A HIERARCHICAL OBJECT LOCALIZATION AND IMAGE RETRIEVAL FRAMEWORK

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

BY

MUTLU UYSAL

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

MARCH 2006

Approval of the Graduate School of Natural and Applied Sciences.

_____

Prof. Dr. Canan Özgen

Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Doctor of Philosophy.

_____

Prof. Dr. Ayşe Kiper

Head of Department

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

_____

Prof. Dr. Fatoş Yarman-Vural

Supervisor

Examining Committee Members

Prof. Dr. Adnan Yazıcı (METU, CENG) _____

Prof. Dr. Fatoş Yarman Vural (METU, CENG) _____

Prof. Dr. Volkan Atalay (METU, CENG) _____

Assoc. Prof. Dr. Gözde Bozdağı Akar (METU, CENG) _____

Dr. Pınar Duygulu Şahin (BİLKENT UNV., CS) _____

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

Name, Last name: Mutlu Uysal

Signature:

ABSTRACT

A HIERARCHICAL OBJECT LOCALIZATION AND IMAGE RETRIEVAL FRAMEWORK

Uysal Mutlu
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Fatos Yarman-Vural

March 2006, 106 pages

This thesis proposes an object localization and image retrieval framework, which trains a discriminative feature set for each object class. For this purpose, a hierarchical learning architecture, together with a Neighborhood Tree is introduced for object labeling. Initially, a large variety of features are extracted from the regions of the pre-segmented images. These features are, then, fed to the training module, which selects the "best set of representative features", suppressing relatively less important ones for each class.

During this study, we attack various problems of the current image retrieval and classification systems, including feature space design, normalization and curse of dimensionality. Above all, we elaborate the semantic gap problem in comparison to human visual system. The proposed system emulates the eye-brain channel in two layers. The first layer combines relatively simple classifiers with low level, low dimensional features. Then, the second layer implements Adaptive Resonance Theory, which extracts higher level information from the first layer. This two-layer architecture reduces the curse of dimensionality and diminishes the normalization problem.

The concept of Neighborhood Tree is introduced for identifying the whole object from the over-segmented image regions. The Neighborhood Tree consists of the nodes corresponding to the neighboring regions as its children and merges the regions through a search algorithm. Experiments are performed on a set of images from Corel database, using MPEG-7, Haar and Gabor features in order to observe the power and the weakness

of the proposed system. The "Best Representative Features" are found in the training phase using Fuzzy ARTMAP [1], Feature-based AdaBoost [2], Descriptor-based AdaBoost, Best Representative Descriptor [3], majority voting and the proposed hierarchical learning architecture.

During the experiments, it is observed that the proposed hierarchical learning architecture yields better retrieval rates than the existing algorithms available in the literature.

Keywords: Object localization, image retrieval, Neighborhood Trees, Hierarchical learning, Best Representative Descriptor, FUZZY ARTMAP, ADABOOST.

ÖZ

KATMANLI BİR MİMARİ İLE
NESNE BULMA VE İMGE SORGULAMA SİSTEMİ

Uysal Mutlu
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Fatos Yarman-Vural

Mart 2006, 106 sayfa

Bu tezde; sorgulanacak olan her imge sınıfı için, "en uygun" öznitelik vektörünü elde eden, nesne temelli bir sorgulama sistemi önerilmektedir. Geliştirilen sistemde, "en uygun" öznitelik vektörünü bulmak için, katmanlı bir öğrenme mimarisi; imgelerin düzgün etiketlenebilmesi için ise “komşuluk ağacı” kavramı önerilmektedir. Bu kapsamda, ilk olarak, otomatik olarak bölütlenmiş imgelerden, öznitelik vektörleri çıkartılmaktadır. Daha sonra, her bir imge sınıfını diğer sınıflardan ayırdedecek, öznitelik ağırlık vektörleri, önerilen mimari kullanılarak bulunmaktadır.

Bu tezde geliştirilen katmanlı öğrenme metodu, içeriğe dayalı imge sorgulama sistemlerinin öznitelik uzayı tasarımı, normalizasyon, boyut artması (curse of dimensionality) ve anlamsal farklılık (semantic gap) gibi temel problemlerini azaltmaya yöneliktir. Bu doğrultuda, göz ve beyin arasındaki ilişkiden yola çıkılarak, iki katmanlı bir sınıflandırma yöntemi önerilmiştir. Birinci katmanda, farklı sınıflandırma araçları, vektör uzayının farklı bölümlerini kullanılarak sınıflandırmalar yapmakta; bir üst katmanda ise, elde edilen sonuçlar birleştirilmektedir. Bu yöntem, öznitelik vektörlerinin ardarda eklenmesiyle elde edilen, çok boyutlu vektörün yol açtığı problemleri, azaltmaya çalışmaktadır.

Önerilen sistem, aynı zamanda, bölütlenmiş imgelerde yer alan nesnelerin etiketlenebilmesi için, komşuluk ağacı kavramını tanımlamaktadır. Komşuluk ağacı,

imge içinde birbiriyle komşu olan bölütlerin, birleştirilmesi ile oluşan bir gösterim biçimidir.

Geliştirilen sistemin performansını denemek için, Corel veri kümesi üzerinde deneyler yapılmıştır. Eğitim modülünde, bir sınıfı en iyi temsil eden özniteliklerin bulunması için önce bulanık ARTMAP ve ADABOOST yöntemleri kullanılmış, daha sonra ise iki katmanlı bir mimari geliştirilerek performans arttırılmıştır.

Yapılan deneyler sonucunda, önerilen katmanlı mimarinin diğerlerinden daha iyi sonuçlar verdiği gözlemlenmiştir.

Anahtar Kelimeler: Nesneye Dayalı İmge Sorgulama, Komşuluk Ağacı, Katmanlı sınıflandırma, En iyi temsil eden Öznitelik, Bulanık ARTMAP, ADABOOST.

To My Family

ACKNOWLEDGMENTS

I express sincere appreciation to Prof. Dr. Fatoş Yarman Vural for her guidance and support. She was always helpful to me. I would like to thank to all of my friends for their encouragement and support. Finally I would like to thank to my family. They have given me encouragement to complete this thesis. I love all of them very much.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER

1 INTRODUCTION

It is well-known that the performance of today's image retrieval and classification systems are far from the user's expectations due to the naiveness of the available models to simulate the human visual system. The complex nature of eye-brain channel, which processes the visual information at the low level in the eye and then transmits it to the brain for higher levels of processing, is still far more complicated then the existing mathematical tools. The major problems of the image classification systems can be analyzed under four major headings as follows:

1. Design of the feature space,
2. Normalization of the feature vector,
3. Curse of dimensionality,
4. Semantic gap.

Let us start by a brief discussion of these well-known problems and try to develop possible roadmaps to bring partial solutions to some of them:

1.1 The Art of Designing Feature Space

There is no systematic approach or methodology to design the feature space. Image Analysis literature is full of features, developed for the specific problem domain. Eigenface, Laplacianface and Fisherface are examples of features for face recognition systems [4]. Horizontal and vertical histograms, curvature information, topological features such as loops, end-points or contour information can be used for optical character recognition systems [5]. Co-occurrence matrix, coarseness, contrast, directionality, line-likeness, regularity and roughness and Gabor transform features can

be used in many application domains such as finger print and texture analysis [6]. Color histogram, color moments, color sets, color correlogram, color layout, scalable color, Wavelet and Haar transform, boundary-based and region-based shape features are commonly used for scene analysis and object localization [7]. These features are developed by analysing the nature of the image database and the need of the problem domain. However, none of the proposed feature spaces are unique, nor they obtain the "best" performance for the specified problem. A feature space, which results in a "best" performance on a data set may totally go astray for another data set, depending on the image content, even for the same problem domain.

In most of the image recognition systems, the feature set, called descriptor, is formed by measuring some low level visual information based on color, texture or shape. In image processing literature, there is a wide range of descriptors for Image Analysis problems. Some of these descriptors have been standardized by the Moving Picture Experts Group (MPEG) [7]. Although MPEG-7 provides a variety of descriptors, design of the feature space by selecting the "good ones" among these descriptors is an open research issue.

One of the most important steps of a recognition system is the design of the feature space. This task requires to make some measurement on the image database and to extract information as compact as possible, so that the objects from the same class have "similar" values and objects from different class have "dissimilar" values according to a metric. It is clear from this statement that definition and/or selection of a metric is another crucial problem in recognition systems.

It is not easy to define a mathematical metric, which is consistent to human visual system. There are many similarity metrics such as Minkowsky [8], Mahalanobis [9], Camberra [10], Quadratic [11], Correlation [12], Chi-square [12] distance metrics; the Context-Similarity measure; the Contrast Model and hyperrectangle distance functions [13]. Although there have been many metrics proposed in the literature, the most commonly used one is the Euclidean Distance [14]. It should be noted that two vectors, which are very similar according to the metric mentioned above, may belong to different classes. Also, two samples from the same visual appearance can be considered very dissimilar according to distance metric.

Concatenation of descriptors, such as curvature, color, energy, edge frequency etc., in a feature vector generates a feature space, which is not uniform under the same similarity metric. For example, for some features Euclidean distance may successfully measure the class similarity, for some others Mahalanobis distance may be a better metric. Since each descriptor has its own characteristics, the semantic information may be lost when it is concatenated by an unrelated feature. Therefore, concatenation of the feature sets and using the same metric for them may have no sense.

Most of the time, the number of the features in each descriptor is not the same. Concatenation of the descriptors results in domination of the higher dimensional descriptor in the training stage, reducing the relative importance of the lower dimensional descriptor. For example, 80 dimensional edge histogram dominates the 3 dimensional color descriptor, even if the color is an important feature to distinguish the classes with dominant color characteristics. In summary, concatenation of descriptors under the same feature vector brings many problems.

1.2 Normalization

In most of the image classification systems [15], [16], [17], feature space is formed by concatenation of features of different types or ranges. Then, some normalization techniques are used to bring the incompatible features to the same dynamic range. As an example, suppose that an application has two attributes *A* and *B,* where *A* has values from 1 to 1000, and *B* has values only from 1 to 10. Then, in this system, the influence of *B* on the distance function is usually overpowered by *A*. In order to solve this problem, the feature is often *normalized* by dividing it by the *range* of it, so that the value for each feature is in the same approximate range, for example in [0,1]. In order to avoid outliers, it is also common to divide the feature by the standard deviation, instead of the range. Trimming the range by removing the highest and lowest few percent of the data may be useful in some applications [13], where the minimum or maximum values outside a certain range are mapped into the pre-defined interval.

However, approaches mentioned above may yield a distorted feature space by uncontrolled and undesired changes in the characteristics of the features. For example,

the values in a gray scale image can be in the range [0,255], where each integer in this interval corresponds to an intensity value. After the normalization process, these values are mapped to [0,1] interval. In the normalized feature space, the Euclidean distance between black and white is 1. On the other hand, if we map color histogram to the interval of [0,1], then the Euclidean distance between the most frequent color and a non-existing color is 1. These two measures, which are the same according to Euclidean distance, have no semantic similarity. Therefore, normalization should be avoided, as much as possible [19].

1.3 Curse of Dimensionality

In many applications such as face, fingerprint, optical character recognition, geographic and molecular biology data analysis, data sets with hundreds or even thousands of features are very common. For the representation of distributions in the high-dimensional feature spaces, there are hardly ever enough samples. According to a rule of thumb, for the estimation of class distributions, the number of samples should be in the order of magnitude higher than the number of dimension of the feature space [20]. This fact is called "Curse of Dimensionality" in the literature, where the required number of samples increases exponentially with the dimension to maintain an acceptable level of accuracy and statistically stability.

The curse of dimensionality problem may be partially, because of the irrelevant and redundant features in the feature vector or because of the insufficient number of the training data. A feature is redundant if another one in the feature space can be used instead of it, without any performance loss. A feature is irrelevant if it has no effect on the classification performance. Selecting a subset of features from the feature space, which eliminates redundant and irrelevant ones and obtaining the "best" set of features in terms of class separability is a common problem in pattern recognition, which is called feature subset selection.

Generally, a feature evaluation criterion and a search algorithm are used to select a feature subset in a large variety of features. The criterion is used to evaluate the performance of possible feature subsets. A search algorithm explores a feature space,

which has a "reasonable" performance. There are two major feature selection methods, namely, filter and wrapper methods [21]. The wrapper methods evaluate feature subsets based on classification results on the training data, while the filter methods use class separability measures as the criterion for feature subset evaluation. After selecting the evaluation criterion, the next step is to choose a search algorithm.

Exhaustive search methods guarantee to find an optimal solution, but it evaluates too many combinations, which can not be solved in polynomial time. Therefore, exhaustive search is rarely used in practice. Methods such as Branch and Bound [22] can be used to find optimal feature space. However, these methods may also converge to exhaustive search if they can not bind the solution. Suboptimal greedy based search algorithms, like sequential forward selection (SFS) and sequential backward elimination (SBE) are mainly used in practice. The disadvantage of SFS and SBE algorithms is that once a feature is selected, it can not be deleted in later stages. This causes selection of redundant features in the feature space. In order to eliminate the drawbacks of these algorithms, methods like the plus- l–take-away-r are proposed in the literature [21].

The authors studied how to select "good" features according to the maximal statistical dependency criterion based on mutual information, in [23]. Because of the difficulty in directly implementing the maximal dependency condition, they derived an equivalent form, called minimal-redundancy-maximal-relevance criterion (mRMR), for incremental feature selection.

Applying genetic algorithms to the feature selection problem is quite straightforward: The chromosomes of the individuals contain one bit for each feature. The value of the bit determines whether the feature will be used in the classification or not. The feature subset is formed by selecting the bits indicated by the chromosome [24]. Many variations of wrapper methods using genetic algorithm, Neural Networks, decision trees are proposed in the literature [24], [25], [26].

AdaBoost is a popular incremental wrapper method to obtain a feature subset from the large set of features [2]. In each step of AdaBoost, the feature with the best performance is selected and a higher weight is put on the miss-classified training data. This weight enables the features, to focus on the misclassified examples, in the next iterations. It is proved that AdaBoost minimizes the margin between positive and

negative examples. However, it is, also, shown that it has the tendency to memorize the training data. Therefore, the performance of the AdaBoost may significantly fall in the test data.

A wrapper method, using a constructive Neural Network, is utilized to select a feature subset, in [27]. This method designs the feature space by incrementally adding hidden nodes and weights to the network during training until a satisfactory solution is found. In doing so, the process of determining the size and the architecture of the network is intertwined with the learning process.

It should be noted that there is no guarantee that methods mentioned above selects `the best` set of features for separability of the feature space. Decreasing the dimension and obtaining `the best` sub-space without decreasing the classification accuracy needs an exhaustive search process.

1.4 Semantic Gap

One of the most crucial problems in Image Analysis systems is the 'semantic gap' between low-level features and high-level concepts of class labels, which creates a serious mismatch between the capabilities of current recognition systems and the user needs. Human beings evaluate similarity according to highly complex visual and cognitive process in the eye-brain channel. On the other hand, current recognition systems evaluate similarity, based on numerical feature extraction and distance measurement. The numerical values of low-level features do not satisfy the users high level semantic concepts and similarity expectations. A red ball and a red apple can be considered same by the current recognition systems, while they have very dissimilar semantic meaning according to human beings. In order to improve the recognition accuracy, approaches need to be developed for narrowing the gap between the low-level features and the richness of the user semantics.

There are several attempts to reduce the semantic gap problem: Recently, relevance feedback approaches have been an active research direction to bridge the gap between the user needs and available recognition methods [28], [29], [30]. Relevance feedback allows the Content-Based Image Retrieval (CBIR) systems to retrieve images

interactively. This approach enables one to select the semantically similar and dissimilar images at each iteration. The selected images guide the system to correct the retrieval results. Iterative refinement by relevance feedback reduces the semantic gap in CBIR systems. The high level concepts and perception subjectivity can be captured by the support of the user to some degree.

In [31], the authors introduced cluster-based retrieval of images by unsupervised learning (CLUE) to tackle the semantic gap problem. CLUE is built on a hypothesis that images of the same semantics tend to be clustered. It attempts to narrow the semantic gap by retrieving image clusters based on the feature similarity of images to the query. Additionally, the approach takes into account the within cluster similarity.

Ensemble of classifiers attacks the semantic gap problem by learning a meta-level classifier to combine the predictions of multiple base-level classifiers. In the base level, a series of recognition engines are trained by different low-level features, called descriptors. Then, the prediction of each recognition engine is combined at the meta-level. Several approaches for generating base-level classifiers are proposed in the literature. One approach is to generate classifiers by applying different learning algorithms to a single data set. Another possibility is to apply a single learning algorithm with different parameter settings to a single data set. Finally, methods like bagging [32] and boosting [2] generate multiple classifiers by applying a single learning algorithm to different versions of a given data set. Two different methods for manipulating the data set are used: Random sampling with replacement in bagging and re-weighting of the misclassified training examples in boosting.

Boosting and bagging generate training sets by resampling from the original data set to the learning algorithm, which builds up a base classifier for each training set. There are two major differences between bagging and boosting. Firstly, boosting changes adaptively the distribution of the training set, based on the performance of previously created classifiers, while bagging changes the distribution of the training set randomly. Secondly, boosting uses a function of the performance of a classifier as a weight for voting, while bagging uses equal weight voting [32].

Techniques for combining the predictions obtained from the multiple base level classifiers suffer due to the lack of a sound method to combine the results of the base-

level recognition engines. For this purpose, voting and stacking are employed to label the samples. In voting, each base-level classifier gives a vote for its prediction. The prediction, receiving most of the votes, is selected as the final prediction. In stacking, an algorithm is used to learn how to combine the predictions of the base-level classifiers. There are also techniques, which perform weighted majority voting and use the learned weights of each classifier during voting [33]. Some other techniques include minimum, maximum, summing, averaging etc. operations to find the final prediction of the classifiers.

The methods for reducing the semantic gap between the low level image features and complex needs of vision problems are still far from the needs of content based image retrieval systems. This issue is a hot research area in computer vision, artificial intelligence and image analysis area.

1.5 The Roadmap of This Thesis

In order to attack the major problems of CBIR systems mentioned above, in this thesis, we propose a hierarchical learning system based on Adaptive Resonance Theory (ART), introduced by Stephen Grossberg in 1976 [34]. ART is a neural theory of human and primate information processing in the brain. The theory tries to answer the question: How does brain encode information? The theory, also, brings a solution to the stability-plasticity dilemma, which can be defined as the problem of being able to learn new things continuously, quickly and stably without catastrophically forgetting the past knowledge. The principles, which led to the introduction of ART, were derived from an analysis of experimental literature in human visual perception and reinforcement learning, including attentional blocking and cognitive-emotional interactions [34]. There are many architectures to implement the Adaptive Resonance Theory, including ART, ART-1, ART-2, Fuzzy ART, ARTMAP [1].

The available architectures, for Adaptive Resonance Theory, receive a set of concatenated low level normalized features and act as a wrapper by suppressing the features, which do not have discriminative power by assigning them low membership values, in the learning stage. We believe that this utilization of Adaptive Resonance

Theory bears the problems of semantic gap, curse of dimensionality and normalization, as mentioned above. Additionally, the low-level normalized input features are contradictory to the basic idea of Adaptive Resonance Theory, which deals with high level semantic information. Finally, the ART architectures in the literature lack the hierarchical structure of eye brain channel, omitting the low-level information processing in the eye.

In this thesis, we try to emulate the eye-brain channel by a two-layer stack generalization method. The base-layer partially emulates the eye by extracting and learning a set of low level and low dimensional color, texture and shape features. Each low level feature set describes a certain visual phenomenon. For this reason, the individual feature sets are called descriptors. After the feature extraction step, each of the descriptors is received by a different classifier. The proposed approach does not involve concatenation of descriptors. Therefore, there is no need for normalization. The dimension of each feature set corresponding to a descriptor, varies depending on the type, image resolution or the application domain. However, each descriptor spans a different vector space. Therefore, the dimensions of the feature spaces, spanned by the individual descriptors are much less than the dimension of the feature spaces spanned by the vectors, obtained by concatenating all the features. As a result, curse of dimensionality problem is mostly avoided.

The meta-layer, proposed in this thesis, partially emulates the associative memory of human brain by implementing the Adaptive Resonance Theory. For this purpose, Fuzzy ARTMAP architecture, proposed by Carpenter et.al. in 1992 [1], is used. Fuzzy ARTMAP architecture at the meta-layer enables us to utilize the high level information generated by the base-layer classifiers, resulting a decrease in the semantic gap problem. Experiments indicate that two-layered approach, proposed in this thesis, improves the performance of CBIR systems.

It is well-known that the human-being start learning objects and their characteristics from their birth and continue this learning process incrementally, during all periods of their life. Although how brain achieves this ability is not very well-known, it is accepted that the process of visual perception is a result of supervision. Based on

this fact, the developed system is designed by employing supervision and training the characteristics of objects.

The major goal of this study is to develop an object-based image retrieval framework. In order to retrieve objects from an image database, there are two popular approaches proposed in the literature [16], [35]. The first one is to use a sliding window over the images to localize objects. Finding the appropriate size and shifting amount of the sliding window is the major problem in this approach. The second approach is to segment images for extracting the objects. However, it is well known that segmentation algorithms yield either over-segmentation or under-segmentation. The first approach extracts more than one region corresponding to a single object, where the latter one merges parts of different objects in a single region. In order to retrieve objects from the segmented images, one should propose an algorithm, which extracts objects from the regions.

Combining all possible neighboring segments exhaustively can be a solution alternative for object localization. However, this solution is rarely applied in practice, due to the fact that it can not be solved in polynomial time. Applying a greedy-based heuristic algorithm for combining the neighboring segments is an alternative to exhaustive search. However, the greedy based algorithms result in suboptimal solutions. In most of the cases, these algorithms stop merging the regions without localizing the complete query objects in the image database.

In order to solve the object localization problem mentioned above, in this thesis, we introduce the concept of "Neighborhood Tree" for representing the contiguous regions having the same label. Then, object localization by region merging becomes an optimization problem, which can be solved by a search algorithm over the tree.

In summary, this thesis deals with developing an object-based image retrieval framework, which uses a hierarchical learning schema and introduces a Neighborhood Tree for object localization problem. The thesis is organized as follows: In order to establish the necessary background, Chapter 2 gives a literature survey for the fundamental concepts of object classification and content based image retrieval. In Chapter 3, a general overview of the proposed image retrieval system is introduced. Chapter 4 elaborates the details of the contributions in the proposed retrieval system,

including hierarchical learning architecture and Neighborhood Tree. Chapter 5 gives the experimental results and finally, Chapter 6 concludes the thesis and gives the directions for future work.

CHAPTER

2 BACKGROUND FOR OBJECT LOCALIZATION AND CONTENT BASED IMAGE RETRIEVAL SYSTEMS

This chapter summarizes the literature for object localization and content based image retrieval systems. Firstly, main concepts of object localization will be elaborated. Secondly a literature survey on CBIR systems will be presented. Special attention is given to Automatic Linguistic Indexing of Picture Libraries (ALIP) system of [36] and Object Recognition by Machine Translation of [15], since they provide us a sound base for comparison. Finally training schemas used in the training phase of the proposed system will be overviewed.

2.1 Object Localization

This section discusses the approaches for object localization. A summary of the classification methods used in this study is also presented. Lastly, combination of different classifiers is disputed.

