (for *person* class) and F-measure up to 23% (for *motorbike* class). We notice that the improvement achieved by Fuzzy SVM on poor descriptors, i.e. in terms of reduced classification accuracy, is far more than the rich ones. To make it clearer, fuzzy SVM increases the RS classifier's F-score performance from 16% to 40% in *car* class, but the effect is less than 1% using the SC feature.

Table 7.8: Comparison of methods on VOC 2006 dataset using MAP.

	SVM			FSVM		
Classes	Precision	Recall	F-score	Precision	Recall	F-score
car	0.80	0.55	0.65	0.58	0.70	0.64
cat	0.62	0.27	0.38	0.56	0.35	0.43
cow	0.74	0.36	0.49	0.70	0.39	0.50
horse	0.51	0.16	0.24	0.47	0.21	0.29
m.bike	0.76	0.19	0.30	0.67	0.26	0.37
person	0.67	0.52	0.58	0.55	0.73	0.63

The main impact of fuzzy learning approach in SVM training is an increase in recall but a decrease in precision, in other words it increases the number of retrieved true positives and false positives at the same time. Moreover, we observe that the effect of fuzzy approach highly depends on the performance of the classifiers, in which the improvements achieved by fuzzy approach in poor descriptors are higher than the good ones, in general. For example, the increase in recall for *person* class applying CLD classifier, which is the worst descriptor in this class (discarding RSD), is nearly 38%. However, the increase in recall on the same class is less than 4% for the EHD, which is the best concept classifier for the *person* class.

In the following figures, we provide the class-based evaluations obtained using the VOC 2006 dataset.

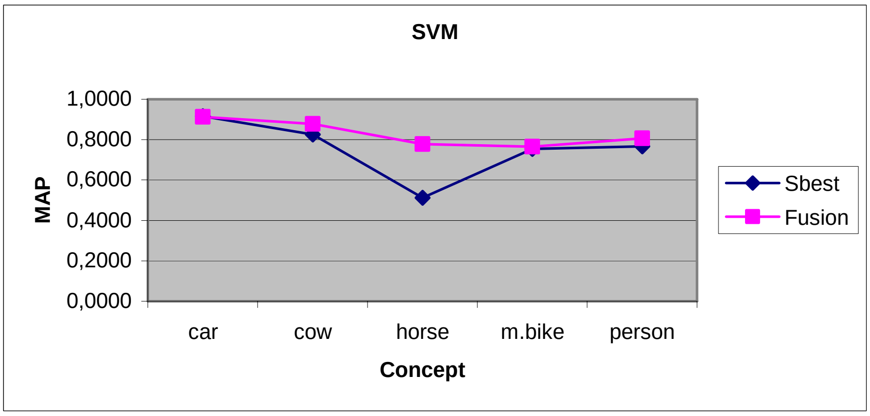


Figure 7.5 Effects of DS-fusion in classical SVM (MAP).

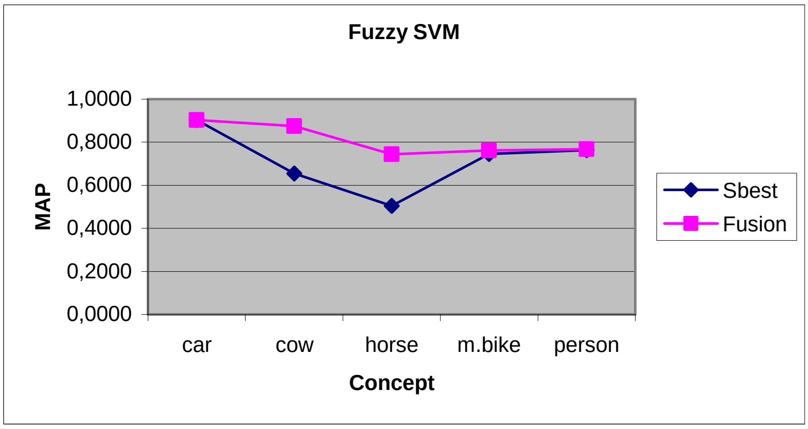


Figure 7.6 Effects of DS-fusion in fuzzy SVM (MAP).

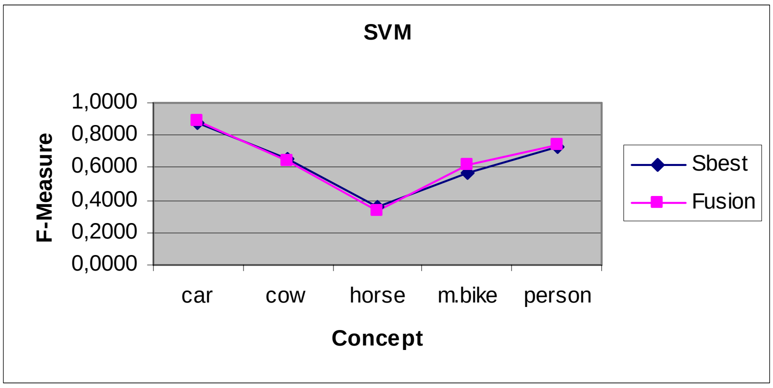


Figure 7.7 Effects of DS-fusion in classical SVM (F-measure).

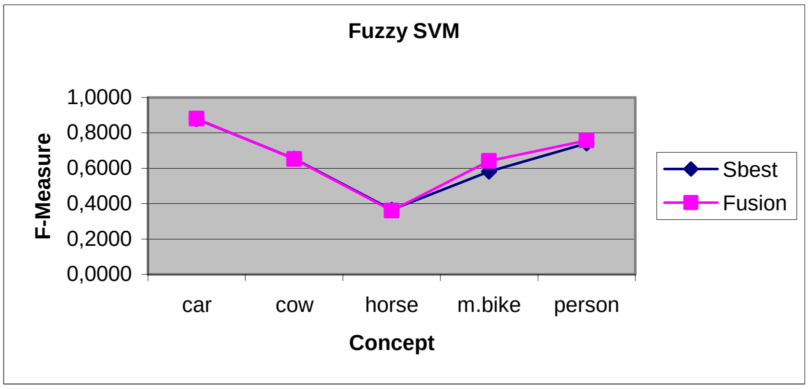


Figure 7.8 Effects of DS-fusion in fuzzy SVM (F-measure).

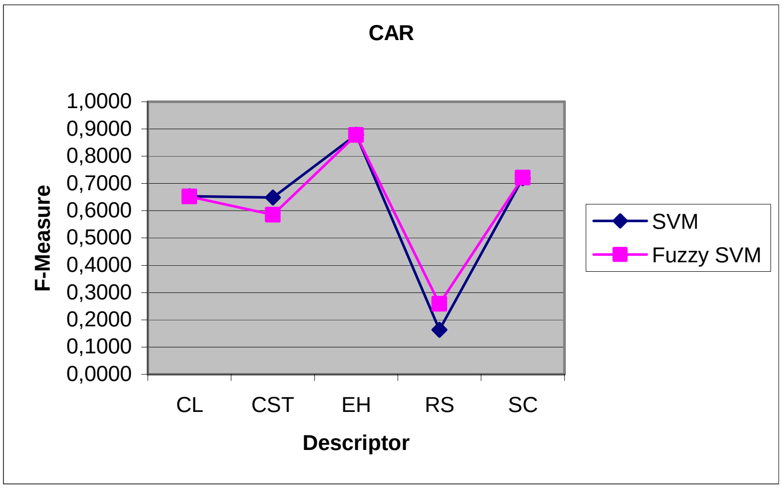


Figure 7.9 F-measure results of SVM and fuzzy SVM (*car* class).