2.1.1 Discussion on Object Localization

Object localization deals with the assignment of a visual data of an object to one of several prespecified categories with a set of images in a database. Since it emerged in late 1960's, automatic localization of objects by machines has been one of the challenging problems of pattern recognition and machine learning communities. This section overviews the four popular approaches for object localization, which are:

1. Sliding Window Approach,
2. Block-based Approaches,

3. Segmentation-based Approaches,
4. Object-part-based Approaches.

Sliding a window over the images is a popular approach to solve the object localization problem. In such an approach, objects are selected in bounding boxes and trained using standard machine learning techniques, like support vector machines [35]. The querying phase involves sliding windows with varying sizes and shifting amounts over the images. The objects included in the sliding window are labeled by using the trained recognition engine. The difficulty in this approach is to determine the optimal size of the sliding window and the shifting amount.

Methods, which divide an image into rectangular blocks and perform object localization based on block information, are also proposed in the literature [36], [37], [38]. The logic behind this approach is to exploit local and spatial properties in the images using block information. In [37], the authors propose a system, which divides images into 16 non-overlapping equal-sized blocks, each of whose dominant orientations are computed. The block information is, then, used to classify "city" and "suburb" scenes. In [38], an image classification algorithm is introduced, where Wavelet coefficients in high frequency bands are computed for each block to classify graph/photograph images. In [36], Wang et. al. propose an Automatic Linguistic Indexing of Picture Libraries approach (ALIP), which initially partitions categories of images into blocks. Then, a two-dimensional multiresolution HMM is trained by color and texture features of blocks of each category. In [39], a graphical model is presented, which relates features of image blocks to objects and performs joint scene and object recognition. The difficulty in all of these block-based approaches is to determine the optimal size of the blocks.

Segmentation is another popular approach to localize objects in the images. The major examples of this approach are NeTra [40], Blobword [16] and Color Region Templates [41]. These methods assume that semantically meaningful segmentation is done accurately. Unfortunately, this assumption is not valid in most of the applications of computer vision. On the other hand, shape features can only be used in segmentation based algorithms, without which is quite difficult to localize objects.

In order to reduce the effect of inaccurate segmentation, Li et. al. put forward an integrated region matching (IRM) scheme for CBIR in [42]. IRM measure allows (for) matching a region of one image to several regions of another image. Note that mapping of the regions between any two images is a many-to-many relationship.

Another popular approach for object localization is called object-part based methods. In this approach, parts, which constitute the whole object, are identified [43]. The appearance of individual parts and the geometric relations between them are modeled. As an example, face consists of eye, nose, and mouth, where the eyes are symmetric above the nose and mouth. The major disadvantage of these methods is being not tolerant to the absence of critical parts that constitute the objects. It is impossible to recognize the face if the eyes are misclassified in such an approach.

In [44], Fergus et. al. propose a method to learn and localize objects from unlabeled and unsegmented cluttered scenes in a scale invariant manner. In their study, an object consists of a number of parts. The process of learning an object category has two steps. The first one is to detect regions and their scales, whereas the second one is to estimate the parameters of the object model. Recognition is performed on a query image by also detecting regions and their scales, and then by evaluating the regions using the model parameters estimated in the learning.

In summary, object localization is still one of the most challenging problems in computer vision. Although there are many proposed methods in the literature, we believe that the science is at the beginning of the road for fully automatic object localization systems.

2.1.2 Classification Methods for Object Localization

The major goal of object classification systems is to determine decision boundaries in the feature space, which increases the classification performance. Feature points among the decision boundaries form the decision regions. If the decision regions are too rigid and precise, problems like over-fitting can occur. If a classifier has over-fitting problem, although the performance of the classifier is nearly 100% for the training set, the performance decreases drastically for the test set. The reason of the decrease is

memorizing the training samples in finding the decision boundaries. The opposite of this problem is under-fitting, which means that the system could not even learn the training samples. Consequently, the performance in both training and test sets are too low. The classifiers should neither over-fit nor under-fit to be successful in the object classification.

There are two main approaches for object classification: supervised and unsupervised. If the label of the training samples is given priori, then this approach is called supervised, otherwise it is unsupervised. The approach, which is used in this thesis, is supervised.

Table 1 lists some of the major classification methods available in the literature, together with the principles behind them.

Table 1: Major Classification Methods. (Reference [79])

METHOD	PRINCIPLE
Nearest Mean Classifier	Finds the mean values of the training classes. Assigns the unknown object to the nearest class mean.
k-Nearest Neighbor Classifier	Finds the 'k' of the closest objects from the training set to the object being classified. Assigns unknown object to the majority class among k nearest objects.
Template Matching	Prototypes of the training classes from training samples are stored as templates. An unknown object is compared to all templates and matched to the nearest one.
Bayesian Classifier	Estimates the class conditional probabilities using the training samples and assigns unknown object to the class with maximum *posteriori* probability.
Linear Classifiers	Finds a hyper-plane that separates two classes. Assigns an unknown object by looking at which side of the hyper-planes it belongs.
Decision Trees	Constructs tree structures for classification. At each node, a comparison is performed. If the condition is satisfied, the right branch of the tree is selected; else the left branch is followed. At the leaf nodes of the tree, the unknown objects are classified.
Neural Networks	Weighted directed graphs, in which the nodes are artificial neurons and directed edges are connections between neurons. Each connection has an associated weight. Tries to find the optimal weights between nodes, to increase the classification accuracy. There are many architectural variations such as: multi-layer perceptrons, radial basis networks, Hopfield nets etc.

2.1.3 Combination of Classifiers

There are many different classification methods, such as k-means, k-NN, decision trees, Bayesian and Linear methods, neural network architectures, etc. as listed in Table 1. The observation, which states that the set of patterns misclassified by different classifiers does not overlap, leads the use of individual classifiers in a combined manner, rather than using a single one to increase the classification performance [45]. However, how to combine different classifiers is an open research area in pattern recognition and machine learning communities.

Majority Voting is the simplest method to combine individual classifiers. However, this approach gives equal weights to each classifier. Thus, a classifier having poor performance can affect the solution of the whole system. In order to overcome the weakness of this approach, weighted majority voting is proposed. On the other hand, finding the weights of each classifier is not an easy procedure in such an approach.

Another approach in the literature combines the classifiers by 'min', 'max', 'average', 'product' and 'sum' rules. In [33], the authors claim that using the 'sum' rule outperforms the other classifier combinations schemes. There are some approaches in the literature, which take the classifier outputs as fuzzy membership values and use fuzzy rules [46], [47], belief functions and Dempster-Shafer techniques [48], [49] to increase the classification performance. Some other techniques train the output classifier separately using the outputs of the input classifiers as new features [45], [50], [51].

Stacked generalization or stacking is a common approach that deals with the task of learning a meta-level classifier to combine the predictions of multiple base-level classifiers [45]. The success of stacking arises from its ability to exploit the diversity in the predictions of base-level classifiers and thus predicting with higher accuracy at meta-level. Alternative combination methods like boosting [2] and bagging [32] deal with multiple classifiers generated by applying the same learning algorithm to different versions of the data.

Another approach used in the literature is neural network ensemble techniques. Neural networks are inherently unstable, where small changes in training set and/or parameter selection may produce large changes in performance [52]. More stable and

smother predictions are generated by combining the predictions from ensemble of neural networks [46], [47].

2.2 A Literature Survey on CBIR Systems

The requirement of accessing to digital image and video resources on large image databases, including the World-Wide Web has made content based image retrieval an important research area. Instead of being manually annotated by text-based keywords, images would be indexed by their own visual content, such as color, texture, shape etc. In this section, the major studies of CBIR systems will be summarized. The major categories, namely, Image-based CBIR, object-based CBIR, relevance feedback and image annotation systems will be overviewed.

In a typical information retrieval system, a user poses a query by providing an existing image (or drawing one) and the system retrieves other "similar" images from the image database [18]. Most of the image retrieval (IR) systems adopt the following two-step approach to search image databases:

1) Indexing: For each image in a database, a feature vector capturing certain essential properties of the image is computed and stored in a feature-base.
2) Searching: Given a query image, its feature vector is computed to the feature vectors in the feature-base and images most similar to the query image are returned to the user.

While most IR systems retrieve images based on overall image comparison, users are typically interested in finding a specific object in the image. In this case, the user specifies an "interesting" sub-region (usually interesting object) of an image as a query. The system should, then, retrieve image containing this subregion or object from the database.

In traditional image retrieval systems, the user requests a query image and the system finds the most similar images to the query image from the image collection. In

such a method, each image in the image collection is compared to the query image with the same set of feature set. Unfortunately, this approach is not appropriate for automatic object identification in large image collections, since each object class has different set of representative features. Color can be an important feature to recognize "sky" object. However, it should be suppressed for recognition of "ball" object, where shape feature should be emphasized.

To overcome this difficulty relevance feedback approach is proposed, where the discriminative features of the query object are arranged according to the feedback of the user [53], [54], [55], [56], [57], [58], [59]. This method works well for querying, however, it can not be applied for image annotation of large image collections because of its dependency to the user.

To annotate objects in large image collections, three major approaches are proposed in the literature. The first one is clustering based methods, which are unsupervised [15]. These methods assume that similar objects are clustered in the feature space. Methods such as expectation maximization are used to assign cluster centres to the objects [60]. However, clustering based approaches can not differentiate the red apple from red ball, since, they will probably be in the same cluster. Second group of approaches is based on training, where a recognition engine is used to annotate images [61], [62]. This group is more consistent with the human visual system, which heavily relies on training. The third and the last of the approaches is semi-supervised methods. These methods are clustering based, but supervision is used to improve the cluster centers [63].

In this section a brief literature survey for the major content-based image retrieval systems will be provided. The available CBIR systems will be summarized under 4 headings:

1- Image-based CBIR Systems,
2- Region-based CBIR Systems,
3- Systems based on Relevance feedback,
4- Systems based on Image annotation.

2.2.1 Image-Based CBIR Systems

In image-based CBIR systems, each image is represented by some numerical measurements, which are called features. The features show the characteristics of images in terms of color, texture and shape. The images are compared to each other using the features, which are extracted from the whole image. The vast majority of current CBIR techniques can retrieve images only by similarity of appearance, using features such as colour, texture and/or shape or combination of some of them.

These techniques suffer from the 'semantic gap' between low-level visual features and high-level category concepts. This is one of the most crucial problems in content-based image retrieval systems. There is a mismatch between the capabilities of current CBIR systems and the needs of users. To improve the retrieval accuracy, a CBIR system must provide maximum support in bridging the 'semantic gap' between the low-level visual features and the richness of the user semantics.

The Query by Image Content (QBIC) is one of the first content-based image retrieval systems, which was developed by IBM research group [64]. The users can perform queries based on sample images, constructed sketches and drawings and selected color/texture patterns in QBIC system.

Photobook [65] provides a set of interactive tools for browsing and searching images and image sequences. These query tools make direct use of the image content rather than relying on text annotations. Photobook is demonstrated on databases, containing images of people, video keyframes, hand tools, fish, texture swatches, and 3-D medical data.

AMORE [66] is a Web search engine that allows the user to retrieve images from the Web by specifying relevant keywords or similar images. Text and image search can also be combined. A Query Result Visualization Environment is embedded into the system to show the query results to the user.

VisualSEEk [67] is a content based image retrieval system, which enables users to perform a wide variety of complex joint color/spatial queries. The user can form queries by diagramming spatial arrangements of color regions. The system utilizes an efficient indexing technique based on color information in retrieval.

WebSEEk [68] is a visual information system prototype for searching for images and videos on the World-Wide Web. Visual information on the Web is collected by automated agents and is catalogued and indexed for fast search and retrieval. The system is evaluated based upon the cataloging of over one half million images and videos from the Web.

2.2.2 Region-Based CBIR

In region-based CBIR systems, first, each image is segmented into regions. This process is performed either by using a segmentation algorithm or dividing images into sub-blocks and grouping them. These methods allow the user to locate a specific object in the image database.

NeTra [69] is an image retrieval system that uses color, texture, shape and spatial location information in segmented image regions to search and retrieve similar regions from the database. It incorporates an automated image segmentation algorithm that allows object or region-based search. Images are segmented into regions and image attributes that represent each of these regions are computed. The system includes an efficient indexing schema for fast search and retrieval. The user composes queries such as "retrieve all images that contain regions that have the color of object A, texture of object B, shape of object C, and lie in the upper of the image", where the individual objects could be regions belonging to different images.

In [70], Smith et. al. propose a method for image classification and querying using composite region templates. Their method classifies and queries images based on the spatial orderings of regions or objects using Composite Region Templates (CRTs). The CRT's capture the spatial information statistically and provide a method to measure similarity in the presence of region insertions, deletions, substitutions, replications, and relocations. The CRT's can be used for classifying and annotating images by assigning symbols to the regions or objects and by extracting symbol strings from spatial scans of the images. The symbol strings can be decoded using a library of annotated CRTs to automatically label and classify the images. The CRT's can also be used for searching by sketch or example by measuring image similarity based on relative counts of the CRT's.

Blobworld [16] is an image retrieval system based on finding coherent image regions, which roughly correspond to objects. The image is segmented into regions by fitting a mixture of Gaussians to the pixel distribution in a joint color-texture-position feature space. Each region ("blob") is then associated with color and texture descriptors. Querying is based on the user specifying attributes of one or two regions of interest, rather than a description of the entire image. In order to make large-scale retrieval feasible, they index the blob descriptions using a tree data structure.

SIMPLIcity (Semantics-Sensitive Integrated Matching for Picture Libraries) [17] is an image retrieval system, which classifies images into semantic categories. An image is represented by a set of regions, roughly corresponding to objects, which are characterized by color, texture, shape, and location. A measure for the overall similarity between images is developed using a region-matching scheme that integrates properties of all the regions in the images.

WALRUS [71] (Wavelet-based Retrieval of User-specified Scenes) partitions the image into regions. A wavelet-based feature vector is calculated for each image region. A dynamic programming algorithm is used to find the matching regions. The fraction of the area of the matching regions to the image size is used to calculate image similarity.

2.2.3 Relevance Feedback

Relevance feedback systems allow the user to retrieve images interactively. User selects the most relevant images and provides a weight of preference for each relevant image. The high level concept and perception subjectivity of the user can be captured by the system to some degree. The relevance feedback approach to image retrieval is a powerful technique and has been an active research area for the past few years. Various techniques have been proposed for this purpose.

In [53], Feng Jing et. al. propose a relevance feedback approach based on region representation. Their system can be considered as a special case of the query point movement method in region-based image retrieval. A composite image as the optimal query is formed by combining all the segmented regions of the positive examples. A

region-based image similarity measure is used to calculate the distance between the optimal query and an image in the database. An incremental clustering technique is also considered to improve the retrieval efficiency.

Ingemar et. al. propose a relevance feedback algorithm in [54], where a systematic evaluation is presented for searching a database. The algorithm takes feedback in the form of relative judgments ("item A is more relevant than item B") as opposed to the stronger assumption of categorical relevance judgments ("item A is relevant but item B is not"). The system, also, exploits a learned probabilistic model of human behavior to make a better use of the feedback.

In [55], Pengyu Hong et. al. propose an approach to utilize both positive and negative feedbacks for image retrieval. Support Vector Machines (SVM) is used for classification. The SVM learns to update the preference weights for the relevant images. The approach releases the user from manually providing preference weight for each positive example.

In [56], Ye Lu et. al. propose a relevance feedback technique to take advantage of the semantic contents of the images in addition to the low-level features. By forming a semantic network on top of the keyword association on the images. They are able to accurately deduce and utilize the semantic contents of images for retrieval purposes. Their method is tested on real-world image collections.

Meilhac and Nastar present a relevance feedback framework for content-based image retrieval, in [57]. Their model is based on non-parametric density estimation of relevant and non-relevant items. Their system is illustrated with several experiments and retrieval results on real-world data.

In [58], Kriengkrai Porkaew et. al. describe a generic query refinement framework, called Multimedia Analysis and Retrieval System (MARS) for learning query representations by relevance feedback from the users. The proposed framework expands the query by modifying the query representation, in which relevant objects are added to the query.

In [59], Y. Rui et. al. propose a relevance feedback model, where the user's high level query and perception subjectivity are captured by dynamically updated weights based on the user's feedback.

2.2.4 Image Annotation

Image annotation systems aim to translate the content of images to linguistic terms. They make the computers learn a large collection of concepts, build models about these concepts and recognize new image collections using the built-in models.

In [61], Y. Mori et. al. propose an automatic approach to annotating and retrieving images based on a training set of images. They assume that regions in an image can be described using a small vocabulary of blobs, which are generated from the image features by clustering. Given a training set of images with annotations, they show that probabilistic models allow predicting the probability of generating a word. This may be used to automatically annotate and retrieve images given a word as a query.

In [15], Duygulu et. al. consider object recognition as the translation of images to words. The "lexicon" for the translation is learned from large annotated image collections, which consist of images that are associated with text. First, the images are segmented into regions, which are represented by a pre-specified feature vector. Then, the regions are clustered in the feature space. The correspondence between the regions and words is learned by Expectation Maximization method.

In [72], Lavrenko et. al. propose an approach for learning the semantics of images, which allows to automatically annotate an image with keywords and to retrieve images based on the text queries. They assume that the image is divided into regions, each of which is described by a continuous-valued feature vector. Given a training set of images with annotations, they compute joint probability densities of image features and words, which allow the system to predict the probability of generating a word.

In [42], S. Feng et. al. present a system, based on a multiple Bernoulli relevance model, to automatically annotate images and videos. The model assumes that a training set of images or videos along with multiple keyword annotations is provided. However, the specific correspondence between a keyword and an image is not provided. Each image is partitioned into a set of rectangular regions and a real-valued feature vector is computed over the regions. The relevance model is taken as a joint-probability distribution of the word annotations and the image feature vectors, which is computed

using the training set. The word probabilities are estimated through a multiple Bernoulli model. The model is, then, used to annotate images in a test set. They perform experiments on both images from a standard Corel data set and a set of video key frames from NIST's Video Trec.

In [43], D. Blei and M. Jordan consider the problem of modeling annotated data with multiple types, where the instance of one type (such as a caption) serves as a description of the other type (such as an image). They describe hierarchical probabilistic mixture model to solve this problem. They conduct experiments on the Corel database.

In [44], the authors propose a computational model of the recognition of real world scenes, which bypass the segmentation and the processing of individual objects or regions. The procedure is based on a low dimensional representation of the scene, called the *Spatial Envelope*. They propose a set of perceptual dimensions (naturalness, openness, roughness, expansion, ruggedness) which represent the dominant spatial structure of a scene. Then, they show that these dimensions can be estimated using coarsely localized spectral information.

In [46], the authors study the statistical properties of natural images belonging to different categories and their relevance for scene and object categorization tasks. They discuss how second-order statistics are correlated with image categories, scene scale and objects. They show how simple image statistics can be used to predict the presence and absence of objects in the scene before exploring the image.

In [47], Yavlinsky et. al. describe a framework for automatically annotating images using non-parametric models of distributions of image features. They show that under the proposed framework, simple image properties such as global color and texture distributions provide a strong basis for reliably annotating images. They report results on subsets of two photographic libraries, the Corel Photo Archive and the Getty Image Archive. They, also, show that the popular Earth Mover's Distance measure can be effectively incorporated within this framework.

2.3 CBIR Systems with Linguistic Indexing and Machine Translation

Two of the well-known methods in the literature the ALIP system in [36] and Object Recognition by Machine Translation of [15] will be overviewed in this section. The performance of the proposed hierarchical schema is, then, compared to these popular systems throughout our experiments.

2.3.1 Automatic Linguistic Indexing of Picture Libraries by Statistical Modeling Approach (ALIP)

In [36], J. Li and J. Z. Wang introduce a statistical model to image annotation problem. Categorized images are used to train a dictionary of hundreds of statistical models each representing a concept. Images of any given concept are regarded as instances of a stochastic process that characterizes the concept. To measure the extent of association between an image and the textual description of a concept, the likelihood of the occurrence of the image, based on the characterizing stochastic process, is computed. A high likelihood indicates a strong association. In their implementation, they focus on a particular group of stochastic processes, that is, the two-dimensional multiresolution hidden Markov models (2D MHMMs). Their experiments have demonstrated the power of the system and its high potential in linguistic indexing of photographic images.

ALIP system has three major components:

1. The feature extraction process,
2. The multiresolution statistical modeling process,
3. The statistical linguistic indexing process.

In feature extraction, an image is partitioned into 4x4 pixel blocks. The system extracts a feature vector of six dimensions for each block. Three of these features are the average color components of pixels in the block. The other three are texture features representing energy in high frequency bands of wavelet transforms.

In multiresolution statistical modeling process, firstly, they define a series of concepts to be trained for inclusion in a dictionary of concepts. For each concept, they prepare a training set of images capturing the concept. Block based features are extracted from each training image at several resolutions. The same fixed feature set is used for each block of pixels. A cross-scale statistical model about a concept is built using training images belonging to a concept, each characterized by a collection of multiresolution features. Figure 1 shows the architecture of the statistical modeling process.

In statistical linguistic indexing process, the system automatically indexes images with linguistic terms based on statistical model comparison. Figure 2 shows the statistical linguistic indexing process of the system.

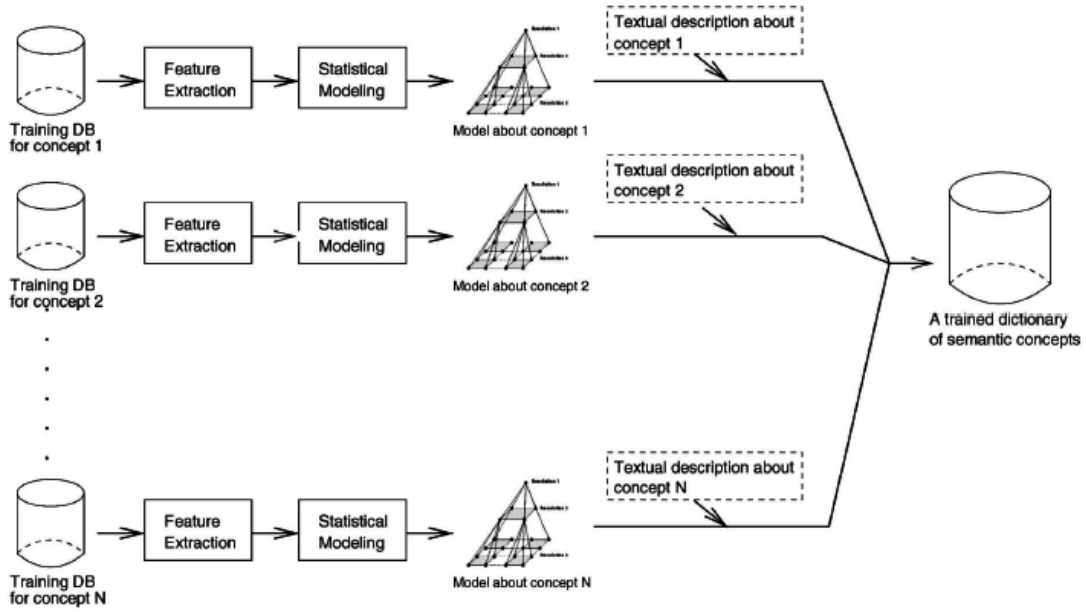


Figure 1: The Architecture of the Statistical Modeling Process. (Taken from Reference [36])

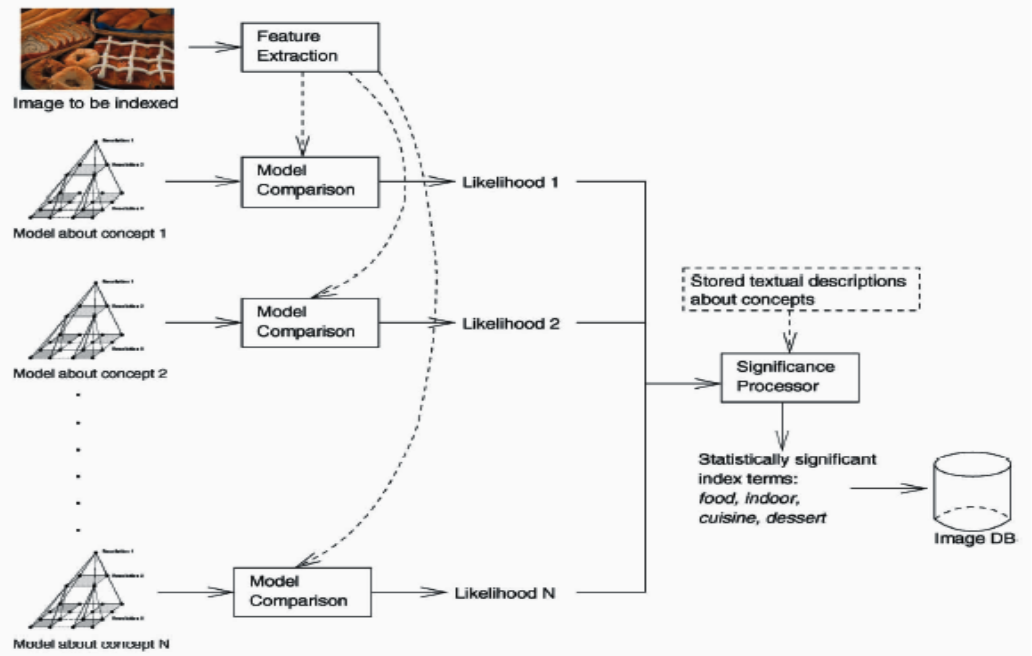


Figure 2: The Architecture of the Statistical Linguistic Indexing Process. (Taken from Reference [36])

2.3.2 Object Recognition by Machine Translation

In [15], Duygulu et. al. proposes an approach to the object recognition problem, which considers object recognition as the translation of image regions to words, like translation of text from one language to another one. The "lexicon" for the translation is learned from large annotated image collections that are associated with text. They, firstly, segment images into regions, which are represented by a feature vector. Then, they cluster feature vectors of regions to obtain a finite set of blobs. The correspondence between the blobs and the words are learned by Expectation Maximization method. After the learning stage, the correspondences are used to predict words corresponding to particular image regions or words associated with whole images. The features used in this study are:

- Area, x, y, boundary / area, convexity, moment of inertia (6),
- Average RGB (3),
- RGB stdev (3),
- Average L*a*b* (3),
- L*a*b* standard deviation (3),
- Mean oriented energy, 30 degree increments (12),

which form a 30 dimensional fixed feature set.

The system is purposely designed as unsupervised. However, for some cases, they showed that integration of supervisory input improves the performance. For this purpose, they used a small number of manually labeled images to use supervised data for better clustering and fix the correspondence errors. They gave the performance results of their study with this hand labeled data set, which is the one that we will also use in our experiments.

2.4 Training Schemas

The learning schemas of the proposed system are based on two major approaches, namely Adaptive Resonance Theory and Adaptive Boosting. In this section, these training schemes will be explained.

2.4.1 Adaptive Resonance Theory

The problem whereby the brain learns quickly and stably without catastrophically forgetting its past knowledge has been called the *stability–plasticity dilemma* [78]. This dilemma must be solved by every system that needs to respond its changing environment rapidly and adaptively. Adaptive Resonance Theory (ART), proposed by Stephan Grossberg in 1976, was designed specifically to solve *stability–plasticity dilemma* [34].

ART is a cognitive and neural theory, which is proposed to explain the challenging behavioral of brain in the area of visual perception. Within the brain, various spatially organized regions, or maps, exist that emerge dynamically. In these maps, neurons that respond to similar features of the sensory input are located near each other. The brain adapts itself in a self-organized manner according to the sensory input that is taken from the environment. This is the main logic behind the ART systems, where they try to learn an accumulating knowledge base in response to changing conditions as in the brain.

The brain's auditory system construct coherent representations of acoustic objects from the jumble of noise and harmonics that relentlessly bombards our ears throughout life [78]. In environments with multiple sound sources, the auditory system is capable of teasing apart the impinging jumbled signal into different mental objects, or streams. ART mechanisms of matching and resonance play a key role in achieving the selectivity and coherence that are characteristic of our auditory experience.

Saccades are eye movements by which an animal can scan a rapidly changing environment. The saccadic system in the brain plans where to move the eyes. How does the saccadic movement system select a target when visual and planned movement

commands differ? How does brain learn to interact with the environment during the selection process? ART matching and resonance are proposed to control the stability of this learning and the attentive selection of saccadic target locations [78].

ART systems contain a self-stabilizing memory that permits accumulating knowledge to be stably stored in response to arbitrarily events in a non-stationary environment under incremental learning conditions. Supervised ART systems can automatically adjust their scale of generalization to match the morphological variability of the data using a min-max learning rule that conjointly minimizes predictive error and *maximizes* generalization using the information that is locally available under incremental learning conditions [34].

ART systems have "many-to-one" mapping property, which enables them to learn different instances of a single class. Having such a property, "black bears" and "polar bears" having different characteristics can be mapped to a single "bear" class. Such a property is very useful in CBIR systems [1].

Human beings learn from both the prototypes and *exemplars*. Sometimes it is enough to learn abstract types of knowledge, such as being able to recognize that a particular object is a face or an animal. At other times, concrete types of knowledge are needed, such as being able to recognize a particular face or a particular animal. Supervised ART systems can learn both types of knowledge [1].

A confidence measure, called *vigilance*, calibrates how well an exemplar needs to match the prototype. The min-max learning rule is realized by *match tracking*, a process that raises the vigilance parameter in response to a predictive error just enough to initiate hypothesis testing to discover a better category [1].

All of the desirable properties of ART systems scale to arbitrarily large problems. On the other hand, ART helps to solve only learned categorization and prediction problems. These problems are, however, core problems in many intelligent systems [1]. ART systems have the ability to rapidly learn to classify large databases in a stable fashion and to focus attention up on feature groupings that are found to be important for each class.

ART properties have been used to explain and predict various cognitive and brain behaviours. There are different families of ART neural networks proposed in the

literature such as: ART-1 for unsupervised clustering of binary input patterns, ART-2 for unsupervised clustering of analog (continuous-valued) input patterns, ARTMAP [73] for supervised classification of input patterns and fuzzy ARTMAP [1] for generalization of ARTMAP using fuzzy set operations.

A central feature of all ART systems is a pattern matching process that compares an external input with the internal memory of an active code. ART matching leads either to a *resonant* state, which persists long enough to permit learning, or to a parallel memory search. If the search ends at an established code, the memory representation may either remain the same or incorporate new information from matched portions of the current input. If the search ends at a new code, the memory representation learns the current input. This *match-based learning* process is the foundation of ART code stability. Match-based learning allows memories to change only when input from the external world is close enough to internal expectations, or when something completely new occurs. This feature makes ART systems well suited to problems that require online learning of large and evolving databases [80].

Fuzzy ARTMAP is a promising architecture that has evolved from the biological theory of cognitive information processing. It is a supervised ART architecture, which can rapidly self-organize stable categorical mappings between m-dimensional input vectors and n-dimensional output vectors. It can perform incremental supervised learning of recognition categories and multidimensional maps in response to input vectors presented in arbitrary order. ARTMAP was initially proposed to classify input patterns represented as binary values. It is refined by Carpenter [1] by redefining ART dynamics in terms of fuzzy set theory operations. Fuzzy ARTMAP learns to classify inputs represented with a fuzzy set of features where each feature is a value in [0-1] scale indicating the extent to which that feature is present.

Fuzzy ARTMAP includes two ART modules ART_a and ART_b as shown in Figure 3. In ART_a , the input patterns are categorized and a match tracking mechanism maps these categories to the class templates coded at ART_b. Match tracking ensures maximum code compression at ART_a templates for minimum predictive error at ART_b templates.

To solve category proliferation problem observed in certain analog ART systems, Carpenter proposed to normalize the input vector for fuzzy ARTMAP architecture. Complement coding is proposed as a normalization rule that preserves amplitude information and represents both the presence and absence of a particular feature in the input pattern. In neurobiological terms, complement coding uses both on-cells and off-cells to represent an input pattern. The corresponding on-cell portion of a weight vector encodes features that are consistently present in category exemplars, while the offcell portion encodes features that are consistently absent. Small weights in complementary portions of a category representation encode as uninformative those features that are sometimes present and sometimes absent [80]. The complement coded input A is defined to be:

$$\mathbf{A} = (\mathrm{a}, \mathrm{a}^c) \equiv (a_1, \ldots, a_{M_a}, a_1^c, \ldots, a_{M_a}^c)$$

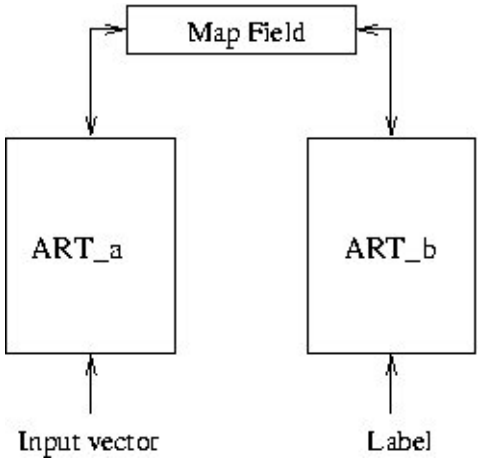


Figure 3: Fuzzy ARTMAP Architecture.

During the learning stage, ART_a receives a stream of input vectors where the entries consist of the membership values of the descriptors for each class. ART_b receives a stream of correct prediction of the input vector given to ART_a. When the activated categories in ART_a and ART_b are in resonance with their corresponding inputs, the map field pairs these activated recognition categories. Thus, the activated category of ART_a (in reponse to a) is mapped to the activated category of ART_b (in response to b). In addition to the map field, there is also a feedback control mechanism between ART_a and

ART_b. By this mechanism, a predictive mismatch in ART_b triggers a search for another recognition category in ART_a. This process is called match-tracking. Being an incremental clustering algorithm, Fuzzy ART works as follows:

Given a set S of input vectors, the following steps are performed for each fuzzy input vector I in S:

1. **Complement-code input I.** Let I = (a_1, a_2, …., a_M), then the complement-coded version of I is I_0 = (a_1, a_2, …., a_M; a^c_1, a^c_2, …, a^c_M) where a^c_i = 1-a_i. This step is necessary in order to avoid category proliferation problem.

2. **Choose a recognition category for the vector I_0.** Choose category J, which maximizes:

$$T_j(I_0) = |I_0 \wedge w_j| / | \text{alpha} + w_j | \qquad (1)$$

where the j_{th} category is represented by the weight vector w_j and ^ is the fuzzy-AND operator. Note that initially there is only one category with a weight vector consisting of all 1's. The expression in (1) determines the degree to which the weight vector w_j is a fuzzy subset of the input I_0. alpha > 0 is called the choice parameter. If more than one category is a fuzzy subset choice, the small but positive parameter breaks the tie by choosing J that maximizes $|w_j|$ among the fuzzy subset choices. Note that $T_j(I_0)$ is always in between [0,1] and can be interpreted as membership of category j.

3. **Resonance or reset.** Resonance occurs if the following match expression holds:

$$| I_0 \wedge w_j | / | I_0 | > Ro \qquad (2)$$

where Ro in [0; 1] is called the vigilance parameter. This parameter determines if the input pattern matches the expectancy for that category well enough or not. Learning can only occur when this vigilance criterion is met. Vigilance parameter defines the criterion

of an acceptable match between the input data and the active category. If (2) does not hold, then a mismatch reset is said to be occurred. A new category (excluding the one currently active) is selected according to (1). If no category resonates with I0, a new category $w_{new} = I_0$ is created.

4. **Learning.** When resonance occurs, learning is achieved by:

$$w^{new}j = Beta*(I \wedge w_{(old)j}) + (1 - Beta)\, w_{(old)j} \qquad (3)$$

where Beta is a parameter adjusting the learning speed. Setting Beta = 1 corresponds to fast learning. FAM is able to map more than one recognition categories in ART_a to the same recognition category in ART_b. Thus, classes whose patterns form distinct clusters in the feature space can be easily learned and recognized. This property of FAM is called many-to-one mapping.

The vigilance parameter calibrates how well an exemplar needs to match the prototype (recognition category). Since it can vary across learning trials, recognition categories are capable of encoding widely differing degrees of generalization or abstraction. With the help of this property a FAM system, for instance, can be trained to recognize a general human face and faces of specific individuals at the same time.

2.4.2 Adaptive Boosting

Building a single highly accurate prediction rule is a difficult task. On the other hand, it is not hard to come up with very rough rules of thumb that are only moderately accurate. An example to explain this fact is given in [74] about predicting spam e-mails. It is very difficult to predict all types of spam e-mails from a single accurate rule. On the other hand, many rules such as: "If the phrase 'buy now' occurs in the e-mail subject, then predict it is spam." can be combined to predict spam e-mails. Although this rule says nothing about what to predict if 'buy now' does not occur in the e-mail subject, it will make predictions that are significantly better than random guessing.

Boosting is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. Boosting algorithm

combines these weak rules into a single prediction rule that will be much more accurate than any one of the weak rules.

Boosting is a general method for improving the performance of any learning algorithm. Boosting can be used to significantly reduce the error of any "weak" learning algorithm that consistently generates classifiers, which are better than random guessing. Boosting works by repeatedly running a weak learning algorithm on various distributions over the training data. Then, the classifiers produced by the weak learner are combined into a single composite classifier.

One of the first effective boosting algorithms was proposed in [75]. The author introduced a method, which improves the accuracy of algorithms for learning binary concepts. He achieved an improvement by combining a large number of hypotheses, each of which is generated by training the given learning algorithm on a different set of examples. In his study, examples that are given to the learning algorithm are generated by, choosing the instances at random from distribution over the instance space.

AdaBoost [2] is a well-known algorithm to obtain a strong classifier from a set of weak classifiers. Each weak classifier performs slightly better than the random guessing. The input to the algorithm is a set of features and their labels $(x_1,y_1),\ldots (x_m,y_m)$ extracted from a training set. The main idea of the algorithm is to maintain a set of weights over the training set. The strong classifier, is then, defined as a linear combination of the weak classifiers using the weights. Initially all weights are equally likely, but on each round, the weights of incorrectly classified examples are increased so that the weak learner is forced to focus on the hard examples in the training set. The final hypothesis H is a weighted majority vote of the T weak hypothesis where $\underline{\alpha_t}$ is the weight assigned to h_t. The weight on training example i on round t is denoted as $D_t(i)$.

The pseudo-code of the AdaBoost algorithm is given in Table 2. "X" denotes the feature vector, where "Y" denotes the class label of it. "Y" is selected as 1 for the examples in the same class and -1 for the ones outside the class. D(i) denotes the weight vector for each training example. Initially, all examples have the same weight. At each iteration, where "T" denotes the iteration number, the weak hypothesis giving the smallest error is chosen. The weight vector is updated so that the misclassified examples are given a higher weight. This enables the algorithm to give more importance to these

examples in the next iterations. The final hypothesis is the majority weighted decisions of the weak hypothesis. If each weak hypothesis performs better than random guessing, the majority weighted decisions of all of them becomes a strong hypothesis.

Table 2: Basic AdaBoost Algorithm. (Reference [2])

Given: $(x_1, y_1), \ldots, (x_m, y_m)$ where $x_i \in X$, $y_i \in Y = \{-1, +1\}$
Initialize $D_1(i) = 1/m$.
For $t = 1, \ldots, T$:

- Train weak learner using distribution D_t.
- Get weak hypothesis $h_t : X \to \{-1, +1\}$ with error

$$\epsilon_t = \Pr_{i \sim D_t}\left[h_t(x_i) \neq y_i\right].$$

- Choose $\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$.
- Update:

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha_t} & \text{if } h_t(x_i) = y_i \\ e^{\alpha_t} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t} \end{aligned}$$

where Z_t is a normalization factor (chosen so that D_{t+1} will be a distribution).

Output the final hypothesis:

$$H(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right).$$

2.5 Chapter Summary

In this chapter, we discussed the fundamental concepts of object localization and content based image retrieval systems. We also summarized two of the well-known methods in the literature, namely, ALIP system in [36] and Object Recognition by Machine Translation of [15]. At the end of the Chapter, training schemas used in the training phase of the proposed system are overviewed.

CHAPTER

3 GENERAL FRAMEWORK FOR OBJECT LOCALIZATION AND IMAGE RETRIEVAL

In this study, we develop an object localization and image retrieval framework for Content Based Image Retrieval Systems. The user can make object-based queries such as: "find me images, which contain leopards." The system is expected to return the images containing the queried object as illustrated in Figure 4.

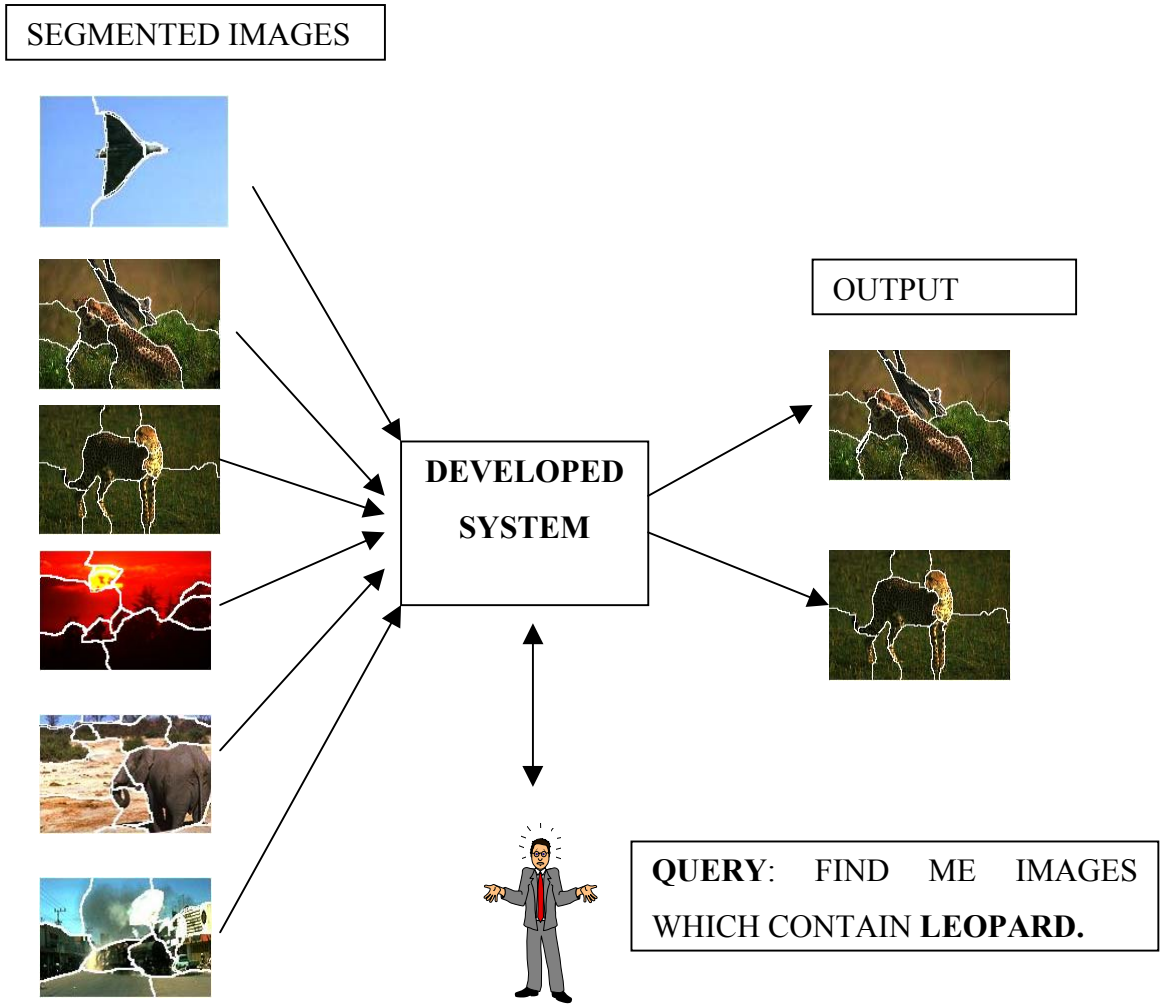


Figure 4: The Diagram Showing the Aim of This Thesis.

The proposed image retrieval framework is based on supervised learning. The user can only query the objects, which are trained before, as in the human visual system. We believe that supervision is essential for performances required in most of the practical applications. This fact is also validated in the experiments.

The general framework of the proposed system can be explained under three major stages: (See Figure 5)

1- Training,
2- Object Labeling,
3- Object Querying.

In the training stage, firstly, the images in the database are segmented. Then, features of the segmented regions are extracted. Finally, these features are fed to the training module of the proposed system. This module trains a recognition engine to label an unknown region or a group of neighboring regions corresponding to an object.

The second stage of the system is object labeling. The input to this stage is a set of unlabeled images in the test set. First, the images are fed to a segmentation module, which yield mostly more than one region for an object. Then, the neighboring regions are merged in the labeling module by using a tree data structure to localize the objects. Finally, as an output, the labeled objects are stored in a database.

The last stage of the developed system is querying, which enables the user to perform object-based image retrieval. The query results are shown to the user depending on the membership values, which are obtained in the labeling module. Note that in the proposed schema, only the objects learned during the training stage can be queried.

In this chapter, we shall bring a bird's eye view to the general structure of the stages mentioned above. Later, in Chapter 4, we shall explain the detailed structure of the individual modules, emphasizing the contributions, introduced in this study.

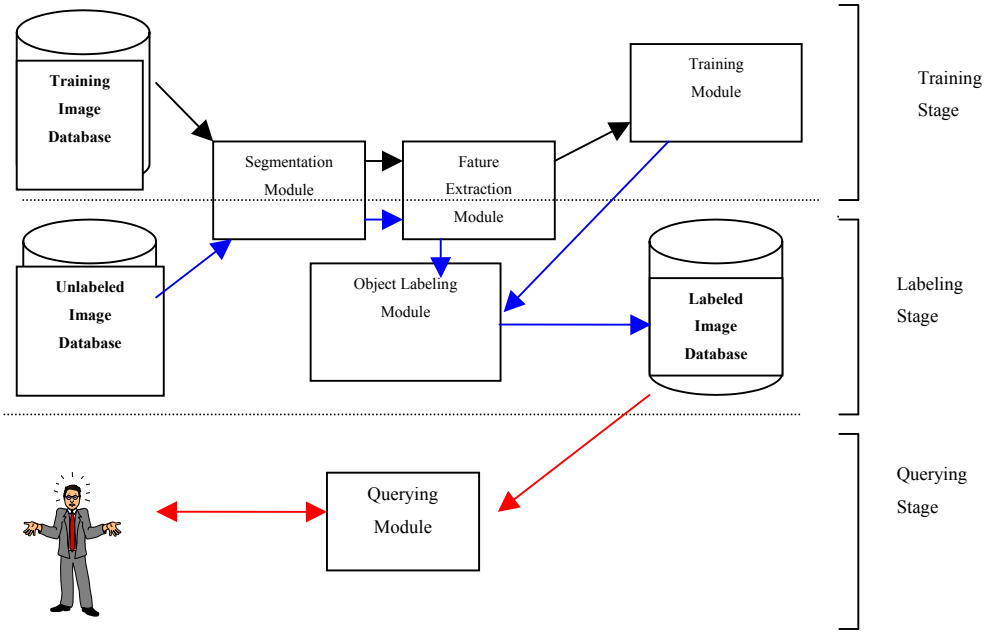


Figure 5: Block Diagram of OLIRF.