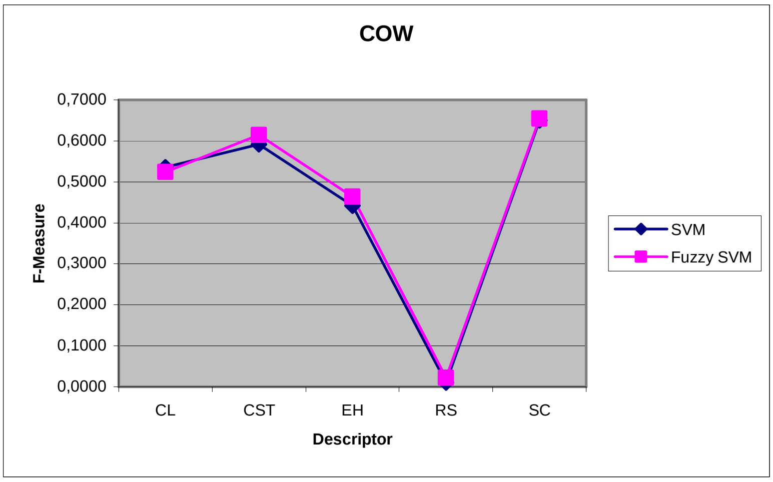


Figure 7.10 F-measure results of SVM and fuzzy SVM (*cow* class).

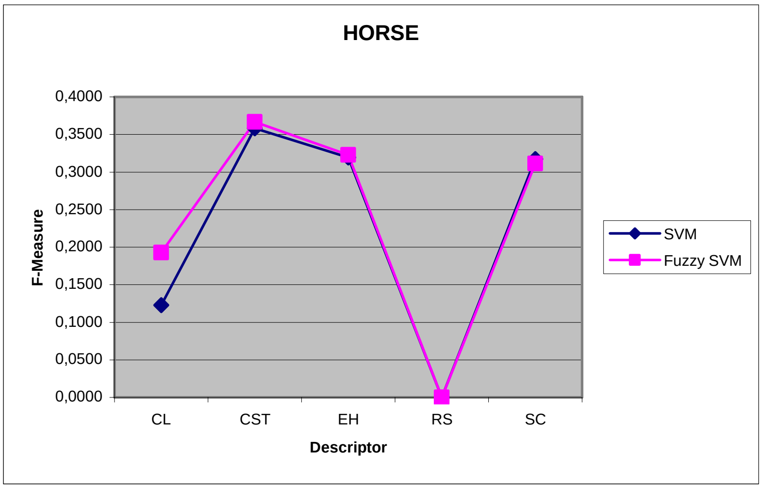


Figure 7.11 F-measure results of SVM and fuzzy SVM (*horse* class).

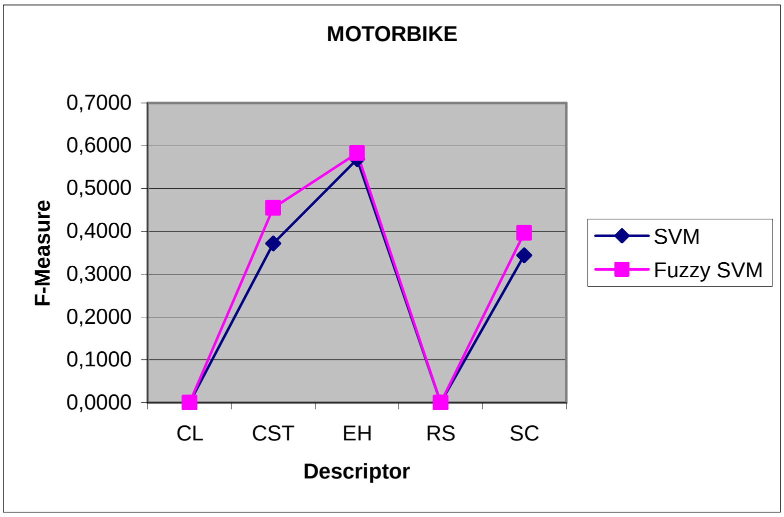


Figure 7.12 F-measure results of SVM and fuzzy SVM (*motorbike* class).

7.5 Combined Classification Results

In this section we present the results obtained in fusion process. As described in Chapter 5, we apply a decision-level fusion on single classifier outputs to obtain final results in classifying the test images. In order to express the improvement provided by DS and Weighted (DS) combinations, we compare the results generated by the best single SVM classifiers with the combined results for each image class, and provide the AP scores in this section.

Table 7.9 and Table 7.10 depict the results obtained by best single SVM along with the combined results generated by four different fusion schemes: Majority Voting (MV), Weighted MV (W-MV), DS, and Weighted DS (W-DS) fusions, respectively.

Table 7.9: MAP scores of best single classifier and fusion in classical SVM.

Method	**Classes**	**Best Single**	**Majority Voting (MV)**	**Weighted MV**	**DS Fusion**	**Weighted DS Fusion**
SVM	airplane	0.54	0.44	0.46	0.55	**0.58**
	boat	0.49	0.42	0.45	0.53	**0.54**
	car	0.61	0.53	0.57	0.62	**0.66**
	cat	0.45	0.36	0.39	0.44	**0.48**
	cow	0.29	0.23	0.27	0.27	**0.31**
	horse	0.46	0.37	0.41	0.54	**0.57**
	motorbike	0.47	0.41	0.44	0.48	**0.53**
	person	0.69	0.60	0.64	0.67	**0.70**
	sofa	0.27	0.22	0.25	0.27	**0.29**
	tv/monitor	0.32	0.28	0.31	0.32	**0.33**
	Average	0.46	0.39	0.42	0.47	**0.50**

Table 7.10: MAP scores of best single classifier and fusion in fuzzy SVM.

Method	**Classes**	**Best Single**	**Majority Voting (MV)**	**Weighted MV**	**DS Fusion**	**Weighted DS Fusion**
FSVM	airplane	0.56	0.45	0.49	0.60	**0.65**
	boat	0.48	0.43	0.47	0.52	**0.59**
	car	0.61	0.55	0.58	0.63	**0.70**
	cat	0.46	0.41	0.42	0.45	**0.49**
	cow	0.30	0.29	0.29	0.29	**0.34**
	horse	0.52	0.47	0.48	0.58	**0.66**
	motorbike	0.51	0.44	0.46	0.53	**0.57**
	person	0.72	0.65	0.68	0.71	**0.78**
	sofa	0.29	0.26	0.27	0.32	**0.35**
	tv/monitor	0.35	0.31	0.32	0.34	**0.40**
	Average	0.48	0.42	0.44	0.50	**0.55**

First, when we compare the best single classifiers and voting schemes, both MV and W-MV combinations do not produce better results. This is mainly caused from the adverse effect of the poor classifiers in the fusion. However, the W-MV performs better than MV, which shows the effect of feature weights in the combination.

Secondly, the DS combination produces better results than the voting schemes. However, compared to the best single SVMs, the DS has different effects on classifier performances: for some classes, such as the *boat* or *horse*, it increases the classification results, but for some classes, such as the *cat* and *person*, the DS also decreases the performance of best SVM.

Thirdly, the use of low-level feature weights in DS fusion has a positive impact on classifier combination, in which the W-DS has produced the best classification results for all classes. When we compared the results of best single SVMs to the W-DS, we obtain performance increases up to %24 and %26 for the *horse* class, in SVM and fuzzy SVM methods, respectively.

Table 7.11 shows the AP results obtained by DS and W-DS in SVM and fuzzy SVM schemes. As can be seen in the table, the use of low-level feature weights has a positive impact on the combined results. When we compare the DS and W-DS results, the latter has an improvement by 15% to former (*car* class) in SVM, and 18% in fuzzy SVM (*tv/monitor*).

Table 7.11: Effect of feature weights on combined results.

		SVM			**FSVM**	
Classes	**DS**	**W-DS**	**Improvement%**	**DS**	**W-DS**	**Improvement%**
airplane	0.55	0.58	5.45	0.60	0.65	8.33
boat	0.53	0.54	1.89	0.52	0.59	**13.46**
car	0.62	0.66	6.45	0.63	0.70	11.11
cat	0.44	0.48	9.09	0.45	0.49	8.89
cow	0.27	0.31	**14.81**	0.29	0.34	**17.24**
horse	0.54	0.57	5.56	0.58	0.66	**13.79**
motorbike	0.48	0.53	**10.42**	0.53	0.57	7.55
person	0.67	0.70	4.48	0.71	0.78	9.86
sofa	0.27	0.29	7.41	0.32	0.35	9.38
tv/monitor	0.32	0.33	3.13	0.34	0.40	**17.65**
Average	0.47	0.50	6.87	0.49	0.55	**11.7**

We compare the results of W-DS combinations in SVM and F-SVM in Table 7.12. The effect of fuzzy learning approach in SVM on the combined classifier results is similar to the individual SVMs, which ranges from 2% (*cat*) to 21% (*tv/monitor*).

Table 7.12: Effect of fuzzy learning on combined results (W-DS).

Classes	W-DS Fusion (SVM)	W-DS Fusion (FSVM)	%
airplane	0.58	0.65	12.07
boat	0.54	0.59	9.26
car	0.66	0.70	6.06
cat	0.48	0.49	2.08
cow	0.31	0.34	9.68
horse	0.57	0.66	**15.79**
motorbike	0.53	0.57	7.55
person	0.70	0.78	11.43
sofa	0.29	0.35	**20.69**
tv/monitor	0.33	0.40	**21.21**
Average	0.50	0.55	11.6

7.6 Classification Results in BOW Model

In order to show the performance of our BOW approach described in Chapter 6 in image classification, we set up several experiments using the same VOC dataset [123]. We construct the visual codebooks by training separate SOMs individually, and then utilize them in the experiments.

In this section, we have explored the performances of different classification techniques along with the impact of distinctive features on these techniques in the classification of test images. For each test image, we evaluate the classifiers using two different inputs: the entire SIFT features extracted from an image, and only the distinctive SIFT features.

The features obtained from training set are used for the following tasks:

- *Construction of codebooks*: One separate SOM is built for each object class in order to represent the codebook of the class.
- *Computation of visual-word frequencies*: For each image class, the probabilities of visual-words are computed using the hit frequencies obtained in feature mapping step of the training phase.
- *Representation of training images*: The representation of images is required in learning SVM classifiers for each object class. The SIFT features of the entire training data is mapped to each SOM individually, and the images in the training set are represented by the codebook histograms obtained in this mapping.
- *Determination of thresholds*: The determination of the optimal threshold values is required before applying the distinctive features presented in Chapter 6. Hence, we

determine the optimal threshold values for each image class by applying a five-fold cross-validation on the training data [90].

The features extracted from test images are used in the following computations:

- *Detection of distinctive features*: We apply the distinctive feature detection method to determine the distinctive features in a test image.
- *Representation of test images*: Similar to the training phase, each image in the test set is also represented in terms of codebook histograms by mapping the features of each image to the codebooks. These vectors are used as input in the SVM classifier.

7.6.1 Experimental Results and Performance Evaluation in BOW

We present the classification results of 10 object classes of our classification system in Table 7.13. The AP is used as in the experiment to enable a direct comparison of our BOW-based classification results with the previous methods and other published works using the same dataset [123].

The evaluation of the results is discussed according to the inputs used during the tests:

a) *Classification using all features:* When all image features are used in classifying the test images, the SVM has produced the best performance. The SVM, in general, has high-generalization capability and less prone to outliers than the NB and KNN classifiers, which makes SVM outperform NB and KNN methods even classifying noisy images.

When we compare the classification performances of NB and KNN, the former is superior to the latter. The main reason in this result is that the NB utilizes the visual-word frequencies, which are computed in the mapping of the images to the corresponding codebooks during the training phase of the classification system. To make it clearer, considering a background keypoint in an image, the effect of the point in NB classifier is proportional to the frequency of the matching visual-word in all classes, and since the word is an incorrect match, the word-frequency is most probably low. However, the matching visual-word is used directly in the KNN classification.

b) *Classification using distinctive features*: First, the utilization of the distinctive features significantly increases the performances of NB and KNN classifiers. Specifically, considering the object classes with an average precision of over 20%, the average increase achieved by using the distinctive features is 57% in KNN and 45% in Bayesian classifications. The distinctive features improve the KNN in classifying noisy objects.

Secondly, the utilization of distinctive features does not contribute the SVM, and degrades the classification performance by 15% on average, since the SVM uses image representations in terms of visual-word histograms in its learning phase, and applying the distinctive feature method on the training images produces a reduced set of features in SVM learning. This makes SVM to loose some information during the training phase.

Table 7.13: The classification results obtained by BOW model (AP).