3.1 Training Stage

The major goal of this stage is to train a recognition engine, which can localize the objects in the querying stage. The input to the training stage is a set of object images, which represent samples from each class. The user selects the objects appearing in the images by using a bounding rectangle. Training stage consists of three major modules, namely segmentation, feature extraction and training. In the following, we shall briefly explain the methodologies to develop these modules.

3.1.1 Segmentation Module

Segmentation module groups "similar" pixels according to a metric to form "homogeneous" regions. The images in the database are segmented into regions using the N-cut segmentation algorithm of reference [76]. N-Cut treats image segmentation as a graph partitioning problem. A criterion, called the normalized cut, for segmenting the

graph is proposed, where the pixels of the image constitute the nodes and the similarities between each pixel pair constitute the edges of the graph. The normalized cut criterion measures, both the total dissimilarity between the different groups as well as the total similarity within the group.

Given an undirected graph, $G = (V, E)$, where V is a set of nodes representing pixels of the image and E is a set of edges between connected pixels. The weight on each edge, $w(i,j)$ is a function of the similarity between nodes i and j. N-Cut algorithm seeks to partition the vertices into disjoint sets $V_1, V_2, \ldots, V_m$, where the similarity among the vertices in set V_i is high and across different sets V_i, V_j is low, according to normalized cut criterion.

The images in the database are stored as a set of regions obtained from the segmentation module for further processing steps. Segmentation algorithms either perform under-segmentation or over-segmentation as illustrated in Figure 6. Since, more than one object is likely to be represented with a single region in an under-segmented image; it is almost impossible to localize the object without missing some parts. Therefore, we prefer N-cut algorithm to perform over-segmentation, which mostly yields parts and rarely yields the whole objects.

a) b)

Figure 6: Examples of Under-segmetation (a) and Over-segmentation (b)

Obviously, the regions obtained from the segmentation module can not be directly used for object labeling. A merging algorithm is required to obtain the whole object from its parts. This algorithm will be presented in Chapter 4.

3.1.2 Feature Extraction Module

This module extracts a set of feature vectors, each of which correspond to a certain visual description, for each region. The feature vectors, called descriptors, are representative of low level visual information. For this purpose, initially, a large variety of descriptors available in the literature, are selected by a preliminary inspection. In this study, Dominant Color (4 features), Color Structure (32 features), Scalable Color (16 features), Edge Direction Histogram (80 features), Region-based Shape (35 features), Gabor (48) and Haar (192) are chosen from references [6], [7] and [77]. The number and type of the descriptors is to be determined in an initial analysis step, depending on the application domain.

Dominant Color Descriptor characterizes an image or region by a small number of representative colors. The descriptor consists of the RGB coordinates of the representative colors and their frequencies. This descriptor provides a very compact representation of the color distribution in the image [7].

Color Structure Descriptor captures both color content and information about the spatial arrangement of the colors. Specifically, it is a histogram that counts the number of times a color is present in an 8x8 windowed neighborhood, as this window progresses over the image rows and columns. This enables us to distinguish images with the same color and different color histogram in a neighborhood, taking into account of the spatial relations [7].

Scalable Color Descriptor measures the color distribution of an image in HSV Color Space. The feature extraction consists of a histogram extraction in HSV color space, uniformly quantized into 256 bins. To reduce the large size of this representation, the histograms are encoded using a Haar transform [7].

Edge Histogram Descriptor represents the spatial distribution of five types of edges (Figure 7), namely four directional edges (horizontal, vertical, left and right diagonal edges) and a non-directional edge for 16 local regions in the image. Since there are five types of edges for each local region, we have total 16*5=80 histogram bins. The filters, used to calculate the edges in an image, are shown in Figure 7 [7].

1	-1
1	-1

1	1
-1	-1

$\sqrt{2}$	0
0	$-\sqrt{2}$

0	$\sqrt{2}$
$-\sqrt{2}$	0

2	-2
-2	2

a) ver_edge_filter() b) hor_edge_filter() c) dia45_edge_filter() d) dia135_edge_filter() e) nond_edge_filter()

Figure 7: Filters Used for Edge Histogram.

Region-Based Shape Descriptor utilizes a set of Angular Radial Transform coefficients, which is a 2D complex transform defined on a unit disk in polar coordinates. This descriptor uses twelve angular and three radial functions to extract 35 features, which are placed directly into the feature vector. Shape of an object may consist of either a single region or a set of regions as well as some holes in the object, as illustrated in Figure 8 (c). Since the region-based shape descriptor makes use of all pixels constituting the shape, it is capable of describing generic shapes, as illustrated in Figure 8 (d) and (e).

Figure 8: Examples of Various Shapes.

Gabor Filter Descriptor, reported by Manjunath and Ma [6] use a dictionary, which contains four scales and six orientations. Each filter captures the relevant texture primitives of the image. Second order statistics of the Gabor Filter (4 scales × 6 orientation = 24 filter) responses of a given texture is used as a texture descriptor. In this way, Gabor features of individual images are extracted separately by finding their mean oriented energies and energy variances. Thus, each image is represented by a vector of size 48. Gabor Filters are commonly used in texture analysis problems. The

representation of images by Gabor Filters has been shown to be optimal in the sense that they minimize the joint two-dimensional uncertainty in space and frequency domain. The filters can be considered as orientation and scale tunable edge and line detectors.

Wavelets are a type of multi-resolution function approximation that allows the hierarchical decomposition of an image. Wavelets encode information about an image from the coarse to fine details, when applied at different scales. The Haar functions are the simplest wavelet basis and provide a mathematically sound extension to an image invariance scheme. Haar wavelets of different scales are used to generate a multi-scale representation of the images. At each scale, three different orientations of Haar wavelets are used, each of which responds to differences in intensities across different axes. In this manner, information about how intensity varies in each color channel in the horizontal, vertical and diagonal directions is obtained [82]. Initially, each image is scaled into a 128 x 128 pixel. Then, 4-layer 2 dimensional Haar wavelet transform is applied. Then the upper left 8 x 8 corner of each transform matrix represents the lowest frequency band of the 2-D image in a particular color component for the level of wavelet transform. Concatenation of vectors obtained by Haar wavelets at different scale forms the **Haar Descriptor**.

The descriptors, obtained above, can be used in many ways to design a feature space. The most straightforward way is to concatenate the features of all descriptors in the same vector and apply normalization to bring them to the same dynamical range. We can, also, find feature subspaces for object classes for better retrieval or classification accuracy. There are many ways to design such sub-spaces in the literature [19]. We propose a robust technique for this purpose, which will be explained in Chapter 4.

3.1.3 Training Module

The goal of the training module is to improve the performance of the querying stage by supervised learning. Training is performed by combining k-NN [79], AdaBoost [2] and fuzzy ARTMAP [1] algorithms in various architectures. The details of the proposed architectures will be explained, later, in Chapter: 4.

There are two major components of training module. The first component trains the whole objects, which are selected and labeled by the user in a bounding rectangle. The second module automatically grabs and labels the regions in the bounding rectangle, which are obtained from the output of the segmentation algorithm (Figure 9). Note that, the images in the training set consist of regions, which are labeled if they belong to the objects to be queried. The rest of the regions are left unlabeled.

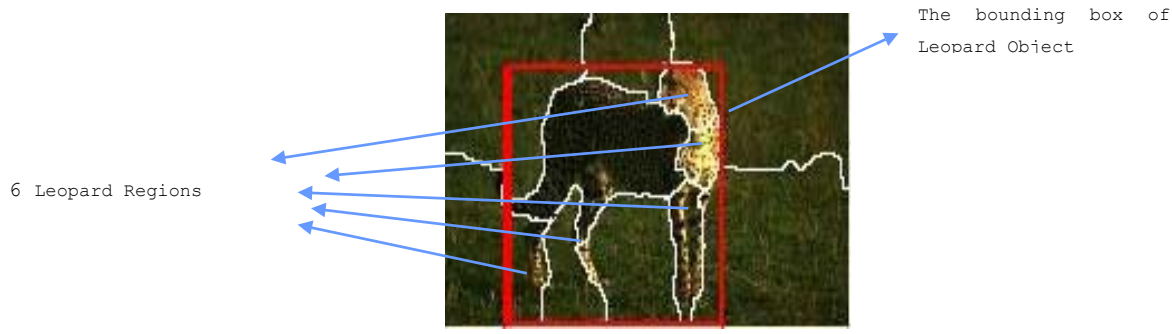


Figure 9: A bounding Box of a Training Object and Its Segments.

The training component used for the whole object, receives a feature vector, which is formed by color, texture and shape features, mentioned in the previous section. The training component for the regions excludes the region based shape feature, since the shape is not a characteristic feature of the sub-regions of the objects.

At the output of the training module, membership values for each query region are obtained. Mathematically speaking, let a given image regions, R_i, for $i = 1, \ldots, I$, belong to the same query object, where I indicates the number of regions in the whole object. Let C_j for $j = 1, \ldots, J$ represent J distinct object classes to be queried. Then, the membership vector for region i is defined as:

$M_i = [M_{ij}]$,

where the entries of the vector, M_{ij} represents the membership value of j^{th} category for region *i*.

The membership values M_{ij} are obtained from the recognition engines. As an example, if k-NN is used as the recognition engine, then M_{ij} is simply equal to the k_j / k, where k_j is the number of images belonging to class *j* among *k* images. If fuzzy ARTMAP is employed as the recognition engine, $|I_0 \wedge w_j| / |w_j|$ is used as the membership value, where I_0 is the input vector, w_j is the weight vector of the j_{th} category and ^ is the fuzzy-AND operator.

3.2 Labeling Stage

The labeling stage receives the images as the input to the segmentation module. The features for all the regions are extracted. Then, the labeling module is used to label an unknown region or a group of neighboring regions corresponding to an object. Figure 10-a and Figure 10-b indicate the labeling process for an unknown region and a group of regions corresponding to an object by the training module.

In order to label an object in the unlabeled image database, one should find an algorithm, which extracts the whole object by combining adjacent segments. For this purpose, we propose a labeling algorithm, based on "Neighborhood Tree" concept, whose details will be discussed in Chapter: 4.

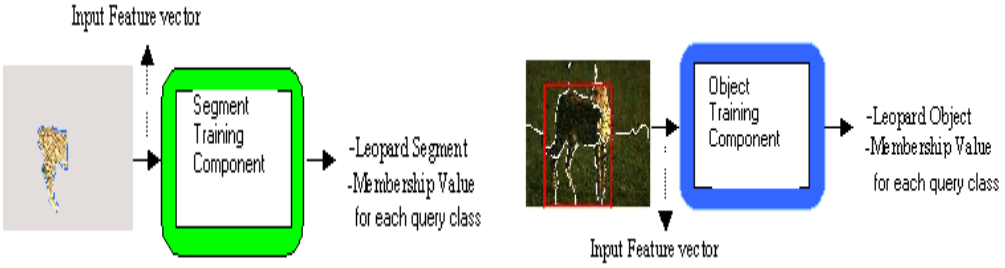


Figure 10: The Input and Output of Training Components a) for Object Parts, b) for Whole Objects.

3.3 Querying Stage

The output of the labeling algorithm introduced in the previous section provides us a set of labeled regions and corresponding membership values. In the querying module, the user selects a query object and the labeled objects in the database are shown to the user from highest to lowest membership values. Figure 11 shows the result of a sample query with the corresponding membership values for the query object "plane".

Figure 11: First 5 Query Results and Corresponding Membership Values.

3.4 Chapter Summary

In this chapter, we present the general framework of the proposed CBIR system and the rationales behind it. We introduce a bird's eye view to the proposed system, explaining the three major stages, namely, training, labeling and querying. In Chapter 4, we present the detailed structure of the individual modules, emphasizing the contributions introduced in this study.

CHAPTER

4 DESIGN OF THE FEATURE SPACE, TRAINING AND LABELING SCHEMAS

This chapter emphasizes the contributions in the proposed Content Based Image Retrieval System. Two important methods are proposed to improve the performance of the current CBIR systems: Firstly, a dedicated feature space is designed for the representation of each object class. Secondly, a method for object localization problem is proposed by introducing the concept of Neighborhood Tree, where the over-segmented images are represented in a tree data structure. Then, the object localization in an image database is reduced to a tree search problem.

The next sections present details of the approaches, mentioned above.

4.1 The Concept of "Best Representative Feature Space"

Consider, a 2-class classification problem, with n and m images in each class, respectively. We have total of 'n+m' images. Assume that n is close to m. Let the images in class-1 be very similar to each other according to color features, but not similar according to shape features; whereas the images in class-2 be very similar to each other according to shape and dissimilar according to color features. In an image retrieval system, if only color feature were used for comparison, a performance of nearly 50% is obtained. Color features give satisfactory results for the objects in class-1 and poor results for the objects in class-2. In order to improve the performance, the system should query images in class-1 with color features and images in class-2 with shape features. This is the main motivation behind the proposed content based image retrieval system, where the discriminative feature set is identified for each object class by using a

preliminary training stage. As indicated in Figure 12, the best representative features for the `bus` is shape, whereas `sky` class can be retrieved by mixing the color and texture features. Therefore, a method, which determines a dedicated feature set depending on the characteristics of each object class, is needed to improve the query performance.

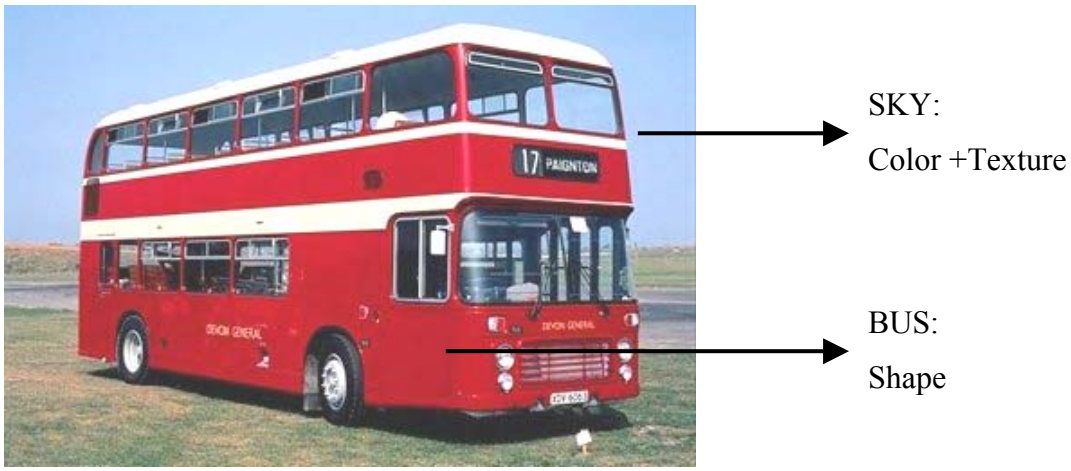


Figure 12: The Diagram Explaining Best Representative Feature Set.

Let us start by clarifying some basic definitions, frequently used in this study. A descriptor is defined as a vector, formed by low level and semantically consistent feature measurements. In other words, descriptors are the feature vectors with the entries each of which represent the same visual phenomena for a given object or region. For example, HIS color vector is a typical descriptor of dimension three. The space spanned by the descriptor vector is called descriptor space. The feature vector is formed by concatenating a large variety of descriptors. The feature space is, then, the vector space spanned by features of all descriptors. Note that the descriptor spaces correspond to the subspaces of the feature space.

The methods, which are introduced in this Chapter, extract a dedicated feature space for each object class by weighting or selecting some of the descriptors in the feature vector. The design process involves the maximization of the retrieval performance on a training data set. For this reason, the resulting feature space is called "best representative feature space".

In this thesis, five methods are employed to design "the best representative feature space" for each object class. Two of the approaches are based on the fuzzy ARTMAP algorithm, given in Section 2.4.1 and AdaBoost algorithm of Section 2.4.2. The other three methods, proposed in this thesis, are:

1. Best Representative Descriptor (BRD) Using Euclidean Distance,
2. Descriptor-Based AdaBoost and
3. Hierarchical Learning Schema based on Adaptive Resonance Theory.

In the following sections, details of the above methods for identifying the best representative feature space together with the training schema will be given:

4.1.1 Best Representative Descriptor Using Euclidean Distance

The major goal of this approach is to select the "best" descriptor for each object class, which maximizes the retrieval performance on the test data. For this purpose, the images in each class are queried by all descriptors using the Euclidean distance. The descriptor, which maximizes the precision value for a class, is identified as the *best representative descriptor* of that class.

Let $D = \{\underline{d}_1, \underline{d}_2, \ldots, \underline{d}_m\}$ be the set of descriptors, where each element, $\underline{d}_i$ representing a descriptor vector. Then, the following algorithm, summarized in Table 3, finds the best representative descriptor $\underline{d}_i$ among m descriptors in set D, for each of the training classes:

Table 3: Construction of Best Representative Feature Space with Euclidean Distance.

Algorithm: Best Representative Feature Space with Euclidean Distance

For each of the training classes, repeat
  For each descriptor $\underline{d}_i$ in D, repeat
    Retrieve the most similar objects for each of the objects in the training class according to descriptor $\underline{d}_i$ using Euclidean distance. Calculate the precision of descriptor $\underline{d}_i$.
  End For
  Output the best representative descriptor $\underline{d}_i$ of the training class as the one, which maximized the precision.
End For

The algorithm of Table 3 finds only one descriptor as the "best descriptor" in terms of retrieval precision. However, it is trivial to extend the BRD to k-best representative descriptors by including a large set of descriptors and finding the best k-representative descriptors among them.

4.1.2 Desciptor-Based AdaBoost

In order to extend the BRD method to k-best representative descriptors, we propose an algorithm, called Descriptor-Based AdaBoost. In this algorithm, each descriptor is selected as the weak classifiers of the AdaBoost algorithm, given in Section 2.4.2. At each iteration, the descriptor, which decreases the classification error, is selected. By increasing the weights of misclassified examples, the algorithm is forced to find the descriptors, which can classify the misclassified examples.

Let, $D = \{\underline{d}_1, \underline{d}_2, \ldots, \underline{d}_m\}$ be a set of the descriptor vectors, $\underline{d}_i$. Let, t be the number of descriptors to be selected, out of m total descriptors. Note that $t<m$. Let, $S = \{\underline{d}_1, \underline{d}_2, \ldots, \underline{d}_t\}$ be the set of selected descriptors. Then, the algorithm, summarized in Table 4 finds the t-best representative descriptors in D, for each of the training classes.

Table 4: Construction of Best Representative Feature Space with Descriptor Based AdaBoost Method.

Algorithm: Best Representative Feature Space with Descriptor Based AdaBoost Method

For each of the training class, repeat

Initialize the weights of the images in the training class. (Using the formulation in the original AdaBoost Algorithm given in Section 2.4.2)

For j = 1..t repeat

For i = 1 .. *m* repeat

Select the descriptor $\underline{d}_i$ in D, which minimizes the classification error.

End For

Set $S_j = \underline{d}_i$

Update the weights of the images in the training class such that the missclassified images are focused at the next iterations. (Using the formulation given in the original AdaBoost Algorithm)

End For

Output the selected descriptors $\{S_1, \ldots, S_t\}$ as the best representative descriptors of the training class.

End For

The algorithm of Table 4, selects *t*-best representative descriptors from the feature space of all the descriptors. The logic behind this algorithm is to boost the performances of *t* descriptors. This algorithm improves the retrieval performance by selecting descriptors, each of which learns to discriminate different group of images in the database.

4.1.3 Hierarchical Learning Schema based on Adaptive Resonance Theory

The most straightforward way of identification of the Best Representative Feature Space, using Fuzzy ARTMAP is to feed the normalized feature vector obtained by concatenating all the descriptors directly to the Fuzzy ARTMAP. This approach suppresses the weights of the irrelevant or redundant features for a particular class, automatically. However, feeding the numerical values of the feature vector to the fuzzy architecture bears some problems. First of all, the input vector is obtained by concatenating all features of different types, dynamical ranges and semantics. This type of feature vector requires normalization, which has some side effects, as mentioned in Introduction Chapter. There is no guarantee for the vector space, spanned by the normalized feature vectors, to bear the same visual information as the initial vector space.

Secondly, although the normalization process maps the numerical entries of the feature vector into [0,1] interval, these values do not correspond to the memberships. On the other hand, Fuzzy ARTMAP relies on the fuzzy-AND operation as its similarity metric. It works only when the entries of the input vector indicate the membership of a feature for discriminating a particular object. Therefore, fuzzy ARTMAP requires the membership values of the feature vector, rather than the measurement of the features.

Finally, existing utilization of Fuzzy ARTMAP architecture also bears the problem of semantic gap and curse of dimensionality. Concatenation of all features of different characteristics in the same vector violates the semantic meaning of each descriptor. And also, each added descriptor increases the dimension of the input feature vector, causing the curse of dimensionality problem.

In order to solve the problems mentioned above, we propose a two-layer stacked generalization method, which emulates the eye-brain channel. The base-layer of the proposed system extracts and learns a set of low level and low dimensional descriptors, which corresponds to the low level visual processes in the eye. The meta-layer emulates the associative memory by implementing the Adaptive Resonance Theory. At this point, we believe that the Fuzzy ARTMAP architecture at the meta-layer utilizes the high level information generated by the base-layer classifiers and decreases the semantic gap.

The proposed two-layer hierarchical architecture processes the low-level features in the base-layer to generate the high-level features. Then, these high-level features are fed to meta-layer, as explained below:

Base-Layer: k-Nearest Neighbor Classifiers:

In this study, we choose k-NN as the base-layer classifier for the sake of simplicity. K-NN is known to be a robust and simple classifier. It assigns an unknown object to the majority class among the *k* nearest neighboring objects. Note that any other classifier may also be selected at the base-layer.

For this purpose, first, a set of low-level color, texture and shape descriptors, $\underline{d}_i$, for $i = 1,...,m$, where m indicates the total number of descriptors, are extracted to represent the visually meaningful information. In the base-layer, each descriptor, $\underline{d}_i$, is fed to a k-Nearest Neighbor $(\text{k-NN})_i$ classifier. Since each $(\text{k-NN})_i$ receives a single semantically consistent descriptor, $\underline{d}_i$, the curse of dimensionality and normalization problem is avoided.

Now, the major problem is how to combine the m k-NN classifiers in the meta-layer to generate high-level features. The classification rate, C_{ij}, of $(\text{k-NN})_i$ classifier for each class j = 1, ..., J indicates the importance of $\underline{d}_i$ to recognize class j. Therefore, C_{ij} of each $(\text{k-NN})_i$ classifier for class j can be interpreted as the membership value of a descriptor to contribute to correctly classify class j.

Meta-Layer: Fuzzy ARTMAP

As mentioned in section 2.4.1, during the learning phase, ART_a receives a stream of input vectors *I,* whereas ART_b receives the stream of correct prediction given *I*. In the proposed architecture, the membership values C_{ij} of the base-layer classifiers, are received as input vectors *I of* ART_a.