	SVM		K-NN		NB	
Classes	**All Feat.**	**Dist.Feat.**	**All Feat.**	**Dist.Feat.**	**All Feat.**	**Dist.Feat.**
airplane	**0.76**	0.73	0.39	0.51	0.49	0.72
boat	**0.69**	0.63	0.72	0.24	0.38	**0.69**
car	**0.75**	0.65	0.34	0.55	0.45	**0.75**
cat	**0.57**	0.45	0.21	0.31	0.31	0.50
cow	**0.42**	0.36	0.06	0.17	0.22	0.39
horse	**0.76**	0.71	0.20	0.38	0.40	**0.76**
motorbike	**0.61**	0.54	0.13	0.28	0.43	0.59
person	**0.84**	0.77	0.43	0.59	0.62	0.77
sofa	**0.46**	0.33	0.33	0.14	0.33	0.43
tv/monitor	**0.50**	0.43	0.12	0.17	0.34	**0.50**
Average	**0.63**	0.56	0.35	0.33	0.40	0.61

If we compare all classification schemes, the best results are achieved by either the SVM using all features or NB using distinctive features as can be seen in Table 7.13. The effect of using the distinctive features is observed in NB classification such that for 4 image classes (*boat, car, horse,* and *tv/monitor*) the NB has resulted in nearly the same classification performances as SVM. Additionally, for some classes, such as *airplane* and *person*, the SVM classifier using distinctive features also produces better or comparable results compared to NB using distinctive features.

7.7 Overall Evaluation and Comparison with Previous Works

In this section, we first provide the overall classification results obtained in our visual analysis system, and then compare the results to the top-performing methods [125-128] using the same dataset.

Table 7.14 shows the overall classification results obtained by different methods using either the MPEG-7 visual features or SIFT features described in this dissertation. We put the best results in the table obtained in each classification scheme. To make it clearer, the left-most three columns in Table 7.14 depicts the results of, 1) SVM classifier using all SIFT features, 2) NB classifier using distinctive SIFT features, and 3) and K-NN classifier using distinctive SIFT features. As mentioned in Chapter 6, these are the top-performing results obtained in our BOW approach. The remaining columns in the table are the results obtained using the MPEG-7 visual features, in which we use single best SVM, MV, W-MV, DS and W-DS combinations, respectively.

Table 7.14: The overall classification results (AP).

	SIFT Features			MPEG-7 Features (FSVM)				
Classes	**SVM All**	**NB Dist.**	**KNN Dist.**	**S Best**	**MV**	**W-MV**	**DS**	**W-DS**
airplane	**0.76**	0.72	0.51	0.56	0.45	0.49	0.60	0.65
boat	**0.69**	**0.69**	0.24	0.48	0.43	0.47	0.52	0.59
car	**0.75**	**0.75**	0.55	0.61	0.55	0.58	0.63	0.70
cat	**0.57**	0.50	0.31	0.46	0.41	0.42	0.45	0.49
cow	**0.42**	0.39	0.17	0.30	0.29	0.29	0.29	0.34
horse	**0.76**	**0.76**	0.38	0.52	0.47	0.48	0.58	0.66
motorbike	**0.61**	0.59	0.28	0.51	0.44	0.46	0.53	0.57
person	**0.84**	0.77	0.59	0.72	0.65	0.68	0.71	**0.78**
sofa	**0.46**	0.43	0.14	0.29	0.26	0.27	0.32	0.35
tv/monitor	**0.50**	**0.50**	0.17	0.35	0.31	0.32	0.34	0.40
Average	**0.63**	0.61	0.33	0.48	0.43	0.45	0.50	0.55

The best classifications are obtained either by the SVM using all SIFT features or the NB classifiers using distinctive features. The W-DS using MPEG-7 visual features follows these two classification schemes. Over the 10 image classes, the SVM using all SIFT features results the best results for 6 classes, and for the remaining 4 classes both the SVM and NB classifiers performs nearly the same (Table 7.14).

Finally, we put the result of previous works and our approaches introduced in this dissertation are shown in Table 7.15. In the table, we provide the performance results of the classifiers shown in Table 7.14 along with the published works in the VOC dataset. The detailed information about the reference works can be found in [123].

When the classification results are compared, we achieve superior or comparable performance results with the reference works in two schemes: 1) SVM classification using all image features, and 2) NB classification using distinctive features. The success in SVM classification originates from the codebook generation process performed in the system in which a separate SOM is trained for each image class using the images that contain an instance of that class. This approach produces more informative codebooks, which eventually facilitates the performance in SVM classification. The second improvement is based on the use of distinctive features, which eliminates the inverse effect of the irrelevant keypoints in NB classification. None of the referenced works make use of such feature elimination during their classification process [125-132].

Our classification methods, especially the SVM using all SIFT features and the NB using distinctive features are in top-performing classifiers. The W-DS and DS also outperform some of the previous studies as can be seen in the Table 7.15.

Table 7.15: Comparison with previous works (AP).

References [22]	a.pln	boat	car	cat	cow	horse	mbk.	prsn.	sofa	tv/mn	*Avg.*
INR-Gen. [129]	**0.77**	**0.71**	**0.78**	**0.58**	**0.42**	**0.77**	**0.64**	**0.85**	**0.50**	**0.53**	**0.66**
SVM-All	**0.76**	0.69	0.75	0.57	0.42	0.76	0.61	0.84	0.46	0.50	**0.64**
XRCE [130]	0.72	0.68	0.75	0.50	0.39	0.75	0.58	0.84	0.50	0.49	**0.62**
NB-Dist.	0.72	0.69	0.75	0.50	0.39	0.76	0.59	0.77	0.43	0.50	**0.61**
QMUL [129]	0.70	0.64	0.71	0.54	0.36	0.71	0.55	0.80	0.41	0.45	0.59
Weighted DS	0.65	0.59	0.7	0.49	0.34	0.66	0.57	0.78	0.35	0.40	0.55
UVA-Fuse [131]	0.67	0.58	0.61	0.41	0.27	0.69	0.51	0.79	0.36	0.40	0.53
INRIA-Lar. [22]	0.62	0.47	0.69	0.44	0.26	0.66	0.55	0.77	0.36	0.43	0.53
MPI BOW [132]	0.58	0.59	0.67	0.40	0.28	0.63	0.53	0.75	0.35	0.40	0.52
MCIP [132]	0.66	0.58	0.61	0.40	0.27	0.66	0.5	0.78	0.31	0.40	0.52
DS Fusion	0.60	0.52	0.63	0.45	0.29	0.58	0.53	0.71	0.32	0.34	0.50
Tsinghua [22]	0.62	0.49	0.62	0.35	0.21	0.65	0.48	0.76	0.32	0.33	0.48
Single Best	0.56	0.48	0.61	0.46	0.30	0.52	0.51	0.72	0.29	0.35	0.48
ToshCam [22]	0.59	0.40	0.60	0.33	0.17	0.63	0.53	0.77	0.31	0.37	0.47
KNN-Dist.	0.51	0.24	0.55	0.31	0.17	0.38	0.28	0.59	0.14	0.17	0.33
PRIPUVA	0.48	0.17	0.45	0.31	0.12	0.30	0.13	0.62	0.13	0.26	0.30

CHAPTER 8

CONCLUSION AND FUTURE WORK

In this dissertation, we have presented a complete and domain-independent visual analysis system to obtain high-level semantic representations of the visual content through performing a series of semantic approaches on low-level visual descriptors. The detection of high-level classes is investigated in multi-layered framework within an integrated system architecture, by training an initial set of binary classifiers, and then performing a novel classifier fusion algorithm on the outputs of individual classifiers.

A unique fuzzy learning approach is introduced in this study, which is not limited to visual features only, but can also be applicable to similar domains that have high-dimensional feature spaces. The membership degrees of visual features are calculated using class-based trained SOM neural networks and exploited in the classifier training process. This fuzzy learning approach in SVM training is brand new for image classification systems, and experimental results shows that even this new learning method can improve the performance of classical SVM in image applications.

The SOM neural network forms the basis for several computations throughout this dissertation, including low-level feature analysis, membership calculations, feature weighting and so on. The SOM structure is also used as a tool to generate the visual vocabularies in the BOW model, which is shown to be an effective method than the typical k-means clustering approach.

We also present a novel BOW approach for classifying images exploiting the distinctive local features in BOW model. We examine several aspects of the approach by performing several tests in the scope of visual analysis: the effect of distinctive features in different classification techniques, and the size of codebooks along with the utilization of SOM. The codebooks generated by SOM and k-Means are compared, in which the former gives superior performance results according to our experiments.

This approach in BOW utilizing SIFT features has two main contributions: 1) we improve the codebook generation process and build more informative codebooks by generating an individual codebook for each image class using the SOM method instead of building a global codebook for all classes, 2) we introduce the detection and use of distinctive features in BOW model, which significantly increases the classification performance of NB and KNN. Although it is not investigated in the context of this dissertation, the elimination of irrelevant image features is also important to build scalable systems for real-world applications, since the distinctive features constitute nearly 20% of the original image features

As future work, different membership methods might be investigated to utilize the fuzzy learning method, in which the fuzzy learning a negative impact on the classifier performance. Also the presented approaches in visual analysis might be used to complement or improve the performances of the previous studies, since the semantic models introduced in the dissertation are domain independent, and can be applied in different multimedia applications.

Considering the future work about our BOW approach described in Chapter 6, as a future work, the use of distinctive features can be investigated further in different image applications, and large image collections can be used to analyze the effect of the distinctive features in terms of computational efficiency without loosing the classification performance.

REFERENCES

[1] L. Parfeni. (2012). *Flickr Boasts 6 Billion Photo Uploads*. Available: http://news.softpedia.com/news/Flickr-Boasts-6-Billion-Photo-Uploads-215380.shtml (Last accessed 2012)

[2] *YouTube Statistics*. Available: http://www.reelseo.com/youtube-statistics/ (Last accessed 2012)

[3] P. Enser and C. Sandom, "Towards a comprehensive survey of the semantic gap in visual image retrieval," presented at the Proceedings of the 2nd international conference on Image and video retrieval, Urbana-Champaign, IL, USA, 2003.

[4] D. Deng, "Braving the semantic gap: mapping visual concepts from images and videos," presented at the Proceedings of the 4th international conference on Advances in Data Mining: applications in Image Mining, Medicine and Biotechnology, Management and Environmental Control, and Telecommunications, Leipzig, Germany, 2004.

[5] M. W. A. Smeulders, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years.," *IEEE Transactions on Pattern Analysis Machine Intelligence,* vol. 22, pp. 1349-1380, 2000.

[6] J. H. Kittler, M.; Duin, R.P.W.; Matas, J, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, pp. 226-239, 1998.

[7] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *International Journal of Machine Learning and Cybernetics,* vol. 1, pp. 43-52, 2010/12/01 2010.

[8] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.

[9] M. S. Huadong Wu , Mel Siegel (contact , Jie Yang , Rainer Stiefelhagen, "Sensor Fusion Using Dempster-Shafer Theory," *Proceedings of IEEE Instrumentation and Measurement Technology Conference,* pp. 21-23, 2002.

[10] M. W. Cees G. M. Snoek , Jan C. Van Gemert , Jan-mark Geusebroek , Arnold W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia " presented at the In Proceedings of the ACM International Conference on Multimedia, 2006.

[11] M. W. Arnold W. M. Smeulders , Simone Santini , Amarnath Gupta , Ramesh Jain, "Content-based image retrieval at the end of the early years," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 22, pp. 1349-1380, 2000.

[12] E. S. Marjo Markkula "End-User Searching Challenges Indexing Practices in the Digital Newspaper Photo Archive," *Information retrieval,* vol. 1, pp. 259-285, 1999.

[13] W. B. Kerry Rodden , David Sinclair , Kenneth Wood, "Does Organisation by Similarity Assist Image Browsing," presented at the Proceedings of the SIGCHI conference on Human factors in computing systems, 2001.

[14] Y. Alemu, Koh, J., and Ikram, M., "Image Retrieval in Multimedia Databases: A Survey," presented at the In Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2009.

[15] C. G. M. S. a. M. Worring, "Multimodal Video Indexing: A Review of the State-of-the-art," *Multimedia Tools and Applications,* vol. 25, pp. 5-35, 2005.

[16] I. K. Milind R. Naphade , Thomas Huang, "Probabilistic Semantic Video Indexing," *In Proceedings of Neural Information Processing Systems,* pp. 967-973, 2000.

[17] H. Eidenberger, "How good are the visual MPEG-7 features? ," *IEEE Visual Communications and Image Processing Conference,* pp. 476-488, 2003.

[18] I. I. JTC1/SC29/WG11. (2004). *MPEG-7 Overview (version 10)*. Available: http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm (Last accessed 2012)

[19] T. Sikora, "The MPEG-7 visual standard for content description - an overview," *IEEE Transactions On Circuits And Systems For Video Technology,* vol. 11-6, pp. 696-702, 2001.

[20] P. S. B.S. Manjunath, T. Sikora, *Introduction to MPEG-7: multimedia content description interface*: John Wiley and Sons, 2002.

[21] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," *Neural Networks, IEEE Transactions on,* vol. 10, pp. 1055-1064, 1999.

[22] J. W. M. Everingham, L. Van Gool, C. Williams, and A. Zisserman, "The PASCAL visual object classes challenge," *International Journal of Computer Vision,* vol. 88, pp. 303-338, 2010.

[23] A. T. B. C. Russell, K. P. Murphy, W. T. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision,* vol. 77, pp. 157-173, 2008.

[24] B. Yao, X. Yang, and S.-C. Zhu, "Introduction to a large-scale general purpose ground truth database: methodology, annotation tool and benchmarks," presented at the Proceedings of the 6th international conference on Energy minimization methods in computer vision and pattern recognition, Ezhou, China, 2007.

[25] R. Kohavi and F. Provost, "Glossary of terms," *Machine Learning,* vol. 30, pp. 271-274, 1998.

[26] R. M. M.S. Lotfabadi, "The Comparison of Different Classifiers for Precision Improvement in Image Retrieval," *Proceedings of the 2010 Sixth International Conference on Signal-Image Technology and Internet Based Systems,* pp. 176-178 2010.

[27] M. Zhu, "Recall, Precision and Average Precision," *Working Paper,* 2004.

[28] F. L. L. Zhang, and B. Zhang, "Support vector machine learning for image retrieval," *Proceedings of International Conference on Image Processing,* pp. 721-724, 2001.