Let us now explain the computation of membership matrix, which is the input of the meta layer: $C = [\underline{C_i}]$ corresponds to the membership matrix whose entires $\underline{C_i} = [C_{ij}]$ indicating the membership vectors for $i = 1..m$, where *m* indicates the total number of descriptors. $\underline{C_i}$ consists of *the entries* C_{ij} denoting the membership value of i^{th} descriptor created by k-NN classifier for class j. Suppose that there are total of *J* classes. Let, the number of occurrence of each class among the nearest k-neighbors of an input feature vector using i^{th} descriptor be $[k_1, \ldots, k_J]$. Note that $k_1 + \ldots + k_J = k$. Then, the corresponding membership vector of descriptor i, for the input feature vector is equal to $\underline{C_i} = [k_1/k , k_2/k, \ldots, k_J/k]$. It should also be noted that the sum of the entries of the membership vector $\underline{C_i}$ is equal to 1, as expected.

In the meta-layer, fuzzy ARTMAP receives the membership values, C_{ij}, of each descriptor for each class as high-level feature vectors. This does not only eliminates the normalization process required for the fuzzy ARTMAP, and also enables fuzzy ARTMAP architecture to work with membership values. Figure 13 indicates the proposed hierarchical architecture.

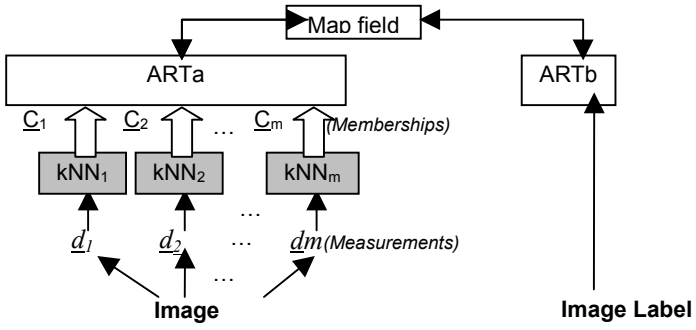


Figure 13: Proposed Hierarchical Architecture.

The proposed two layered architecture, using fuzzy ARTMAP as the meta-layer, has some key properties, which makes it a very suitable system for CBIR. First of all, it achieves a many-to-one mapping, which enables the system to map the samples having different characteristics to the same class. This property is an important feature of fuzzy ARTMAP architecture, as explained in Background Chapter. Having this property, the objects can be grouped under the same class, even if their feature vectors are dissimilar. Such a property cannot be achieved by classical distance-based similarity measuring CBIR systems.

Secondly, the proposed two-layer system can be regarded as a salient feature detector for each class. The base-layer obtains membership values of each descriptor for each class by using k-NN classifiers. The upper layer learns the degree of importance of each descriptor for each class using the membership values of base-layer. The system is capable of assigning larger weights to the color and texture features for retrieving the "sky" class and to the shape feature for the "bus" class. In other words, the best representative features space for each object class is obtained by weighting each descriptor.

Thirdly, the proposed architecture can be a solution to the stability-plasticity dilemma by using FAM architecture at the meta-layer. With the match-based learning property of ART systems, they change their memories only when the inputs are close enough to their expectations, or when something completely new occurs. This property enables stable learning of large and evolving databases as desirable for CBIR systems.

Lastly, the proposed system achieves fine or broad categorization to map the input vectors to the classes by chancing the value of vigilance parameter. High vigilance values result in finer categorization, whereas with low vigilance values, a broader categorization is achieved. In this way, various levels of abstraction are achieved.

4.2 Localization of an Object In a Segmented Image

Segmentation is usually the first step in object localization. It is very difficult to perform an exact segmentation, which partitions an image into regions corresponding to whole objects. Segmentation algorithms either under-segment an image having more than one object in a homogenous region or over-segment an object by splitting it into several regions. This is, basically, because of the fact that most of the available methods in the literature do not incorporate expert knowledge about the image content. As a result, the segmented regions may or may not correspond to meaningful objects. Since splitting an under-segmented region is much more difficult than merging over-segmented ones, we prefer over-segmentation in object localization. Figure 14 shows some sample images, which are over-segmented with N-Cut algorithm [76].

Figure 14: Sample Segmented Images.

In order to retrieve objects in over-segmented image database, one should find an algorithm, which extracts objects by combining the adjacent segments, as indicated in Figure 15.

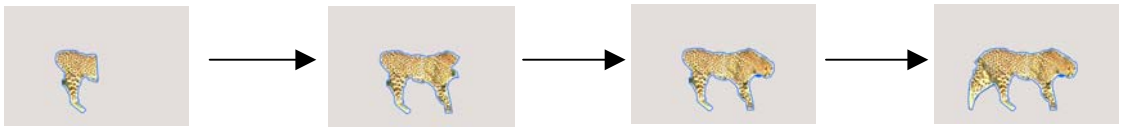


Figure 15: Concatenation of Regions to Form Leopard Object.

In this thesis, we propose two object localization approaches:

1. Greedy-Based Approach
2. Neighborhood Tree Approach

Details of the above approaches are discussed in the following subsections. Table 5 shows the notation used in the pseudo-codes of the algorithms.

Table 5: Notation.

I : Total number of segmented regions in an image.
J : Total number of query classes.
R_i : Segmented region i.
M_{ij} : Membership value of the Segmented region *i* for the class *j*.
L_i : Label of the Segmented region *i*. Note that L_i = p where
$M_{ip} = \max \{ M_{ij} \}$ for j=1..J

4.2.1 Greedy-Based Approach For Object Localization

It is well-known that Greedy is a heuristic algorithm, which simply selects the best alternative at each iteration. A possible approach for object extraction is to label each segment and then apply a greedy-based heuristic algorithm for combining the neighboring segments having the same label. The pseudo-code of the algorithm is given Table 6. Figure 16 shows the process of object localization with the greedy-based algorithm.

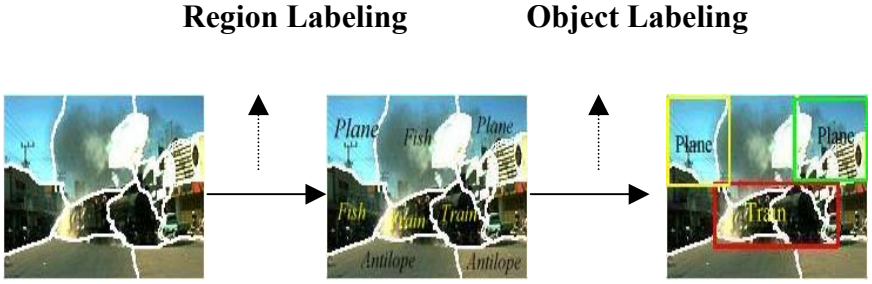


Figure 16: Process of Greedy-Based Object Localization Algorithm.

Table 6: Greedy-Based Localization Algorithm for a Segmented Image.

Algorithm: Greedy-Based Object Localization

Region Labeling

Calculate the membership value M_{ij} , of each segmented region R_i , for each query class j by using the training module for segments.

For each of the R_i repeat

Find p where M_{ip}= max { M_{ij} } for j=1..J

Assign the label of each segmented region R_i as L_i = p

Set $L_{i'} = L_i$ ($L_{i'}$ keeps the initial labels of all regions)

End For

Region Merging

Set each of the regions R_i as unprocessed.

For all of the unprocessed regions R_i repeat

For all of the unprocessed regions R_k repeat

If R_i is in the neighborhood of R_k and $L_i = L_k$

Merge, R_i and R_k and set R_i and R_k as processed

Set $R_i = R_i \cup R_k$

End If

End For

End For

Object Labeling

For each R_i, repeat // R_i consists of merged regions having the same label

Calculate the membership value M_{ij} , of each segmented region R_i , for each query class j by using the training module for objects.

Relabel R_i as L_i by selecting the label of the maximum membership of M_{ij}.

If $L_i = L_{i'}$ then store R_i and its label L_i

End For

In the greedy-based approach, a region is simply assigned to the label, which has the maximum membership value. As an example, a region with a sky label having a membership value of 0.8999 and with a plane membership value of 0.90 results in a

label of plane (see Figure 16). This crisp logic, behind the algorithm may cause misclassification of many regions with a very slight difference in the membership values. In order to solve the problems of Greedy-based approach, we propose a data structure, called "Neighborhood Tree".

4.2.2 Neighborhood Tree For Object Localization

Neighborhood Tree is defined as a directed acyclic graph with no circuits, which is constructed to concatenate over-segmented regions for object localization. The absence of circuits means that there is always exactly one path to get from one vertex of the tree to any other. The crisp labels of the greedy-based approach are replaced by a more flexible structure, which consider all the fuzzy membership values and their associated labels under a tree data structure.

Initially, the label and membership value of each segmented region of the image is computed by the training module. In a given image, for each query object, we construct a Neighborhood Tree as follows: First, we form a list, consisting of the regions with maximum membership values, which have the same label as the query object. Among the regions in the list, the one having the maximum membership value is selected as the starting node of the Neighborhood Tree. For each neighboring region of the starting node, a child node is added to the tree. The tree grows downward adding neighboring regions at each level. Note that, we form only one Neighborhood Tree for each query object, in a given image. If there exist more than one objects in the image, one can restart the construction of the Neighborhood Tree as long as there are uncovered regions in the list.

One of the major problems in construction of the Neighborhood Tree is the complexity introduced by each additional layer. To avoid this problem, a pruning algorithm is developed after the first level neighbors of the starting node: When a new node is formed in the tree, the training module computes the new membership values and the corresponding labels for the collection of the regions representing this node. If the label of the maximum membership value does not match to the label of the starting node, the path is pruned.

Figure 17 illustrates the construction of the Neighborhood Tree with a pre-segmented sample image for query object "plane". Note that there are 5 regions and "Region 1" is selected as the starting node of the Neighborhood Tree. For each neighboring region of "Region 1", a child node is added to the tree. At the second level, the node representing the concatenation of regions 1,2 and 3 is pruned. Since it is not recognized as "plane" by the training module.

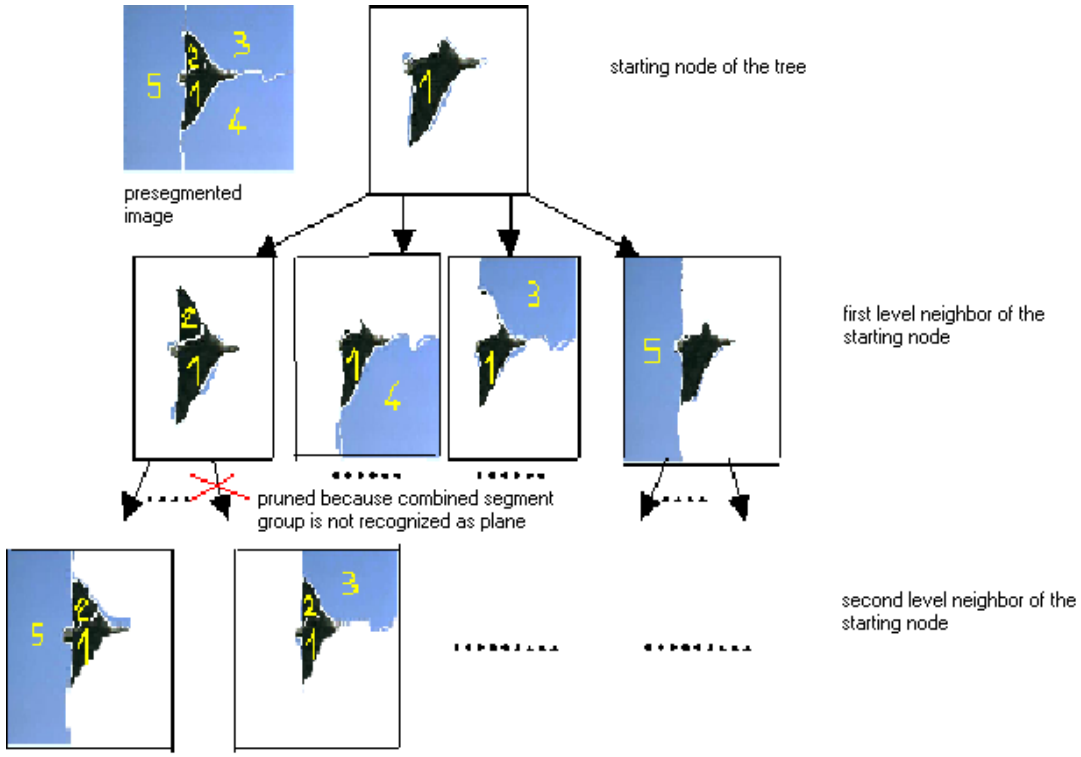


Figure 17: Construction of the Neighborhood Tree for a Pre-segmented Image.

The membership of each node is obtained using the training module for objects during the construction the Neighborhood Tree. The node having the maximum membership value is stored at a local variable. Therefore, there is no need for the tree traversal for object localization. At the end of the construction, the node having the maximum membership value, which was stored in the local variable, is labeled as the query object. This algorithm is given by the pseudo-code in Table 7:

Table 7: Construction of the Neighborhood Tree for a Query Object.

Algorithm: Neighborhood Tree Construction and Object Localization

```
public void main()
{
   Set M = 0;                                  //  indicates the initial maximum membership value
  R_i = findStartingNode(queryLabel);          //   finds starting node of the tree
  generateTree(R_i ,1, queryLabel);            //   generates the neighborhood tree recursively
  Store R with membership value M.             //   localizes objects
}


public Region findStartingNode(int queryLabel)
{
 Calculate the membership value M_ij , of each segmented region R_i , for each query class   j by
 using the training module for segments.

 Form a list, consisting of the regions with maximum membership values, which have the same
 label as the queryLabel.

 Among the regions in the list, select the region with maximum membership value as the starting
 node of the Neighborhood Tree and return that region.
 }

public void generateTree(Region Ri ,int level, int queryLabel)
{
  If checkForLocalization(Ri, level, queryLabel) == TRUE        // Checks for prunning
       For all regions R_k which are in the neighborhood of Ri repeat
         generateTree(R_i U R_k ,level+1,queryLabel);  //this function is called recursively for R_i U R_k
       End For
  End If
}

public Boolean checkForLocalization (Region Ri ,int level, int queryLabel)
{
  Find L_i and M_i                             // L_i = p , where M_ip= max { M_ij } for j=1..J
  If (level < 3) ||  (L_i == queryLabel)       // level < 3 because no pruning in the first 2 layers
      If (M_i > M) && (L_i == queryLabel)
         M = M_i;                              // M stores the maximum membership value
         R  = R_i;                             // R stores the localized object region
      End If
      Return TRUE;
  End If
  Return FALSE;
}
```

The output of the Neighborhood Tree provides us an image database indexed by the labels and membership values of the set of segmented regions, each of which corresponds to an object. The labeled objects are stored to be retrieved in the querying stage of the developed system.

The drawback of the Neighborhood Tree is the complexity introduced in creating the tree. Greedy-Based Approach is only a path in the Neighborhood Tree. In fact, the Greedy approach is a specialised case of the Neighborhood Tree, where at each level, only the node having the same label as the starting node and having the maximum membership value is added to the tree. Although, pruning algorithm decreases the complexity of the Neigborhood Tree approach, it is much more time consuming than the Greedy-Based Approach.

Considering the assumption that there are average of *n* regions in an object, each region has approximately 4 neighbours corresponding to main directions (left, right, top, bottom) and at each level half of the nodes are pruned; the complexity of the Neighborhood Tree algorithm can be approximately computed as $O(2^n)$. On the other hand, the complexity of the Greedy-based algorithm is $O(n^2)$.

4.3 Chapter Summary

In this chapter, the contributions in the proposed Content Based Image Retrieval System are emphasized. Three new methods are proposed to design a dedicated feature space for each object class, which are best representative descriptor, descriptor based AdaBoost and hierarchical learning schema based on Adaptive Resonance Theory. The Greedy-Based and Neighborhood Tree approaches are also introduced to attack the object localization problem in an image database.

CHAPTER

5 EXPERIMENTAL RESULTS

The proposed content-based image retrieval framework is developed in C++ Builder and tested over a subset of Corel Draw image database, UMIST [83] and ORL [84] face databases. We compare the performance of our framework with one of the most popular CBIR systems, available in the literature, namely, the ALIP in [36]. We also, tested the performance of our hierarchical learning architecture by comparing it with the face recognition systems of [83] and [84], on the UMIST and ORL databases. Examples of the Corel images are given in Appendix A.

In [36], Wang et al. tests the performance of ALIP system on a data set from Corel, which contains 10 image categories each of which consists of 90 images. Among them, 40 images are spared for training and the rest is used for testing. The UMIST Face Database consists of 565 images of 20 categories, where 290 of them are used for training and 285 for testing. ORL face data set contains 40 categories of faces, where each category contains 10 face images, with half of the data, labeled for supervision.

As mentioned before, this study is based on two important ideas: The first idea is to extract a dedicated feature space for each object class. The second one is to convert the task of object localization into a tree search problem in an over-segmented image. Therefore, two set of experiments are designed to validate the power of the proposed system based on the ideas mentioned above.

In the first set of experiments, we compare our methods to the some of the popular methods [36], [83], [84], available methods in the literature. The experiments are performed on data sets of ALIP system in [36], UMIST system in [83] and ORL system in [84].

In the second set of experiments, we test the performance of the Greedy-Based and Neighborhood Tree approaches with the data set used in [63]. In the following sections, the details of the experiments are given.

5.1 Experiments On ALIP Data Set

In order to find the best set of representative features for each query object, we employ the following methods:

1. Best Representative Descriptor using Euclidean Distance,
2. Feature-based AdaBoost,
3. Descriptor-based AdaBoost,
4. Fuzzy ARTMAP,
5. Hierarchical Learning Schema based on Adaptive Resonance Theory.

We evaluate these methods according to 4 criteria:

1. Selection of parameters,
2. Classification performances,
3. The relationship between the number of training samples and learning rate,
4. The relationship between the number of training classes and performance.

The method, which yields the best classification performance, is employed in the training module, in the proposed retrieval system.

5.1.1 Selection of the Parameters

The goal of this set of experiments is to identify the best parameter values for each of the five methods, namely, Best Representative Descriptor, fuzzy ARTMAP, feature-based AdaBoost, descriptor-based AdaBoost and hierarchical learning system based on Adaptive Resonance Theory. The best parameter values for each method are used in the rest of the experiments.

5.1.1.1 Best Representative Descriptor

As it is mentioned in Chapter 4.1.1, in this method, the images in each class are queried by all the descriptors using the Euclidean distance. The descriptor, which maximizes the precision value of each class, is selected as the Best Representative Descriptor of that

class. “The number of retrieved images" is an important parameter, which determines the precision value.

Table 8 shows the effect of this parameter on the precision for the training and test data set of ALIP system. Figure 18 shows the graphical representation of Table 8.

An evaluation of Table 8 indicates that, even for the very low number (5) of retrieved images, the best representative feature cannot approach to 100% in the training set. This is an indication of an under-fitting problem, where the Best Representative Descriptor becomes saturated without learning the training data set. Note that, the highest precision value is 93.75%, which is achieved by setting the "The number of retrieved images" parameter to 10. The major problem of this method is the restriction of the feature space to only one descriptor.

Table 8: Effect of # Query Results to the Performance.

# OF RETRIEVED IMAGES	TRAINING PERFORMANCE%	TEST PERFORMANCE %
5	93,75	75,4
10	93,75	77,6
15	88,75	73,4
20	87	75
25	86,5	75,8
30	87,75	75,4
35	86,75	74
40	85,75	73,8

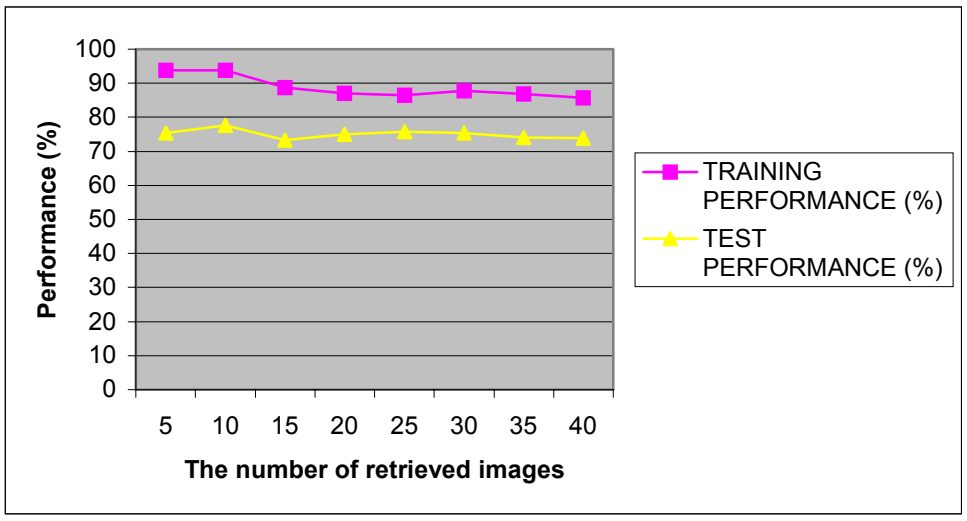


Figure 18: The Performance / # of Retrieved Images Graph of BRD.

5.1.1.2 Feature-Based AdaBoost

Feature-based AdaBoost is implemented with the pseudo-code given in Table 2 of Section 2.4.2. The input feature vector of size 352 is formed by concatenating the color structure, edge histogram, Gabor and Haar descriptors. The parameter, which affects the performance of this algorithm, is “The number of selected features". Remember that AdaBoost algorithm selects a single feature as weak classifier, at each iteration. This parameter indicates the total number of features selected by the AdaBoost algorithm. Note that relatively small number of features is selected among the whole feature vector.

We perform our experiments with different values of “The number of selected features" parameter in range [1,50]. The upper limit of 50 features is sufficient to catch the 100% recognition rate in the training data. Table 9 shows the effect of this parameter on the performance for the training and test data set of ALIP system together with the graphical representation of Figure 19.

Note that, the performance of the algorithm exponentially increases as the number of features is increased to 20-30. Increasing the dimension of the feature space more than 30 does not provide any improvement in the performance, indicating saturation. Therefore, during the experiments the upper limit of 30 features are selected by the AdaBoost algorithm.

An observation from Table 9 is that feature-based AdaBoost algorithm has an over-fitting problem. Even with a small number of features, which is 30, it achieves a performance of 100% with the training set. However its performance for the test set decreases drastically. This result shows the `memorization` tendency of the algorithm.

Table 9: Effect of # of Selected Features to the Performance.