[29] V. Vapnik, *The nature of statistical learning theory*: Springer-Verlag, 1995.

[30] J. B. Kilian Q. Weinberger , Lawrence K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *In NIPS*, ed: MIT Press, 2006.

[31] D. d. R. M. Egmont-Petersen, H. Handels, "Image processing with neural networks—a review," *Journal of Pattern Recognition,* 2002.

[32] M.-L. S. S.-C. Chen, C. Zhang, and M. Chen, "A Multimodal Data Mining Framework for Soccer Goal Detection Based on Decision Tree Logic," *International Journal of Computer Applications in Technology,* vol. 27, pp. 312-323, 2006.

[33] C.-J. L. V. S. Tseng, and J.-H. Su, "Classify by representative or associations (CBROA): A hybrid approach for image classification," *in Proc. 6th Int. Workshop on Multimedia Data Mining: Mining Integrated Media and Complex Data,* pp. 37–53, 2005.

[34] T. Bayes, and Price, Richard, "An Essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, communicated by Mr. Price, in a letter to

John Canton, M. A. and F. R. S," *Philosophical Transactions of the Royal Society of London,* vol. 53, pp. 370-418, 1763.

[35] T. F. Q. Douglas A. Reynolds , Robert B. Dunn, "Speaker verification using Adapted Gaussian mixture models," *Digital Signal Processing,* 2000.

[36] G. H. J. a. P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers," *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence,* pp. 338-345, 1995.

[37] M. I. Jordan, "Hierarchical mixtures of experts and the EM algorithm," *Neural Computation,* vol. 6, pp. 181-214, 1994.

[38] M. Collins, "Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms," pp. 1-8, 2002.

[39] B. B. Tahayna, M. ; Alhashmi, S.M. ; O'Daniel, T. , "Optimizing support vector machine based classification and retrieval of semantic video events with genetic algorithms," *17th IEEE International Conference on Image Processing (ICIP),* pp. 1485-1488, 2010.

[40] Y. S. a. S. Ozawa, "A hierarchical approach for region-based image retrieval," *IEEE International Conference on Systems, Man and Cybernetics,* vol. 1, pp. 1117-1124, 2004.

[41] G. T. C. Papadopoulos, K. ; Mezaris, V. ; Kompatsiaris, I. ; Izquierdo, E. ; Strintzis, M.G. , "A Comparative Study of Classification Techniques for Knowledge-Assisted Image Analysis," *Ninth International Workshop on Image Analysis for Multimedia Interactive Services,* pp. 4-7 2008.

[42] a. J. K. R.Eberhart, "A New Optimizer Using Particle Swarm Theory," *6th International Symposium on Micro Machine and Human Science,* vol. 4, pp. 39-43, 1995.

[43] R. S. Z. X. He, and M. A. Carreira-Perpinan, "Multiscale conditional random fields for image labeling," *IEEE Conf. Computer Vision and Pattern Recognition* pp. 695-702, 2004.

[44] H. H. Bannour, C. , "Towards ontologies for image interpretation and annotation," *International Workshop on Content-Based Multimedia Indexing (CBMI),* pp. 211-216, 2011.

[45] D. Z. Ying Liu, Guojun Lu, Wei-Ying Ma, "Asurvey of content-based imageretrieval with high-levelsemantics," *Pattern Recognition,* vol. 40, pp. 262-282, 2007.

[46] H. L. B. Evaggelos Spyrou , Theofilos Mailis , Eddie Cooke, "Fusing mpeg-7 visual descriptors for image classification," *Int. Conf. Artificial Neural Networks,* pp. 847-852, 2005.

[47] M.-y. C. Alexander G. Hauptmann, Michael G. Christel, Wei-Hao Lin, Jun Yang, "A Hybrid Approach to Improving Semantic Extraction of News Video," *A Hybrid Approach to Improving Semantic Extraction of News Video,* pp. 79-86, 2007.

[48] B. H. R. Benmokhtar, "Perplexity-based evidential neural network classifier fusion using mpeg-7 low-level visual features," *Proceedings of the 1st ACM international conference on Multimedia information retrieval,* pp. 336-341, 2008.

[49] A. A. A. Golestani, K.A. ; Jahed Motlagh, M.R., "A Novel Adaptive-Boost-Based Strategy for Combining Classifiers Using Diversity Concept," *6th IEEE/ACIS International Conference on Computer and Information Science,* pp. 128-134, 2007.

[50] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," presented at the Proceedings of the Second European Conference on Computational Learning Theory, 1995.

[51] R. H. Benmokhtar, B. ; Berrani, S.-A. , "Low-level feature fusion models for soccer scene classification," *IEEE International Conference on Multimedia and Expo,* pp. 1329-1332 2008.

[52] L. N. Masisi, V. ; Marwala, T. , "The use of entropy to measure structural diversity," *IEEE International Conference on Computational Cybernetics, ICCC 2008. ,* pp. 41-45, 2008.

[53] V. N. a. T. M. L. Masisi, "The effect of structural diversity of an ensemble of classiffiers on classification accuracy," *Johannesburg : Witwatersrand University,* 2008.

[54] L.-Y. Duan, "A Mid-level Representative Framework for Semantic Sports Video Analysis," *Proceedings of ACM Multimedia,* pp. 33-44, 2003.

[55] J. C. Platt, *Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods*: MIT Press, 1999.

[56] T. Kohonen, "The self organizing map," *Proceedings of IEEE,* vol. 78, pp. 1464–1480, 1990.

[57] S. A. Koskela M, "Clustering-based analysis of semantic concept models for video shots," *Proceedings of the international conference on multimedia and expo,* pp. 45-48, 2006.

[58] L. T. K. Chou, and I. Shyu, "Performances analysis of a multiple classifiers system for recognition of totally unconstrained handwritten numerals," *4th International Workshop on Frontiers of Handwritten Recognition,* pp. 480–487, 1994.

[59] Y. Z. Y. Liu, "One-against-all multi-class SVM classification using reliability measures," *IEEE International Joint Conference on Neural Networks,* pp. 849-854, 2005.

[60] Y. M. Deng, B.S., "Unsupervised segmentation of color-texture regions in images and video," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 800-810, 2001.

[61] H. Pan, Y. Zhu, and L. Xia, "Efficient and accurate face detection using heterogeneous feature descriptors and feature selection," *Comput. Vis. Image Underst.,* vol. 117, pp. 12-28, 2013.

[62] J. M. W. J. Shotton, C. Rother, A. Criminisi, "Texton-Boost: joint appearance, shape and context modeling for multiclass object recognition and segmentation," *In Proc. of the European Conference on Computer Vision,* pp. 1-15, 2006.

[63] X.-Y. Wang, X.-J. Zhang, H.-Y. Yang, and J. Bu, "A pixel-based color image segmentation using support vector machine and fuzzy C-means," *Neural Netw.,* vol. 33, pp. 148-159, 2012.

[64] Q. Ge, L. Xiao, J. Zhang, and Z. H. Wei, "An improved region-based model with local statistical features for image segmentation," *Pattern Recogn.,* vol. 45, pp. 1578-1590, 2012.

[65] C.-H. Chung, S.-C. Cheng, and C.-C. Chang, "Adaptive image segmentation for region-based object retrieval using generalized Hough transform," *Pattern Recogn.,* vol. 43, pp. 3219-3232, 2010.

[66] D. Zhang, M. M. Islam, and G. Lu, "A review on automatic image annotation techniques," *Pattern Recogn.,* vol. 45, pp. 346-362, 2012.

[67] D. D. Burdescu, C. G. Mihai, L. Stanescu, and M. Brezovan, "Automatic image annotation and semantic based image retrieval for medical domain," *Neurocomput.,* vol. 109, pp. 33-48, 2013.

[68] M. Lux, "Caliph & Emir: MPEG-7 photo annotation and retrieval," presented at the Proceedings of the 17th ACM international conference on Multimedia, Beijing, China, 2009.

[69] K. P. a. D. A. a. C. S. a. Y. K. a. S. Staab, "MOntoMat-Annotizer: Image annotation. linking ontologies and multimedia low-level features," *10th Intnl. Conf. on Knowledge Based, Intelligent Information and Engineering Systems,* 2006.

[70] B. L. T. C-Y Lin, J R Smith, "VideoAnnEx: IBM MPEG-7 Annotation Tool for Multimedia Indexing," *IEEE Intl. Conf. on Multimedia and Expo,* 2003.

[71] L. Z. a. B. Zhang, "Relationship between support vector set and kernel functions in SVM," *Journal of Computer Science and Technology,* vol. 17, pp. 549-555, 2002.

[72] J. A. Lee and M. Verleysen, "Self-organizing maps with recursive neighborhood adaptation," *Neural Netw,* vol. 15, pp. 993-1003, Oct-Nov 2002.

[73] S. Hasan and S. M. Shamsuddin, "Multistrategy self-organizing map learning for classification problems," *Intell. Neuroscience,* vol. 2011, pp. 1-11, 2011.

[74] E. Pampalk, G. Widmer, and A. Chan, "A new approach to hierarchical clustering and structuring of data with Self-Organizing Maps," *Intell. Data Anal.,* vol. 8, pp. 131-149, 2004.

[75] M. Koskela, J. Laaksonen, and E. Oja, "Entropy-Based Measures for Clustering and SOM Topology Preservation Applied to Content-Based Image Indexing and Retrieval," presented at the Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 2 - Volume 02, 2004.

[76] C. Saavedra, R. Salas, Sebastian. Moreno, H\, *et al.*, "Fusion of self organizing maps," presented at the Proceedings of the 9th international work conference on Artificial neural networks, San Sebastian, Spain, 2007.

[77] Y.-C. Liu, C. Wu, and M. Liu, "Research of fast SOM clustering for text information," *Expert Syst. Appl.,* vol. 38, pp. 9325-9333, 2011.

[78] D. Xi and S.-W. Lee, "Face Detection Based on Support Vector Machines," presented at the Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines, 2002.

[79] H. C. Z. G. G. L. S. Li, "A Novel Support Vector Machine Fuzzy Network for Image Classification Using MPEG-7 Visual Descriptors," *International Conference on MultiMedia and Information Technology,* pp. 365-368, 2008.

[80] D. Trung Le; Dat Tran ; Wanli Ma ; Sharma, "A new fuzzy membership computation method for fuzzy support vector machines," *Third International Conference on Communications and Electronics (ICCE)* pp. 153-157, 2010.

[81] A. Hauptmann, R. Yan, and W.-H. Lin, "How many high-level concepts will fill the semantic gap in news video retrieval?," presented at the Proceedings of the 6th ACM international conference on Image and video retrieval, Amsterdam, The Netherlands, 2007.

[82] a. S. W. C. Lin, "Fuzzy support vector machines," *IEEE Transactions on Neural Networks,* vol. 13, pp. 464-471, 2002.

[83] X. Z. X. Xiao, "An improved fuzzy support vector machine," *International Ubiqutious Computing and Education,* pp. 125-128, 2009.

[84] A. L. Shilton, D.T.H., "Iterative Fuzzy Support Vector Machine Classification," *IEEE International Fuzzy Systems Conference,* pp. 1-6, 2007.

[85] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery,* vol. 2, pp. 121-167, 1998.

[86] X.-l. B. X. Qian, "Medical Image Classification based on Fuzzy Support Vector Machines," *International Conference on Intelligent Computation Technology and Automation,* vol. 2, pp. 145-149, 2008.

[87] X. X. X. Zhang, "An Improved Fuzzy Support Vector Machine," *International Symposium on Intelligent Ubiquitous Computing and Education,* pp. 125-128, 2009.

[88] J. L. S. R. H. K. Qian, "Image Classification Based on Fuzzy Support Vector Machine," *International Symposium on Computational Intelligence and Design,* vol. 1, pp. 68-71, 2008.

[89] R. Rifkin and A. Klautau, "In Defense of One-Vs-All Classification," *Journal of Machine Learning Research,* vol. 5, pp. 101-141, 2004.

[90] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," 2009.

[91] G. Cybenko, "Approximations by superposition of a sigmoidal function," *Mathematics of Control, Signal and Systems,* vol. 2, pp. 303-314, 1989.

[92] B. M. F. Souvannavong, and B. Huet, "Multi modal classifier fusion for video shot content retrieval," *Proceedings of WIAMIS,* 2005.

[93] S. G, *A mathematical theory of evidence*: Princeton: Princeton University Press, 1976.

[94] G. J. a. M. J. W. Klir, *Uncertainty-Based Information: Elements of Generalized Information Theory*: Physica-Verlag, 1998.

[95] A. P. Dempster, "Upper and Lower Probabilities Induced by a Multivalued Mapping," *Studies in Fuzziness and Soft Computing,* vol. 219, pp. 57-72, 2008.

[96] R. Haenni, "Shedding new light on Zadeh's criticism of Dempster's rule of combination," *International Conference on Information Fusion,* vol. 2, 2005.

[97] M. Li, Sethi, I., "Confidence-Based Classifier Design," *Pattern Recognition,* vol. 39, pp. 1230–1240, 2006.

[98] S. M. W. Oleg Golubitsky "Confidence Measures in Recognizing Handwritten Mathematical Symbols," *International Conference on Intelligent Computer Mathematics,* pp. 460-466, 2009.

[99] C. B. a. S. Hartmann, *Probability in Physics*: Oxford University Press, 2011.

[100] C. E. Shannon, "A Mathematical Theory of Communication," *Bell System Technical Journal* vol. 27, pp. 379–423, 1948.

[101] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *In NIPS*, 2007.

[102] Y. G. Jiang, J. Yang, C. W. Ngo, and A. G. Hauptmann, "Representations of Keypoint-Based Semantic Concept Detection: A Comprehensive Study," *Multimedia, IEEE Transactions on,* vol. 12, pp. 42-53, 2009.