# OF SELECTED FEATURES	TRAINING PERFORMANCE%	TEST PERFORMANCE %
1	57,5	53
5	86,5	71,8
10	93,75	76,4
15	98	75,8
20	99,25	77,4
25	99,75	75,4
30	100	78,2
35	100	77,2
40	100	77
45	100	77,6
50	100	77

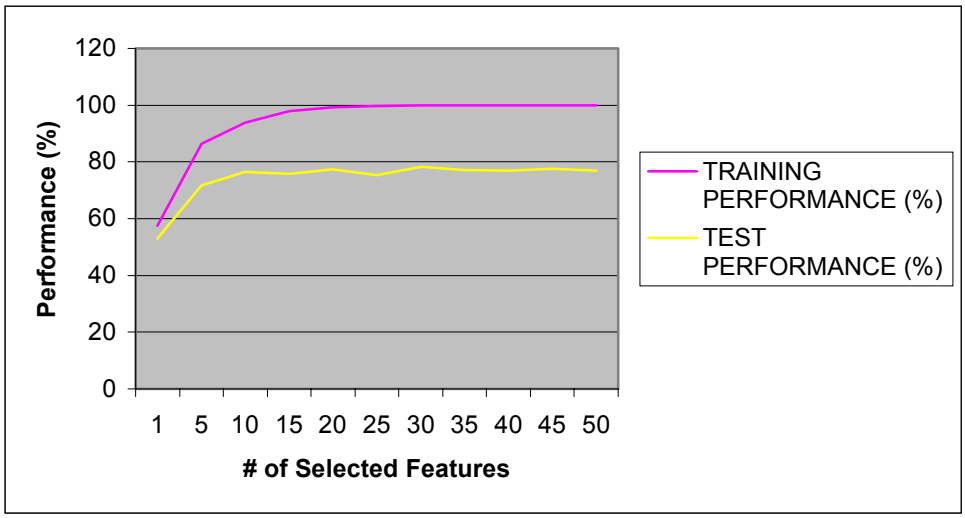


Figure 19: The Performance / # of selected features Graph of Feature-based AdaBoost.

5.1.1.3 Descriptor-based AdaBoost

Descriptor-based AdaBoost is implemented with the pseudo-code given in Table 4 of Section 4.1.2. Color structure, edge histogram, Gabor and Haar descriptors are used as the weak classifiers of the algorithm. Since it is simple to implement, K-nearest neighborhood algorithm is used for evaluating the performance of each descriptor.

"The number of descriptors" and "k" in K-nearest neighborhood algorithm are the major parameters affecting the performance of this method. Table 10 shows the

effect of the parameters on the performance for the training and test data set of ALIP. Note that, setting "the number of descriptors" and "k" parameters to 3 and 10 respectively achieves the highest performance.

The logic behind the AdaBoost method is to increase the weights of the miss-classified samples, so that the algorithm searches new descriptors to correct the samples in the wrong classes. However, if the miss-classified samples can not be classified correctly by all of the descriptors, boosting them becomes meaningless. This is the reason why increasing the number of descriptors does not increase the performance of the algorithm after three descriptors, as indicated in Table 10. In summary, this method needs a large pool of descriptors to be able to boost them.

Table 10: Effect of # of Descriptors and "k"to the Performance.

# OF DESC.	K-NN (K PARAMETER)	TRAINING PERFORMANCE %	TEST PERFORMANCE %
2	5	88	73,6
2	10	85,5	75,2
2	15	86,25	73,4
2	20	84,5	74,4
2	25	79,5	74,4
3	5	88,25	76
3	10	88,5	78,4
3	15	86,25	76,2
3	20	84,25	76,6
3	25	84,5	74,6
4	5	87,75	70,6
4	10	86,25	76,6
4	15	87,5	73,8
4	20	85,5	76,2
4	25	84,75	75,2

5.1.1.4 Fuzzy ARTMAP

The performance of Fuzy ARTMAP is very much dependent on the "vigilance" parameter, which calibrates the level of categorization. High vigilance results in finer categorization, whereas the low vigilance allows a broader categorization. In the extreme cases, when the vigilance is 1, all the input patterns are put into different categories and final number of category nodes is equal to the number of input patterns. When the vigilance is 0, all the input patterns are put into the same category and the final number

of category nodes is 1. In summary, according to the value of this parameter, the input vectors are mapped to the classes with different levels of categorization.

The experiments, repeated with different values of this parameter are tabulated in Table 11. Note that, the highest performance for the test data set is obtained as 82.8%, when the "vigilance" parameter is set to 0.8. The number of category nodes for this set-up is 30, where each training class is represented with an average of 3 category nodes. This shows that the image classes in ALIP test data set can be represented by an average of 3 category nodes.

Table 11: Effect of Vigilance to the Performance.

VIGILANCE	TRAINING PERFORMANCE%	TEST PERFORMANCE %	NUMBER OF CATEGORY NODES
0.95	100	61,6	251
0.93	99,75	67,4	168
0.90	99,75	80	122
0.87	99,75	78	95
0.85	99,75	76,2	63
0.83	99,5	81,2	41
0.80	99,75	82,8	30
0.75	98,75	81,2	16
0.70	98,75	82,4	8
0.65	99	79,2	6
0.60	91,25	80,6	6
0.55	97,75	79,2	6
0.50	97,5	77,4	6

Figure 20 shows the graphical representation of Table 11.

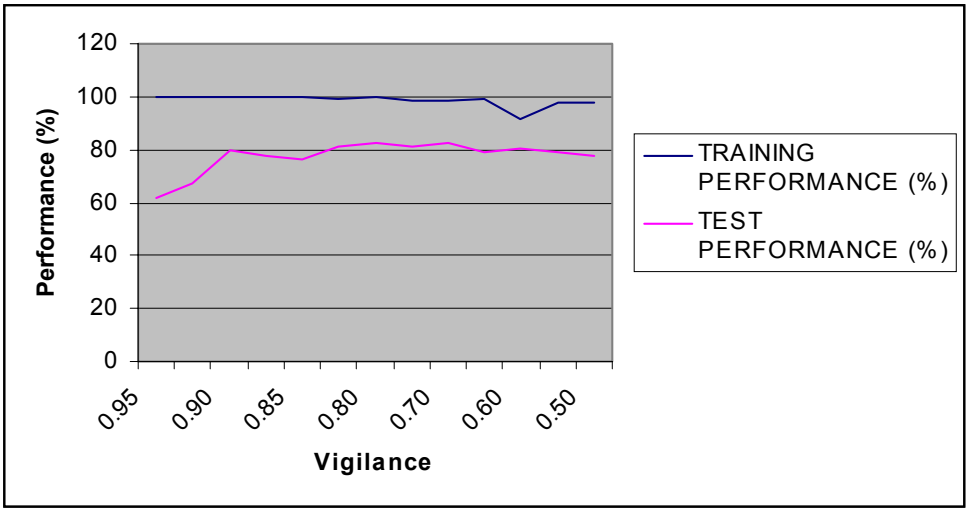


Figure 20: Vigilance / Performance Graph of FAM.

5.1.1.5 Hierarchical Learning System Based On Adaptive Resonance Theory

There are two critical parameters in this approach. The first one is the "vigilance" which is used in fuzzy ARTMAP classifier in the upper layer of the architecture. The second one is the "k" parameter in K-nearest neighborhood classifier of the base-layer.

Table 12 shows the effect of these parameters on the performance. Note that the proposed system is quite stable to the changes of vigilance and k-parameter, especially for the training set, where the performance is nearly 100% for all values of the parameters. The worst performance obtained with this algorithm, which is 82.6%, is higher than the best performances of the other methods, explained above. Throughout the experiments, k is taken as 10 and vigilance is set to 0.93.

Table 12: Effect of K and Vigilance to the Performance.

K-NN (K PARAMETER)	VIGILANCE	TRAINING PERFOMANCE %	TEST PERFOMANCE %
5	0.95	100	83
5	0.93	100	83,2
5	0.90	100	84,2
5	0.87	99,75	85
5	0.85	100	84
5	0.83	100	83,8
5	0.80	100	82,6
10	0.95	100	85,4
10	0.93	100	86,8
10	0.90	100	84,4
10	0.87	100	86
10	0.85	99,75	86,2
10	0.83	100	85
10	0.80	99,5	82,6
15	0.95	100	85,4
15	0.93	100	85
15	0.90	100	83,6
15	0.87	100	84,2
15	0.85	99,75	84,6
15	0.83	99,5	83,8
15	0.80	99,5	83,6
20	0.95	100	84,8
20	0.93	100	85,4
20	0.90	100	82,8
20	0.87	100	85,2
20	0.85	99,75	85,4
20	0.83	99,25	85,2
20	0.80	99,5	85,8

In the above experiments, the training algorithms are tested in a range of parameter values. The parameters, corresponding to the best performance, are summarized in Table 13. F-AdaBoost and D-AdaBoost represent feature-based and descriptor-based AdaBoost methods, respectively.

Table 13: Best Performances and Corresponding Parameter Values.

TRAINING ALGORTIHM	PARAMETER NAME	BEST PARAMETER VALUES	PERFORMANCE (%)
BRD	# of Retrieved Images	10	77.6
F- AdaBoost	# of Selected Feautures	30	78.2
D- AdaBoost	# of Descriptors - K	3 - 10	78.4
Fuzzy ARTMAP	Vigilance	0.8	82.8
Hierarchical Learning	Vigilance - K	0.93 - 10	86.8

The parameter values in this table are used in the rest of the experiments. In the next section, the classification performances of the training algorithms proposed in this thesis are tested and compared to the classical training algorithms.

5.1.2 Comparison of Classification Performances

The goal of this set of experiments is to test the classification performances of Best Representative Descriptor, fuzzy ARTMAP, feature-based AdaBoost, descriptor-based AdaBoost and Hierarchical learning systems. We also, compare the above methods to the classical classification schema, which uses the same feature set for all classes. From now on, the later group is called fixed feature space methods.

Table 14 indicates the retrieval rates of 50 images for fixed feature space and the best representative feature space methods. Note that, in this table, hierarchical learning schema and majority voting methods are two-layered stack generalization approaches, whereas the others consist of only a single layer. In the table, "CS" stands for color structure, "EH" for edge histogram, "BRD" for the best representative descriptor, “FAM” for the fuzzy ARTMAP, "F-AdaBoost" for the feature-based AdaBoost and "D-AdaBoost" for the descriptor-based AdaBoost methods.

Table 14: Performance of the Methods Tested in This Study.

CLASSES	SINGLE LAYERED											TWO-LAYERED	
	FİXED FEATURE SPACE							BEST REPRESENTATIVE FEATURE SPACE					
	CS	EH	GABOR	HAAR	CS + EH	CS + EH+ GABOR	CS + EH+ GABOR+ HHAR	BRD	F-AdaBoost	D-AdaBoost	FAM	Hierarchical Learning	Majority Voting
0 Africa	39	21	27	29	38	33	39	40	37	38	40	38	38
1 Beach	30	26	24	36	32	31	39	31	30	33	40	38	34
2 Building	39	23	22	4	41	39	38	39	33	37	37	38	38
3 Buses	39	45	41	10	44	47	39	38	45	46	46	47	39
4 Dinosaurs	50	50	50	50	50	50	50	50	50	50	50	50	50
5 Elephants	26	24	30	33	30	31	36	29	27	28	37	37	32
6 Flowers	46	41	44	31	49	49	49	46	43	39	49	48	47
7 Horses	49	45	34	50	50	50	50	50	50	49	50	50	49
8 Mountains	27	21	13	28	28	30	30	25	39	32	30	40	28
9 Food	41	13	30	2	39	45	30	40	37	40	35	48	40
TOTAL	386	309	315	273	401	405	400	388	391	392	414	434	395
%	**77,2**	**61,8**	**63**	**54,6**	**80,2**	**81**	**80**	**77,6**	**78,2**	**78,4**	**82,8**	**86,8**	**79**

In the fixed feature space methods, k-NN is used as the recognition engine. Among four descriptors, Color Structure (CS) gives the best result with 77.2% classification accuracy. This result indicates that color characteristics are more discriminative than the other features to differentiate 10 categories of the ALIP data set. Concatenating the color structure, edge histogram and Gabor descriptors increases the performance of k-NN to 81%. At this point it is very difficult to decide which additional descriptor is to be concatenated to further increase the overall classification performance. Additionally, there is no guarantee that this set of descriptors results in a similar performance in another data set.

In the variable feature space methods, using single layer architecture, F-AdaBoost and FAM are used as the wrapper methods. The input to these single layer architectures is the 352-dimensional feature space obtained by concatenating all the

descriptors under the same feature vector. As it is seen from Table 14, with an 82.8% classification rate, FAM has a better performance than the F-AdaBoost which only reaches to 78.2%. On the other hand, since the BRD method selects only a single descriptor, the performance of 77.6% remains below the F-AdaBoost. This fact indicates that the dimension of the best representative descriptor is not sufficient to discriminate the categories of ALIP. D-AdaBoost extends BRD method to k-best representative descriptors by boosting the performances of the descriptors for each object class. The performance of D-AdaBoost method is increased to 78.4%. However, this result is still below of the performance of FAM.

The superiority of FAM compared to the AdaBoost and BRD methods in the image classification and retrieval problems is due to two major reasons: First of all, AdaBoost selects the most salient features and weights them to form a strong classifier from the weak classifiers, whereas Fuzzy ARTMAP weights all the features and use almost all of them. Secondly, fuzzy ARTMAP architecture introduces new nodes to split an object class in case it has `dissimilar` features. Therefore, the Fuzzy ARTMAP method can assign dissimilar vectors into the same object class. AdaBoost can not cope with distinct representations of the same class.

Table 15 shows the BRD for each of the classes obtained in the training phase. Note that Gabor descriptor never wins against the other features to become the BRD, whereas, the color structure and Haar become BRD for 5 and 4 objects, respectively. The most widespread descriptor, the Color Structure, indicates that ALIP data set is rather sensitive to color information. Selection of HAAR is due to the multi-resolution nature of the classes of beach, elephant, mountain and horses. It is quite intuitive to select the edge histogram for the bus class, due to the dominance of shape compared to color. Finally, the ALIP data set is not sensitive to directional texture information. This characteristic of the database eliminates the Gabor descriptor.

Table 15: Best Representative Descritors for Different Object Classes.

BRD	OBJECT CLASSES
Color Structure (CS)	Africa, Building, Dinesours, Flowers , Food
Edge Histogram (EH)	Buses
Gabor	-
Haar	Beach, Elephants, Horses, Mountains

Table 16 shows the descriptors chosen by the D-AdaBoost algorithm. Note that three descriptors are boosted to obtain a strong classifier as a combination of weak classifiers. Descriptor- and feature-based AdaBoost methods perform very similar on the ALIP data set. This result is rather surprising, considering the computational complexity required for the D-AdaBoost, which is more than that of the F-AdaBoost. We believe that increasing the number of descriptors may result in a positive impact on the boosting.

Table 16: Descriptors Used for Descritor-Based AdaBoost Algorithm For each Object Class.

CLASS	1. DESCRIPTOR	2. DESCRIPTOR	3. DESCRIPTOR
Africa	CS	EH	GABOR
Beach	HAAR	CS	GABOR
Building	CS	GABOR	EH
Buses	EH	CS	GABOR
Dinesours	CS	HAAR	GABOR
Elephants	HAAR	GABOR	CS
Flowers	CS	EH	HAAR
Horses	HAAR	CS	GABOR
Mountains	HAAR	EH	CS
Food	CS	GABOR	EH

Finally, the proposed hierarchical architecture surpasses all of the methods mentioned above with 86.8%. In order to gain some insight for the internal learning mechanism of FAM in the second layer, we plot the average of the recognition nodes created in ART_a in response to the samples of each class, in Figure 21. As it is observed from the figure, FAM successfully learns the degree of importance of each feature for

recognizing a particular class. For example, for the `bus` and 'food' classes, HAAR features are relatively less important then the other features (See Figure 22 and Figure 23). This result is consistent to the results obtained in Table 14.

In summary, when a fixed descriptor is used during the queries, combination of color structure, edge histogram and Gabor descriptors gives the highest retrieval rate, which is 81%. Majority voting of descriptors yields only 79% classification accuracy. However, if the hierarchical learning schema is used as the recognition engine, then the correct retrieval rate reaches to 86.8%, which significantly outperforms the other methods.

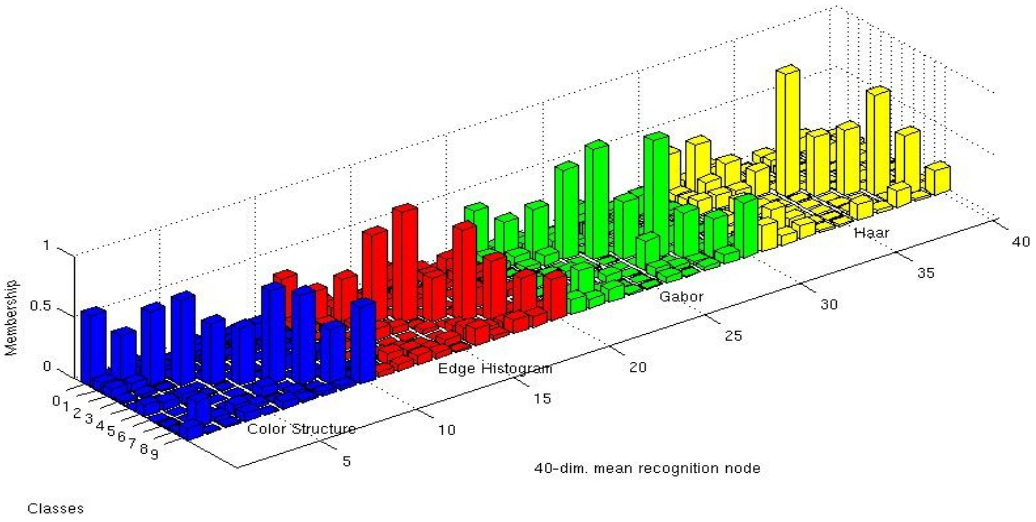


Figure 21: Plot of Mean Recognition Nodes for Each Class. Refer to Table 14 for the Interpretation of the Class Numbers.

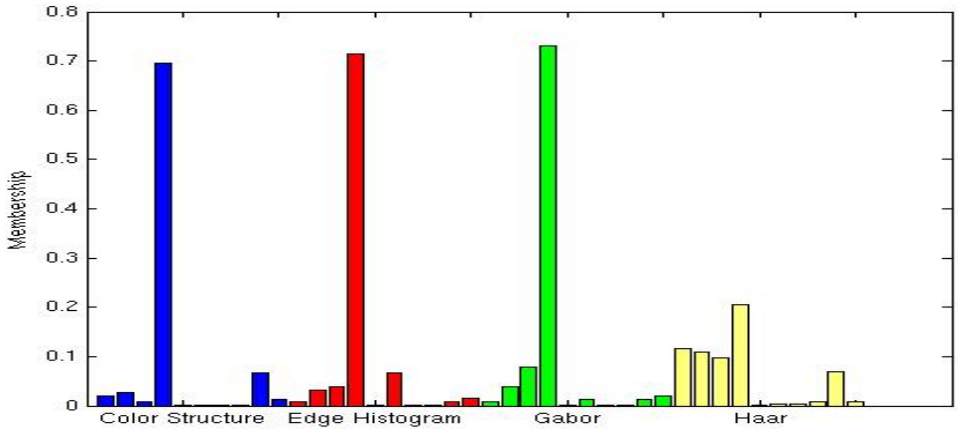


Figure 22: Weights of the Mean Recognition Node for the Samples of the BUS Class.

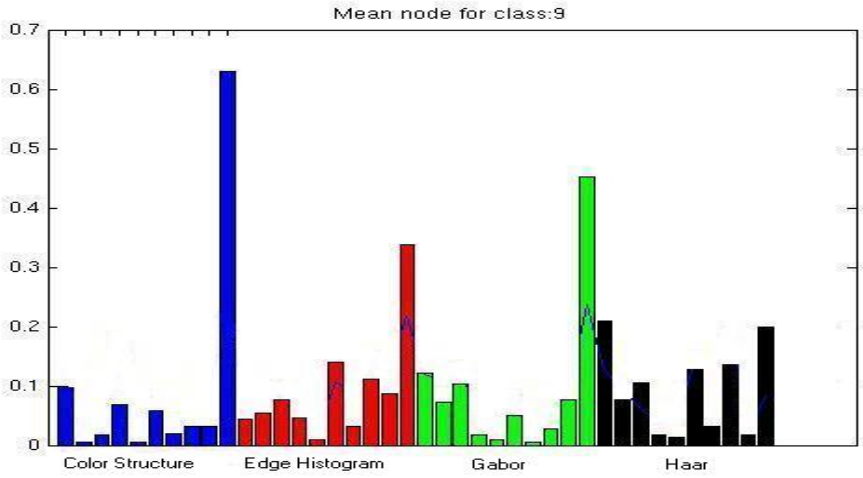


Figure 23: Weights of the Mean Recognition Node for the Samples of the FOOD Class.

5.1.2.1 Comparison of the Proposed Multi-Layered Architecture to the ALIP System

In [36], Jia Li and James Z. Wang present a system for Automatic Linguistic Indexing of Pictures (ALIP). To provide numerical results on the categorization performance, they evaluate the system, based on a controlled subset of the COREL database, formed by 10 image categories, given in Table 17. Instead of annotating the images, their program was used to select the category of each test image. They use the classification power of the system as an indication of the annotation accuracy. An image is considered to be annotated correctly, if the computer predicts the true category the image.

Table 17 shows the automatic classification result of ALIP system. Each row lists the percentage of images assigned to one of the 10 categories by the computer. Numbers on the diagonal show the classification accuracy for every category.

Table 18 gives the recognition performance of the proposed hierarchical architecture. It is observed that the proposed method performs better than the ALIP system in image categorization. The average performance of ALIP for 10 categories is 63.8%, whereas the average performance of proposed hierarchical schema is 86.8%.

Table 17: ALIP Classification Performance.

	Africa	Beach	Building	Buses	Dinesours	Elephants	Flowers	Horses	Mountains	Food
Africa	**52**	2	4	0	8	16	10	0	6	2
Beach	0	**32**	6	0	0	0	2	2	58	0
Building	8	4	**64**	0	8	6	0	0	6	4
Buses	0	18	6	**46**	2	8	0	0	16	4
Dinesours	0	0	0	0	**100**	0	0	0	0	0
Elephants	8	0	2	0	8	**40**	0	8	34	0
Flowers	0	0	2	0	0	0	**90**	0	2	6
Horses	0	2	0	0	0	4	24	**60**	4	6
Mountains	0	6	6	0	2	2	0	0	**84**	0
Food	6	4	0	2	6	0	8	0	6	**68**

Table 18: Proposed Hierarchical Learning Schema's Classification Performance.

	Africa	Beach	Building	Buses	Dinesours	Elephants	Flowers	Horses	Mountains	Food
Africa	**76**	2	6	2	2	4	2	2	0	4
Beach	0	**76**	4	2	0	2	0	2	14	0
Building	3	2	**76**	1	0	0	0	2	4	0
Buses	0	0	4	**94**	0	0	0	2	0	0
Dinesours	0	0	0	0	**100**	0	0	0	0	0
Elephants	4	8	4	0	0	**74**	0	2	8	0
Flowers	2	0	2	0	0	0	**96**	0	0	0
Horses	0	0	0	0	0	0	0	**100**	0	0
Mountains	2	18	2	0	8	0	0	0	**80**	0
Food	0	2	2	0	0	0	0	0	0	**96**

The comparison of Table 17 and Table 18 indicates a striking performance of the proposed system compared to ALIP. There are many reasons for this result: First of all, the dimension of the feature space of the ALIP system is much smaller (6 features/block) than that of the hierarchical system, which varies between 32-192/image. However, note that our feature vector represents the entire image, whereas ALIP system extracts the feature vectors by a multiple of the number of blocks in each image. Secondly, the 2-D Hidden Markov Model is faster than the proposed system, bringing a substantial amount of efficiency to the system. Finally, utilization of the same feature set for all the categories brings a serious bottleneck to the ALIP system. This fact is clearly observed form Table 17, where the performance of the retrieval is significantly reduced for the Bus category, compared to our system, since there are no shape features in the ALIP system. The low performance of ALIP on the beach category is due the complex nature of the beach images, which contain many dissimilar indexes such as human, sand, ocean, travel etc. As a result the proposed 6 dimensional feature falls short in discriminating the beach images. On the other hand, the second layer of the proposed architecture is capable of assigning the features that are not close to each other into the same category by introducing more than one node for each category, which allows many-to-one mapping. Similar arguments are valid for the elephant category.

5.1.3 The Relationship Between the Number of Training Samples and the Learning Rates

Let us now test the influence of number of training samples on the learning rates of Best Representative Descriptor, feature-based AdaBoost, descriptor-based AdaBoost, fuzzy ARTMAP and Hierarchical learning methods. We start to perform experiments with a training data set of size 5 for each class. At each step, experiments are repeated by incrementally adding 5 new samples to the classes.

Table 19 shows the learning rates as a function of number of samples for different classifiers. Note that: the most tolerant method to the small number of training samples is the proposed hierarchical learning schema, with learning rate of %66.2 even with 5 training samples for each class. The learning remains high compared to other methods as we increase the size of the training set. Therefore, the layered architecture has an increased learning capacity compared to the single layered systems.

Table 19: The Effect of Sample Count to Different Classification Methods.

# of Training Samples	CS	EH	GABOR	HAAR	CS+EH+GABOR +HAAR K-NN	BRD	F-AdaBoost	D-AdaBoost	FAM	Hierarchical Schema
5	60	42,4	45	41,2	61,4	49,8	52,2	55,6	58	66,2
10	67,6	52,6	54,5	46	66,4	64,2	63,8	64,4	70,2	72,8
15	67,2	57,6	55	46,6	68,4	67,6	67,6	68,2	70,4	77,8
20	68,4	59,4	56	48,6	70	68,2	71	71	75,2	82,6
25	68,4	60,4	56,2	49,2	72	71,8	72,4	72,4	75,4	81,6
30	72,4	60,8	58,8	50	76,4	74,4	78,4	75,8	79	84
35	74,6	60,8	60,8	53	78,6	77,8	77,6	73,4	82,8	83,8
40	77,2	61,8	63	54,6	80	77,6	78,2	78,4	82,8	86,8

Figure 24 shows the graphical representation of Table 19. The performance of all of the methods monotonically increases as the number of training samples also

increases, which is consistent with our intuition. However, some slight decreases occur related to the characteristics of the added samples. If the samples do not represent the training classes, then the performance of all of the methods decreases.

Another observation from Figure 24 is that the performance curves intersect at some points, such as edge histogram and Gabor. This result is due to the characteristics of the samples, incrementally added to the data set. Some of the samples can be represented by the edge histogram descriptor, but some others are better represented by the Gabor descriptor to classify the correct category of the images.

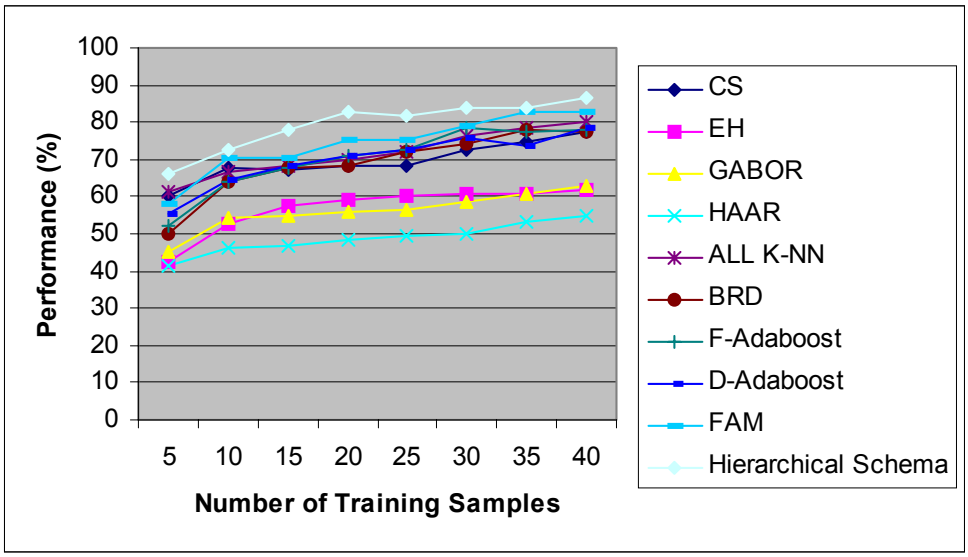


Figure 24: The Effect of Number of Training Samples to Different Classification Methods.

5.1.4 The Relationship between Number of Classes and Performance

This set of experiments is designed to explore the relationship between the number of classes and the performance of Best Representative Descriptor, feature-based AdaBoost, descriptor-based AdaBoost, fuzzy ARTMAP and Hierarchical learning schema based on Adaptive Resonance Theory. For this purpose, we increase the total number of classes to 20, by adding extra 10 classes from Corel Draw to the ALIP data set. We calculate the performance of each method starting from a two-class problem and incrementing until 20-class.

Table 20 shows the relationship between the number of classes and the performance of various methods. Note that the most tolerant methods to the increase in the number of training classes are the proposed hierarchical schema and FAM methods, which achieve nearly % 70 performances even with 20 training classes. The decrease of the performance is rather stable compared to other methods, as we increase the category number from 2 to 20. On the other hand, the other methods, especially, the fixed feature space methods are more sensitive to the increase in the number of classes. This fact is observed from Table 20, which indicates average of 40% percent decrease in the fixed feature based method, whereas 30% percent decrease in the variable space method.

The performances of BRD, F-AdaBoost and D-AdaBoost are nearly %60 when the number of classes is increased to 20. The poor performance of BRD is due to the insufficiency of the dimension to discriminate even small number of classes. Since BRD method selects only one descriptor for each class, it suffers from discriminating the samples of 20 classes. This fact is explained in detail in Section 5.1.2. On the other hand, 4 descriptors of D-AdaBoost are quite short to boost for obtaining a strong classifier when the number of classes is increased. Finally, 30 features of F-AdaBoost cannot cope with the increase in the number of classes.

As the number of training classes increases, the performances of the hierarchical schema and FAM methods get closer to each other. It is well-known that the performance of k-NN recognition engine decreases as the number of classes increases. Since k-NN recognition engine is used as the base level of the hierarchical schema; the performance loss of k-NN negatively affects the performance of the hierarchical schema.

Figure 25 shows the graphical representation of Table 20. The performance is inversely proportional to the number of training classes. However, some slight increases occur related to the characteristics of the added class. As an example, the fifth class in the ALIP data set is the 'dinosaurs'. All of the methods have a performance of %100 in classification of dinosaurs images. Therefore, adding the dinosaur class increases the total performance of the system.

Figure 25 indicates that the performance curves intersect at some points. This result is also due to the characteristics of the added classes. Adding a new class may affect the performance of one descriptor in positive and other one in negative manner.

Table 20: The Effect of Number of Training Classes on the Performance of Different Classification Methods.

# of TRAINING CLASSES	CS	EH	GABOR	HAAR	CS+EH+GABOR +HAAR K-NN	BRD	F-AdaBoost	D-AdaBoost	FAM	Hierarchical Schema
2	89,00	88,00	88,00	91,00	95,00	93,00	90,00	93,00	92,00	95,00
3	81,33	75,33	74,00	58,00	86,67	78,67	80,00	80,00	82,00	83,33
4	79,00	76,00	72,00	48,50	84,00	75,50	82,00	81,00	84,50	85,00
5	83,20	77,60	78,00	58,80	87,20	83,60	85,60	86,00	87,60	88,40
6	78,33	71,67	74,33	58,33	83,67	80,33	79,00	79,33	84,67	85,33
7	80,29	71,43	75,14	61,14	85,71	82,00	80,57	82,86	86,00	86,29
8	81,25	69,75	72,00	61,75	85,25	83,00	83,00	83,50	87,75	88,25
9	77,56	65,78	63,56	60,22	82,22	79,33	78,22	80,67	83,56	84,44
10	77,20	61,80	63,00	54,60	80,00	77,60	78,20	78,40	82,80	86,80
11	76,18	60,91	64,00	58,18	81,64	79,45	80,36	79,27	84,36	87,27
12	73,17	60,50	61,67	61,33	83,00	80,67	81,83	80,67	85,67	86,83
13	67,38	59,23	58,77	58,77	79,38	74,92	79,38	77,54	83,08	83,38
14	64,14	56,29	58,14	52,57	77,00	68,57	75,71	73,86	80,29	80,86
15	63,20	56,40	55,07	50,80	74,53	68,00	73,87	70,27	77,47	78,00
16	63,63	53,75	56,00	48,75	74,75	65,75	73,13	67,50	77,38	77,63
17	60,24	51,29	52,71	46,00	70,35	62,00	68,71	64,47	73,88	74,24
18	57,33	47,67	48,89	44,67	66,78	60,33	65,56	62,33	70,00	70,44
19	55,58	45,68	47,79	43,26	64,63	60,21	63,26	59,58	69,89	70,32
20	53,70	44,30	45,20	41,90	62,40	57,50	60,90	57,10	68,00	69,00

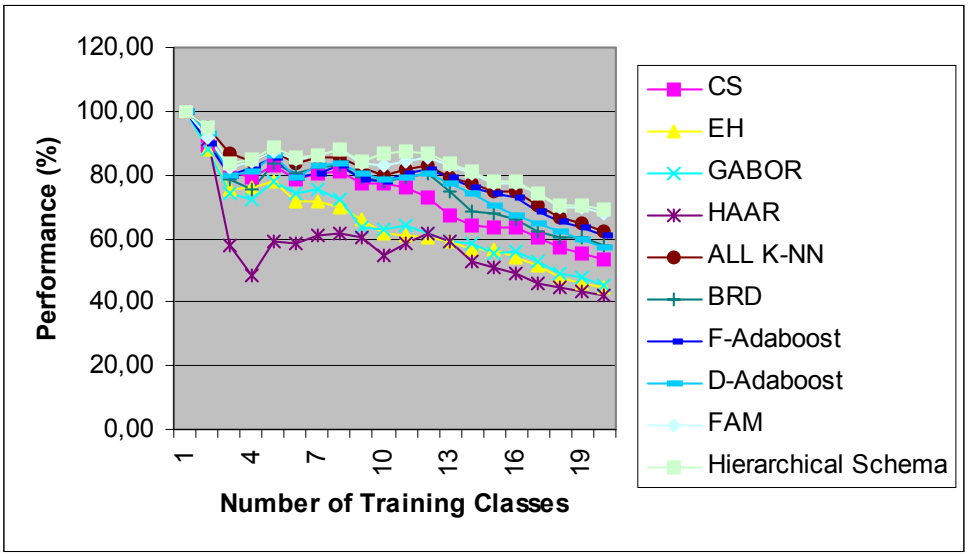


Figure 25: The Effect of Training Class Count to Different Classification Methods.

5.2 Best Representative Feature Experiments With Two Face Data Sets

It is well known that face recognition problem is an important hot research area with a lot of available benchmarks. For this reason, the performance of the proposed methods is tested in face recognition domain. Two standard database, namely, ORL [84] and UMIST [83] are employed for comparing the performance of the proposed system to those of the classical ones, as we did in the previous section.

The UMIST Face Database consists of 565 images of 20 people, each covering a range of poses from profile to frontal views. Subjects cover a range of race/sex/appearance. ORL data set contains 40 category of face, where each category contains 10 face images. For some of the categories, the images were taken at different times, varying the lighting, facial expressions (smiling / closed eyes / etc.) and facial details (glasses / no glasses / etc.) All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position.

We use Eigenface, Laplacianface, Fisherface, Haar and Gabor features [85], [86], [6], [82] during our experiments. These are the most popular and widely accepted features in face recognition problem.

Table 21 shows the values of the parameters used in the experiments. The parameters are selected by varying them to catch the best retrieval performance on the test data.

Table 21: Parameters Used in UMIST Data Set Experiments.

TRAINING ALGORTIHM	PARAMETER NAME	PARAMETER VALUES USED IN EXPERIMENTS
BRD	# of Retrieved Images	3
F- AdaBoost	# of Selected Features	100
D- AdaBoost	# of Descriptors - K	3 - 3
Fuzzy ARTMAP	Vigilance	0.93
Hierarchical Clasification	Vigilance - K	0.93 - 3

In both ORL and UMIST Face Data Set, the experiments are repeated 5 times by randomly splitting them into training and test sets. The UMIST date set consists of 290 training and 285 test images, whereas the ORL data set is split to equal size for training and test images. The average performance results obtained from the experiments are given in Table 22 and Table 23, respectively.

Table 22: Average Performance Results of Different Recognition Engines for UMIST Data Set.

	EIGENFACE	LAPLACIANFACE	HAAR	GABOR	ALL CONCATANETED	BRD	F-ADABOOST	D-ADABOOST	FAM	Hierarchical learning
DATA SET 1	81,75	75,09	88,42	83,16	87,72	83,51	84,91	84,21	87,02	92,98
DATA SET 2	87,02	76,14	91,93	83,86	91,23	87,02	90,53	84,91	93,33	92,63
DATA SET 3	89,82	79,65	94,04	88,77	94,04	90,53	89,12	91,58	85,61	94,74
DATA SET 4	87,37	79,65	89,82	85,96	90,53	87,37	92,98	84,21	78,95	92,98
DATA SET 5	86,67	82,46	89,12	80,35	89,47	86,32	87,02	84,91	79,65	91,93
AVERAGE (%)	**86,53**	**78,60**	**90,67**	**84,42**	**90,60**	**86,95**	**88,91**	**85,96**	**84,91**	**93,05**

Table 23: Average Performance Results of Different Recognition Engines for ORL Data Set.

	EIGENFACE	LAPLACIANFACE	FISHERFACE	HAAR	ALL CONCATANETED	BRD	F-ADABOOST	D-ADABOOST	FAM	Hierarchical learning
DATA SET 1	60,5	58,5	27,5	82	69,5	63,5	80,5	73,5	63	88,5
DATA SET 2	60	55,5	69	84,5	76,5	64,5	87,5	80	86	87
DATA SET 3	67	53	19	88,5	73	70,5	85	76,5	65	90
DATA SET 4	65	52,5	49,5	87,5	77,5	68,5	82,5	74,5	81,5	90,5
DATA SET 5	68	62	44,5	85	72,5	71,5	78	72,5	78,5	89,5
AVERAGE (%)	**64,1**	**56,3**	**41,9**	**85,5**	**73,8**	**67,7**	**82,7**	**75,4**	**74,8**	**89,1**

Table 22 and Table 23 summarize the classification performances of different classification algorithms for the UMIST and ORL face data sets. The performance of Eigenface, Laplacianface, Fisherface, Haar and Gabor descriptors for UMIST and ORL data set reported in the literature are very confusing and conflicting [83], [84]. It should be noted that the experiments in the literature depend on many factors. One of the most important determining issues is the utilization of the recognition engine. Using more complicated classification methods such as neural networks or some pre-processing techniques such as histogram equalization, normalization etc. can increase their performances. However, the aim in these experiments is not to compare the recognition engines with each other. Our basic goal is to show the power of the hierarchical architecture and its impact in decreasing the semantic gap problem for any type of recognition engine. In fact, in the proposed architecture, any recognition engine can be utilized instead of K-NN, as long as the output of the first layer yields a set of membership values for each class. During our experiments the *k* value are set to *3* in calculating the performances.

An analysis of Table 22 and Table 23 indicate that when a fixed descriptor is used, Haar gives the highest recognition rate with 90.67% for UMIST and 85.5% for ORL data sets. However, if the hierarchical learning schema is used for each object class, then the correct retrieval rates reach to 93.05% and 89.1% respectively, which significantly outperforms the fixed feature space methods.

Another interesting result of these experiments is that the performance of classical fuzzy ARTMAP architecture is %84.91 and %74.8, which is quite low compared to the hierarchical learning schema. The concatenated feature vector has many redundant and irrelevant features. Classical fuzzy ARTMAP architecture can not cope with the redundancy and irrelevance. On the other hand, proposed hierarchical schema is quite stable to the redundancy and irrelevance of the features.

In this set of experiments, feature-based AdaBoost achieves better performance than the descriptor-based AdaBoost. Feature-based AdaBoost reduces redundancy and irrelevance and achieves %88.91 and % 82.7 performances by decreasing the dimension size to 100. On the other hand, since the elimination of the individual features in a descriptor is not allowed, D-AdaBoost can only achieve 85.96% and 75.4% performance, on UMIST and ORL database, respectively.

5.3 Object Localization Experiments

In the first part of the object localization experiments, we compare two of the proposed methods, namely, Neighborhood Tree and Greedy-Based approach with the data set in [63]. The set contains 6 CD's from Corel Draw. Each CD contains 100 images with half of the data reserved for training and the rest for test. In the second part, we test the performance of the proposed image retrieval framework.

5.3.1 Comparison of Neighborhood Tree to Greedy-Based Approach

The goal of this experiment is to compare two object localization methods to each other. For this purpose, the images in the test data set are labeled and then queried by using both of the approaches. Since there are 50 images for each of the class in the test data set, precision is calculated retrieving 50 images for each category. Table 24 shows the comparison results of the methods according to precision values.

Table 24: Comparison of Precision Values of These Two Methods.

OBJECT	Precision (%) of the Neighborhood Tree	Precision (%) of the Greedy-Based Approach
Eagle	66	58
Elephant	60	56
Horse	74	68
Lion	62	58
Plane	74	66
Tiger	82	76
AVERAGE	69.67	63.67

Table 24 indicates that Neigborhood Tree outperforms the Greedy-Based Approach in object localization. One of the major advantages of the Neighborhood Tree over the Greedy-Based Approach is the utilization of all combinations of regions, which may represent the whole object, through the fuzzy membership values. (See Figure 26 and Figure 27) Greedy algorithm only merges the regions having exactly the same label

with the query object. Therefore, slight differences in the features vector may causes misclassification of the regions. It is possible to label a plane as eagle using the crisp features, in the labeling stage. Due to this fact, the greedy based labeling algorithm can stop merging the regions without obtaining the whole object.

For some of the objects, certain regions have significant effect on the localization. If that regions are misclassified by the labeling module, it is likely that Greedy algorithm cannot localize the whole objects. On the other hand, since Neighborhood Tree searches almost all the alternatives over the tree, it is more likely to localize the objects from the images. As a summary, Neighborhood Tree approach is more tolerant to noise and slight variations of the regions than the Greedy-based approach.

Figure 26 shows a sample labeling from the greedy-based approach, where the whole object could not be extracted from the image properly, because of the misclassified regions.

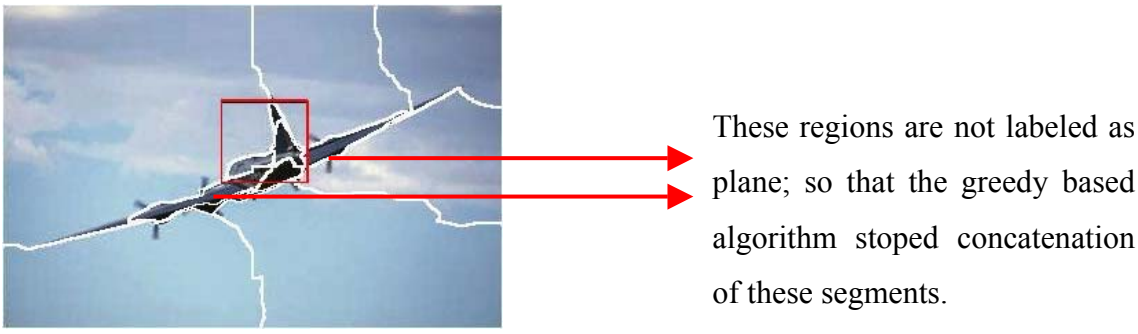


Figure 26: Sample Labeling of Greedy based Approach.

Figure 27 shows the output of a sample labeling from the Neighborhood Tree algorithm. The figure shows how the problem of merging the over-segmented regions of objects was solved with Neighborhood algorithm.

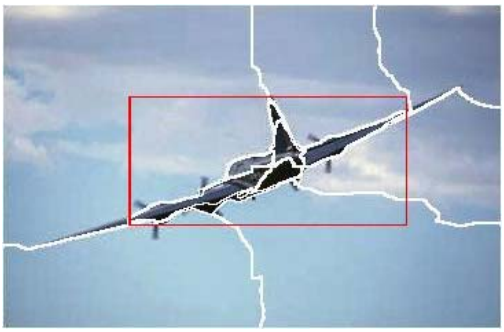


Figure 27: Sample Labeling of Neighborhood Tree.

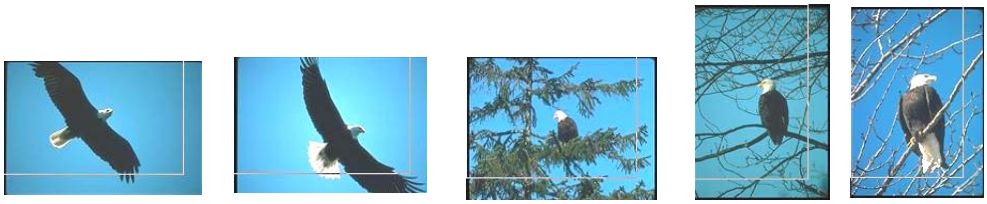
5.3.2 Performance of the Proposed Hierarchical Object Localization and Image Retrieval Framework

The goal of this experiment is to test the performance of the proposed image retrieval framework. For this purpose, the images in the test data set of [63] are labeled and then queried by the developed system. Note that hierarchical classification schema is employed in the training module, whereas neighborhood tree approach is used in the labeling module of the proposed image retrieval framework. Table 25 shows the performance of the proposed approach with the test data set. The numbers at the top of the table show the number of retrieved images in the querying phase.

Table 25 : Query Results of the Proposed Image Retrieval Framework. The Numbers at the Top Refers to the Number of Retrieved Images.

OBJECT	5	10	15	20	25	30	35	40	45	50
Eagle	5	10	15	19	20	23	26	29	31	33
Elephant	5	7	10	13	16	18	24	26	29	30
Horse	5	10	15	18	22	26	30	33	36	37
Lion	5	8	11	14	17	20	24	28	30	31
Plane	5	10	15	18	24	27	29	33	34	37
Tiger	5	10	15	20	24	27	30	34	38	41

Figure 28 shows the first 5 query outputs for the corresponding object.



Eagles

Elephants

Horses

Lions

Planes

Tigers

Figure 28: First 5 Query Results.

5.4 Chapter Summary

This chapter gives the experimental results of the developed object localization and image retrieval system. The proposed content-based image retrieval framework is tested over a subset of Corel Draw image database, UMIST and ORL face databases. The "Best Representative Features" are found in the training phase using Fuzzy ARTMAP, Feature-based AdaBoost, Descriptor-based AdaBoost, Best Representative Descriptor, majority voting and the hierarchical learning schema. The experimental results indicate that the proposed hierarchical learning schema yields better retrieval rates than the other methods. It is also observed that the proposed neighborhood tree approach achieves significantly better precision values than Greedy-Based approach in image retrieval based on object localization.

CHAPTER

6 SUMMARY, CONCLUSIONS AND FUTURE DIRECTIONS

Let us conclude the thesis by first, summarizing the proposed object-based image retrieval framework. Then, we discuss pros and cons, which lead us to further improvements in our future research study.

The major goal of this thesis is to develop an image retrieval framework for the predefined object classes. In order to reach this goal, first, we propose a hierarchical supervised schema to learn the object classes. Then, we develop an object localization method, which extracts the query objects from the images of the database.