[103] L. Hui-lan, W. Hui, and L. Loi-Lei, "Creating Efficient Visual Codebook Ensembles for Object Categorization," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 41, pp. 238-253, 2011.

[104] D. Lowe, "Object Recognition from Local Scale-Invariant Features," in *Proc. of the International Conference on Computer Vision ICCV, Corfu*, 1999, pp. 1150-1157.

[105] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," *IJCV,* vol. 73, pp. 213-238, 2007.

[106] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, New York, NY, USA, 2006, pp. 2169-2178.

[107] J. Yang, Y. Jiang, A. Hauptmann, and C. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *MIR '07: Proceedings of the international workshop on Workshop on multimedia information retrieval*, Augsburg, Bavaria, Germany, 2007, pp. 197-206.

[108] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-Up Robust Features (SURF)," *Comput. Vis. Image Underst.,* vol. 110, pp. 346-359, 2008.

[109] Y. Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," in *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Washington, DC, USA, 2004, pp. 506-513.

[110] Y. Jiang, C. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proceedings of the 6th ACM international conference on Image and video retrieval*, Amsterdam, The Netherlands, 2007, pp. 494-501.

[111] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, San Diego, CA, USA, 2005, pp. 886-893 vol. 1.

[112] C. Elkan. (1997). *Boosting and Naive Bayesian Learning*. Available: http://pages.cs.wisc.edu/~dyer/cs540/handouts/elkan97boosting.pdf (Last accessed 2012)

[113] A. Kibriya and E. Frank, "An Empirical Comparison of Exact Nearest Neighbour Algorithms," ed, 2007, pp. 140-151.

[114] Sivic and Zisserman, "Video Google: a text retrieval approach to object matching in videos," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Nice, France, 2003, pp. 1470-1477 vol.2.

[115] K. Mikolajczyk and C. Schmid. (2005). *A PERFORMANCE EVALUATION OF LOCAL DESCRIPTORS*. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.5507 (Last accessed 2012)

[116] D. Nister and H. Stewenius, "Scalable Recognition with a Vocabulary Tree," 2006, pp. 2161-2168.

[117] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval," in *2007 IEEE 11th International Conference on Computer Vision*, Rio de Janeiro, Brazil, 2007, pp. 1-8.

[118] S. Z. a. Q. T. a. G. H. a. Q. H. a. W. Gao, "Generating Descriptive Visual Words and Visual Phrases for Large-Scale Image Applications," *Image Processing, IEEE Transactions on,* vol. 20, pp. 2664-2677, 2011.

[119] R. J. López-Sastre, T. Tuytelaars, F. J. Acevedo-Rodríguez, and S. Maldonado-Bascón, "Towards a more discriminative and semantic visual vocabulary," *Computer Vision and Image Understanding,* 2010.

[120] P. Tirilly, V. Claveau, and P. Gros, "Language modeling for bag-of-visual words image categorization," in *Proceedings of the 2008 international conference on Content-based image and video retrieval*, Niagara Falls, Canada, 2008, pp. 249-258.

[121] T. de Campos, G. Csurka, and F. Perronnin, "Images as Sets of Locally Weighted Features," *Computer Vision and Image Understanding,* 2011.

[122] T. Villmann, R. Der, M. Herrmann, and T. M. Martinetz, "Topology preservation in self-organizing feature maps: exact definition and measurement," *Neural Networks, IEEE Transactions on,* vol. 8, pp. 256-266, 1997.

[123] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision,* vol. 88, pp. 303-338, 2010.

[124] U. L. Altintakan, A. Yazici, and M. Koyuncu, "A novel fuzzy visual object classification approach," presented at the IEEE Int. Conf. on Fuzzy Systems, 2012.

[125] V. Viitaniemi and J. Laaksonen, "Techniques for Image Classification, Object Detection and Object Segmentation," in *Visual Information Systems. Web-Based Visual Information Search and Management*. vol. 5188, M. Sebillo, G. Vitiello, and G. Schaefer, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 231-234.

[126] X. Liu, D. Wang, J. Li, and B. Zhang, "The feature and spatial covariant kernel: adding implicit spatial constraints to histogram," presented at the Proceedings of the 6th ACM international conference on Image and video retrieval, Amsterdam, The Netherlands, 2007.

[127] F. Perronnin and C. Dance, "Fisher Kernels on Visual Vocabularies for Image Categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, 2007, pp. 1-8.

[128] M. M. a. C. S. a. H. H. a. J. v. d. Weijer, "Learning Object Representations for Visual Object Class Recognition," presented at the Visual Recognition Challange workshop, in conjunction with ICCV, 2007.

[129] J. Zhang, M. Marszaek, S. Lazebnik, and C. Schmid, "Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study," *Int. J. Comput. Vision,* vol. 73, pp. 213-238, 2007.

[130] F. a. D. Perronnin, Christopher R., "Fisher Kernels on Visual Vocabularies for Image Categorization," in *IEEE conference on computer vision and pattern recognition*, 2007.

[131] K. E. A. a. G. van de Sande, T. and Snoek, C. G. M., "Evaluation of Color Descriptors for Object and Scene Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[132] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1-8.

VITA

Ümit Lütfü Altıntakan was born in Nevşehir on April 3, 1978. He received the B.S. and M.S degrees in Computer Engineering Department of METU in 2001 and 2005, respectively. He worked at Turkish Armed Forces Rehabilitation Center from 2001 to 2007 as software engineer in healthcare systems. From 2007 to 2012, he worked at Turkish Air Force Headquarters in enterprise resource management systems as team leader and project manager in various functional areas including human resource management, material management, logistics and finance. He is assigned to a multinational military post in NATO E3A Air Base, in Geilenkirchen/Germany on August 2012. From then, he has been working on Surveillance Radar Operational Computer Program for Airborne Early Warning Air Crafts (AWACS) as project lead and software analyst. He also worked in a number of Scientific and Technological Research Council of Turkey (TUBITAK) projects in the field of visual analysis. His research interests include database management systems, image classification and understanding, semantic analysis of visual data, content based image retrieval and semantic healthcare systems.

PUBLICATIONS

1. Ü.L. Altıntakan, A. Yazıcı, and M. Koyuncu, "A novel fuzzy visual object classification approach", in *IEEE Int. Conf. on Fuzzy Systems (FUZZ-IEEE)*, pp. 1-6, 2012.

2. Ü.L. Altıntakan, A. Yazıcı, and M. Koyuncu, "A neural network based visual analysis system using fuzzy learning and high-level fusion for semantic concept extraction", in *Imaging Science Journal SI on Computational Intelligence in Image Processing*. (Manuscript submitted on April 2012 and still under review)

3. Ü.L. Altıntakan, "High-level classifier fusion using SVM and Dempster-Shafer theory", *2nd International Conference on Computing and Computer Vision*, 2013.

4. Ü.L. Altıntakan and A. Yazıcı, "An effective visual classification system using distinctive image features and self-organizing maps", in *IEEE Transactions on Multimedia*. (Manuscript submitted September 2013.)