The proposed image retrieval framework brings two major contributions, which improves the performance of the existing CBIR systems. Firstly, a dedicated feature subspace is extracted for each object class, which is called as `the best representative feature space`, rather than using the same fixed feature space for all of them. Secondly, object localization problem is solved by a search algorithm on a data structure, which represents the adjacent object regions in the Neighborhood Tree.

In order to find the Best Representative Feature space, we propose three new methods: First, a single descriptor is selected for each of the object class. Then, this method is extended to k-best representative descriptors by adapting the AdaBoost algorithm. Finally, a hierarchical learning schema, based on the Adaptive Resonance Theory, to find a discriminative feature set for each object class is introduced.

6.1 Discussion

The proposed layered architecture is inspired from the human visual system. The first layer combines classifiers with low level, low dimensional features, partially emulating the eye. The second layer implements Adaptive Resonance Theory, which extracts higher level information from the first layer and learns which recognition engine to use

for which training object class to what extent, as in the associative memory of human brain.

The experimental results indicate that the proposed layered architecture based on Adaptive Resonance Theory outperforms the popular methods available in the literature, for the data set given in Chapter 5. The existing problems about normalization, curse of dimensionality, feature space design, and semantic gap is partially resolved by the proposed system.

Rather than concatenating the descriptors, Hierarchical Learning Schema combines the results of independent classifiers at the meta-layer. This approach not only preserves the semantic information of each descriptor elaborating the semantic gap problem and also avoids the normalization and curse of dimensionality problems. The output of each classifier in the first layer gives the degree of importance of the input descriptor in recognizing each particular class. Therefore, if a descriptor is relatively more discriminative than the other descriptors in recognizing a certain class, then the sub-space, which represents this descriptor, is emphasized in the feature vector of the second layer using FAM architecture.

One of the important observations from the experiments is that the hierarchical learning schema can even learn the characteristics of training classes with a small number of training samples. Such a property is very important for CBIR systems. Most of the time, it is not possible to generate sufficient number of training samples for each class.

Another important property of the hierarchical learning schema is its robustness to the increase in the number of training classes. The performance remains relatively stable, when the number of training classes is increased. However, it should be noted that using k-NN as the base level recognition engine has a negative impact on the hierarchical architecture. It is well known that the performance of k-NN is known to decrease rapidly as the number of training classes is increased.

In this thesis, two approaches, namely, Greedy-Based and Neighborhood Tree, are proposed for object localization in a segmented image. Rather than merging the N-Cut segments in an image using a greedy algorithm, the Neighborhood Tree is adapted for object labeling. The systematic merging of the search algorithm applied on the

Neighborhood Tree facilitates extracting the query object from a set of over segmented regions. The algorithm limits the possible combinations of object pieces that are labeled by training algorithm based on the neighborhood information.

Neighborhood Tree Approach outperforms the Greedy-Based Approach for object localization. The reason of the increase in the performance is that the neighborhood tree approach is capable of searching more alternatives by utilizing a tree data structure. The Greedy Approach is a specialised case of the neighborhood tree, where only the regions having the same label with the query object with maximum membership values are used in construction of the tree. On the other hand, the major drawback of the neighborhood tree approach is the complexity introduced in construction of the tree.

6.2 Future Directions

In this study, for the sake of simplicity, we implemented a two-layered architecture for the proposed hierarchical learning schema. Extended to multi-layer by combining color, texture and shape descriptors separately in the upper layers, this architecture improves the classification performance of the proposed method.

The aforementioned architecture is implemented with k-NN as the base layer and fuzzy ARTMAP as the meta-layer recognition engines. Using different classifiers in base and meta layers such as Boost-MAP, MART is likely to increase the performance of the proposed system.

In our experiments, the descriptor-based and feature-based AdaBoost methods achieve nearly the same performance. We believe that if the number of descriptors is more and the performance of each descriptor is better, descriptor-based AdaBoost gives even more precise results.

We believe that shape descriptor should be emphasized to increase the classification performances. The human beings give more importance to shape recognizing objects. We can even recognize objects from gray scale photographs, which have no color and little texture information. Therefore, adding more shape features to the system is likely to increase its success.

The output of the labeling algorithm provides us a very convenient infrastructure to construct a fuzzy database, since the labeling module gives the membership values of each object being in each image. The proposed system can be extended to perform fuzzy queries by employing a fuzzy object-oriented database modeling such as FOOD proposed in [81].

Finally, the proposed system uses Neighborhood Trees for object localization over segmented images. Other algorithms will be proposed to increase the object localization performance.

REFERENCES

[1] Gail A. Carpenter, Stephen Grossberg, Natalya Markuzon, John H. Reynolds and David B. Rosen, Fuzzy ARTMAP: A Neural Network Architecture for Incremental Supervised Learning of Analog Multidimensional Maps, IEEE Transactions on Neural Networks, Vol. 3, No: 5, pp. 698-713, September 1992.

[2] Yoav Freund, Robert E. Schapire, A Short Introduction to Boosting, Journal of Japanese Society for Artificial Intelligence 1999, pp 771-780.

[3] M. Uysal, F. Y. Vural, Selection of the Best Representative Feature and Membership Assignment For Content-Based Fuzzy Image Database, pp 141-151, CIVR-2003.

[4] Daniel B Graham and Nigel M Allinson, Characterizing Virtual Eigensignatures for General Purpose Face Recognition, Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences, Vol. 163, pp 446-456, 1998

[5] O. Trier and A. Jain and T. Taxt, "Feature extraction methods for character recognition A survey", Pattern Recognition 29, pp. 641-662, 1996.

[6] B.S. Manjunath, P. Wu, S. Newsam, H.D. Shin, A texture descriptor for browsing and similarity retrieval, Signal Processing: Image Communication, 2000

[7] L. Cieplinski, W. Kim, J.-R. Ohm, M. Pickering, and A.Yamada, MPEG-7 Visual part of eXperimentation Model Version 12.0,12/06/2001

[8] Naber, Gregory L., The Geometry of Minkowski Spacetime, Springer-Verlag, New York, 1992. ISBN 0-387-97848-8

[9] W. Krzanowski, Principles of Multivariate Analysis, Oxford Science Publications, Oxford, 1988. p.233

[10] Michalski, Ryszard S., Robert E. Stepp, and Edwin Diday, A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts. Progress in Pattern Recognition, Vol. 1, 1981, New York: North-Holland, pp. 33-56.

[11] Batchelor, Bruce G., Pattern Recognition: Ideas in Practice. New York: Plenum Press, 1978, pp. 71-72.

[12] Diday, Edwin, Recent Progress in Distance and Similarity Measures in Pattern Recognition. Second International Joint Conference on Pattern Recognition 1974, pp. 534-539.

[13] D. Randall Wilson, Tony R. Martinez, Improved Heterogeneous Distance Functions, Journal of Artificial Intelligence Research 6 (1997) 1-34

[14] J. C. Gower, "Euclidean distance geometry," The mathematical scientist, 1982.

[15] Duygulu P., Barnard K., N de Fretias, Forsyth D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In Proceedings of the European Conference on Computer Vision, (2002) 97-112

[16] Carson C., Thomas M., Belongie S.,Hellerstein J.M., Malik J.: Blobworld: A System for Region-Based Image Indexing and Retrieval. Proc. Visual Information Systems, (1999) 1355-1360.

[17] Wang J.Z., Li J., Wiederhold G.: SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries. IEEE Trans. Pattern Anal. Machine Intell., vol. 23, no. 9. (2001) 947-963.

[18] Y. Rui, T. S. Huang, and S. F. Chang, "Image retrieval: Current techniques, promising directions and open issues," J. Vis. Commun. Image Represent., vol. 10, pp. 39--62, 1999.

[19] Pattern Classification. (2nd ed.) by Richard O. Duda,. Peter E. Hart. and David G. Stork. Wiley Interscience. 680 pages ISBN: 0-471-05669-3

[20] Cor J. Veenman and David M.J. Tax, LESS: A Model-Based Classifier for Sparse Subspaces, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 27, NO. 9, pp. 1496-1500, SEPTEMBER 2005.

[21] Mao K. Z., "Fast Orthogonal Forward Selection Algorithm for Feature Subset Selection", IEEE Transactions on Neural Networks, Vol. 13, No. 5, pp. 1218-1224, September 2002.

[22] P.M. Narendra and K. Fukunaga. A branch and bound algorithm for feature subset selection. IEEE Trans. on Computers, 26:917-922, September 1977.

[23] Peng H., Long F., Ding C.,"Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max Relevance, and Min-Redundancy", IEEE Transactions on Pattern Analysis And Machine Intelligence, Vol. 27, No:8, pp. 1226-1238, August 2005.

[24] D. Whitley, "A genetic algorithm tutorial," Statistics and Computing, vol. 4, pp. 65--85, 1994.

[25] C. Saunders, M.O. Stitson, J. Weston, L. Bottou, B. Schlkopf, and A. Smola, Support Vector Machine Reference Manual, Technical Report, CSD-TR-98-03, Royal Holloway, Univ. of London, Egham, UK, Mar. 1998

[26] Kohavi, R., John, G., Wrappers for Feature Subset Selection, Artificial Intelligence, Vol. 97 (1-2) (1997) 273-324.

[27] Daniel Monegatto Santoro, Maria do Carmo Nicoletti, Investigating a Wrapper Approach for Selecting Features Using Constructive Neural Networks, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'05)

[28] I. Ruthven and M. Lalmas, A survey on the use of relevance feedback for information access systems, Knowledge Engineering Review. 18 (2). 2003. pp. 95-145.

[29] I. Ruthven, M. Lalmas, and C. J. van Rijsbergen, Incorporating user search behaviour into relevance feedback, Journal of the American Society for Information Science and Technology. 54 (6). 2003. pp. 528-548.

[30] Salton, G. & Buckley, C., Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, 1990, 41(4), 288-297.

[31] Yixin Chen, James Z. Wang and Robert Krovetz, CLUE: Cluster-based Retrieval of Images by Unsupervised Learning,' IEEE Transactions on Image Processing, vol. 14, no. 8, pp. 1187-1201, 2005

[32] Breiman, L., Bagging predictors, Machine Learning 24,1996, pp: 123-140.

[33] J. Kittler, M. Hatef, R.P.W. Duin and J. Matas, "*On Combining Classifiers*", IEEE Trans. Pattern Analysis and Machine Intell., Vol. 20, No. 3, 1998, pp. 226-239

[34] Grossberg, Stephen (1976a), "Adaptive Pattern Classification and Universal Recoding: 1. Parallel Development and Coding of Neural Feature Detectors", Biological Cybernetics 23, pp.121-134, Reprinted in Anderson & and Rosenfeld, 1988.

[35] C. Papageorgiou and T. Poggio, A trainable system for object detection. *Intl. J. Computer Vision*, 38(1):15–33, 2000

[36] Li J., Wang J.Z.: Automatic Linguistic Indexing of Pictures By a Statistical Modeling Approach. IEEE Trans. Pattern Anal. Machine Intell., vol. 25, no 9, pp. 1075-1088, 2003.

[37] M. M. Gorkani and R. W. Picard. Texture orientation for sorting photos 'at a glance'. In *Proc. 12th Int. Conf. on Pattern Recognition*, pages:459-464, 1994.

[38] J. Z. Wang, J. Li, and G. Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947- 963, 2001

[39] K. Murphy, A. Torralba, and W. Freeman. Using the forest to see the trees: a graphical model relating features, objects, and scenes. In *Advances in Neural Information Processing Systems 16*. Cambridge, MA:MIT Press, 2004

[40] W.Y. Ma and B.S. Manjunath, NeTra: A Toolbox for Navigating Large Image Databases, Proc. IEEE Int. Conf. Image Processing, pp. 568-571, 1997.

[41] J.R. Smith and C.S. Li, Image Classification and Querying Using Composite Region Templates, J. Computational Vision and Image Understanding, vol. 75, no. 1-2, pp. 165-174, 1999

[42] J. Li, J.Z. Wang, and G. Wiederhold, IRM: Integrated Region Matching for Image Retrieval," Proc. 8th ACM Int. Conference on Multimedia, pp. 147-156, October 2000.

[43] E. Rivlin, S.J. Dickinson, and A. Rosenfeld. Recognition by functional parts. Computer Vision and Image Understanding, 62(2):164{176, September 1995

[44] R. Fergus, P. Perona, A. Zisserman,Object Class Recognition by Unsupervised Scale-Invariant Learning, In: Proc. CVPR. Volume II. (2003), pp. 264-272.

[45] D.H. Wolpert, "Stacked Generalization," Neural Networks, vol. 5, no. 2, pp. 241-260, 1992.

[46] C.M. Friedrich. Ensembles of evolutionary created artificial neural networks and nearest neighbour classifiers. In Proc. 3rd On-line Conference on Soft Computing in Engineering Design and Manufacturing (WSC3), pages 288 298, 1998.

[47] G. Giacinto and F. Roli. Design of effective neural network ensembles for image classification processes. Image Vision and Computing Journal, 19(9-10):699-707, 2001.

[48] J. Franke and E. Mandler, "A Comparison of Two Approaches for Combining the Votes of Cooperating Classifiers," Proc. 11th IAPR Int'l Conf. Pattern Recognition, Conf. B: Pattern Recognition Methodology and Systems, vol. 2, pp. 611-614, 1992.

[49] G. Rogova, "Combining the Results of Several Neural Network Classifiers," *Neural Networks*, vol. 7, no. 5, pp. 777-781, 1994.

[50] A. Krogh and J. Vedelsby, "Neural Network Ensembles, Cross Validation, and Active Learning," *Advances in Neural Information Processing Systems 7*, Cambridge, Mass.: MIT Press, 1995.

[51] T.S. Huang and C.Y. Suen, "Combination of Multiple Experts for the Recognition of Unconstrained Handwritten Numerals," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, no. 1, pp. 90-94, Jan. 1995.

[52] L.K. Hansen and P. Salamon, "Neural Network Ensembles," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993-1,001, Oct. 1990.

[53] Feng Jing, Bo Zhang, Fuzong Lin, Wei-Ying Ma, Hong-Jiang Zhang, A novel region-based image retrieval method using relevance feedback, International Multimedia Conference Proceedings of the 2001, ACM workshops on Multimedia.

[54] Ingemar J. Cox and Matthew L. Miller and Thomas P. Minka and Peter N. Yianilos, An Optimized Interaction Strategy for Bayesian Relevance Feedback , CVPR'98

[55] Tian Q., Hong, P., Huang, T. S., (2000) Update relevant image weights for content-based image retrieval using support vector machines, Proc. IEEE Int. Conf. On Multimedia and Expo, Vol. 2, pp. 1199-1202.

[56] Ye Lu, Chunhui Hu, Xinquan Zhu, HongJiang Zhang, Qiang Yang, "A Unified Semantics and Feature Based Image Retrieval Technique Using Relevance Feedback", ACM MULTIMEDIA 2000--

The 8th ACM International Multimedia Conference, Los Angeles, California , October 30 - November 3, 2000

[57] Christophe Meilhac and Chahab Nastar, Relevance feedback and category search in image databases. In Proceedings of IEEE Intemational Conference on Multimedia Computing and Systems, pages 512-517. IEEE Computer Society, June 1999.

[58] Kriengkrai Porkaew and Kaushik Chakrabarti. Query refinement for multimedia similarity retrieval in mars. In Proceedings of the seventh ACM international conference on Multimedia (Part 1), pages 235-238.ACM Press, 1999.

[59] Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: a power tool in interactive content-based image retrieval. IEEE Transactions on Circuits and Systems for Video Technology, 8(5):644-655, 1998.

[60] A. P. Dempster, N. M. Laird and D. B. Rubin. Maximum Likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society, Series B , 1(39):1-38,1997

[61] Y Mori, H Takahashi, and R Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In Proceedings of the International Workshop on Multimedia Intelligent Storage and Retrieval Management, 1999.

[62] D Blei and M Jordan, Modeling annotated data. In Proceedings of the ACM SIGIR Conference on Research and Development in Informaion Retrieval, pages 127-134, 2003.

[63] Pınar Duygulu-Şahin, P.H.D. Thesis, Translating Images to Words: A Novel Approach for Object Recognition, METU, February 2003.

[64] W. Niblack, R. Barber, W. Equitz, M. Flicker, E. Glasman, D. Petkovic, P. Yanker, C. Flaoutsos and G. Taubin. The QBIC Project: Querying Images By Content Using Color Texture and Shape, In Procedings of The SPIE Conference on Storage and Retrieaval for Image and Video Databases, 1908:173-187, 1993.

[65] A. Pentland, R. W. Picard, and S. Sclaroff. Photobook: A content -based manipulation of image databases. International Joural on Computer Vision, 18 (3): 233-254,1996.

[66] Sougata Mukherjea, Kyoji Hirata, and Yohinori Hara. Towards a multimedia world wide web information retrieval engine. In Sixth International WWW Conference, 7-11 April, CA, USA 1997

[67] J. R. Smith and S.F. Chang. Querying With color regions using the VisualSEEk content-based visual query system. In M. T. Maybury, editor, Intelligent Multimedia Information Retrieval. AAAI Press, 1997.

[68] J. R. Smith. Integrated Spatial and Feature Image Systems: Retrieval, Compression and Analysis. Phd. Thesis, Graduate School of Arts and Sciences, Colombia University, 1997.

[69] Wei-Ying Ma and B. S. Manjunath. NeTra: A toolbox for navigating large image databases, Multimedia Systems, 7(3):184-198,1999.

[70] J.R. Smith and C.S. Li, ™Image Classification and Querying Using Composite Region Templates, Int. J. Computer Vision and Image Understanding, vol. 75, no. 1-2, pp. 165-174, 1999.

[71] A. Natsev, R. Rastogi, and K. Shim. WALRUS: a similarity retrieval algorithm for image databases. In Proceedings of the ACM SIGMOD *International Conference on Management of Data*, volume 28(2) of *SIGMOD Record (ACM Special Interest Group on Management of Data)*, p. 395–406, Philadelphia, PA, USA, June 1–3, 1999, 1999.

[72] J Jeon, V Lavrenko, and R Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In Proceedings of the ACM SIGIR Conference on Research and Development in Infrmation Retrieval, pages 119-126, 2003.

[73] G. A. Carpenter and S. Grossberg. Pattern Recognition by Self-Organizing Neural Networks, Chapter 10. Cambridge, MA, MIT Press, 1994.

[74] Robert E. Schapire, The Boosting Approach to Machine Learning An Overview, MSRI Workshop on Nonlinear Estimation *and Classification, 2002.*

[75] Yoav Freund. Boosting a weak learning algorithm by majority. *Information and Computation*, 121(2):256–285, 1995.

[76] Jianbo Shi, Jitendra Malik, Normalized Cuts and Image Segmentation, IEEE Transactions and Pattern Analysis and Machine Intelligence, 22(8):888--905, 2000.

[77] James Ze Wang, Gio Wiederhold, Oscar Firschein, Sha Xin Wei, Content-based image indexing and searching using Daubechies' wavelets, Int J Digit Libr (1997) 1: 311-328.

[78] Gail A Carpenter, Stephen Grossberg and Peggy Israel Doerschuk, Handbook of Neural Computation release 1, 1997, IOP Publishing Ltd and Oxford University Press.

[79] Anil K. Jain, Robert P.W. Duin, and Jianchang Mao, Statistical Pattern Recognition: A Review, IEEE Transactions On Pattern Analysis And Machine Intelligence, VOL. 22, NO. 1, pp. 4-37, JANUARY 2000.

[80] Gail A. Carpenter, Stephen Grossberg, Adaptive Resonance Theory, The Handbook of Brain Theory and Neural Networks, Second Edition,2002, Cambridge, Massachusetts: MIT Press.

[81] A. Yazici and R. George, Fuzzy Database Modeling, Physica-Verlag Press, New York, 1999.

[82] Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio, Member, IEEE, Example-Based Object Detection in Images by Components, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 23, NO. 4, pp. 349-361, APRIL 2001

[83] Daniel B Graham and Nigel M Allinson, Characterizing Virtual Eigensignatures for General Purpose Face Recognition, Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences, Vol. 163, pp 446-456, 1998

[84] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification", 2nd IEEE Workshop on Applications of Computer Vision, December 1994, Florida.

[85] Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang, Face Recognition Using Laplacianfaces, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 27, NO. 3, pp. 328-340, MARCH 2005

[86] Peter N. Belhumeur, Joao P. Hespanha, and David J. Kriegman, Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 19, NO. 7, pp. 711-720, JULY 1997.

VITA

Mutlu Uysal was born in Aydın on April 10, 1974. He graduated from Bilkent University Computer Engineering and Information Science Department in 1997. He received his M.S degree in Computer Engineering from the Middle East Technical University in 1999. He worked as a teaching assistant in METU Computer Engineering Department from 1997 to 1999. Since then he has been working in one of the Turkish Defence Industry Companies namely S.T.M., for six years. His main areas of interest are multimedia database management systems, pattern recognition and image retrieval.

Publications

- Uysal M., Vural F.Y., A Content-Based Fuzzy Image Database Based on the Fuzzy ARTMAP Architecture, Turk. Journal Elec. Engin.,13,(2005),333-342.

- Uysal M., Vural F.Y., ORF-NT: An Object-based Image Retrieval Framework Using Neighborhood Trees, ISCIS 2005, İstanbul.

- Uysal M., Vural F.Y., A Content Based Image Retrieval System Based Fuzzy ARTMAP Architecture, ACM 2004, New York

- Uysal M., Ö. Özcanlı , Vural F.Y., Bulanık ARTMAP Mimarisini Kullanan İçerik Bazlı Bir İmge Sorgulama Sistemi, SIU 2004, İstanbul.

- Uysal M., Vural F.Y., En İyi Temsil Eden Öznitelik Kullanılarak İçeriğe Dayalı İndeksleme ve Bullanık Mantığa Dayalı Sorgulama Sistemi, SIU 2003, Kuşadası.

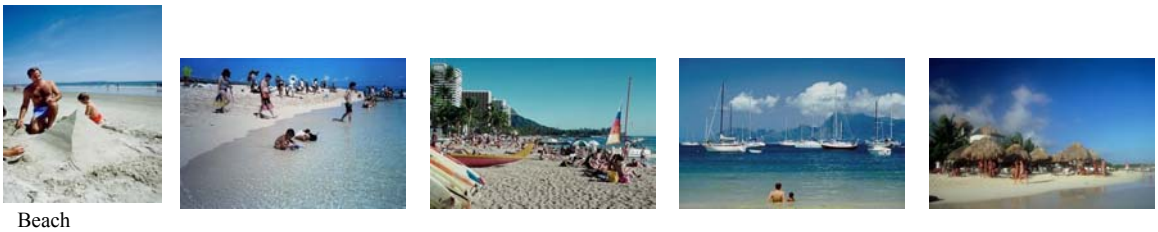
- Uysal M., Vural F.Y., Selection of The Best Representative Feature And Membership Assignment For Content-Based Fuzzy Image Database, CIVR-2003, Illinois.

- Uysal M., Vural F.Y., Tek Boyutlu Renk Aralıkları İle Çalışan Hızlı Bir Renk İndirgeme Yöntemi, SIU-98,6. Sinyal İşleme ve Uygulamaları Kurultayı, Kızılcahamam, Ankara,1998.

- Uysal M., Vural F.Y., A Fast Color Quantization Algorithm By Using One-Dimensional Color Intervals, ICIP-98, International Conference on Image Processing, Chicago, 1998.

Appendix A

Africa



Beach

Buildings

Buses

Dinesours

Figure 29: Example Images Used In ALIP Experiments.

Elephant

Flowers

Horses

Mountain

Food

Figure 29: Example Images Used In ALIP Experiments. (Cont.)