AN IMAGE RETRIEVAL SYSTEM BASED ON REGION CLASSIFICATION

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

BY

ÖZGE CAN ÖZCANLI - ÖZBAY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

COMPUTER ENGINEERING

JUNE 2004

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan Özgen
Director

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

Prof. Dr. 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 Master of Science.

Prof. Dr. Fatoş Yarman - Vural
Supervisor

Examining Committee Members

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

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

Assoc. Prof. Dr. Sibel Tarı (METU, CENG) _____

Assist. Prof. Dr. Pınar Duygulu - Şahin (Bilkent U, CS) _____

Mutlu Uysal, MSc (Teknokent) _____

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: Özge Can Özcanlı Özbay

Signature:

ABSTRACT

AN IMAGE RETRIEVAL SYSTEM BASED ON REGION CLASSIFICATION

Özcanli - Özbay, Özge Can

M.S., Department of Computer Engineering

Supervisor: Prof. Dr. Fatoş Yarman - Vural

June 2004, 60 pages

In this thesis, a Content Based Image Retrieval (CBIR) system to query the objects in an image database is proposed. Images are represented as collections of regions after being segmented with Normalized Cuts algorithm. MPEG-7 content descriptors are used to encode regions in a 239-dimensional feature space. User of the proposed CBIR system decides which objects to query and labels exemplar regions to train the system using a graphical interface. Fuzzy ARTMAP algorithm is used to learn the mapping between feature vectors and binary coded class identification numbers. Preliminary recognition experiments prove the power of fuzzy ARTMAP as a region classifier. After training, features of all regions in the database are extracted and classified. Simple index files enabling fast access to all regions from a given class are prepared to be used in the querying phase. To retrieve images containing a particular object, user opens an image and selects a query region together with a label in the graphical interface of our system. Then the system ranks all regions in the indexed set of the query class with respect to their L_2 (Euclidean) distance to the query region and displays resulting images. During retrieval experiments, comparable class precisions with respect to exhaustive searching of the database are maintained which demonstrates effectiveness of the classifier in narrowing down the search space.

Keywords: Content Based Image Retrieval, Region Labeling, Region Classification, fuzzy ARTMAP, MPEG-7.

ÖZ

BÖLGE SINIFLANDIRMASINA DAYALI BİR GÖRÜNTÜ ERİŞİM SİSTEMİ

Özcanli - Özbay, Özge Can

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Fatoş Yarman - Vural

Haziran 2004, 60 sayfa

Bu çalışmada, veritabanlarında nesne içeriği araması yapabilecek bir görüntü sorgulama sistemi önerilmektedir. Görüntüler Düzgelenmiş Kesikler (Normalized Cuts) algoritması ile bölütlenmiş bir bölge kümesi şeklinde temsil edilmektedir. En alt seviyede ise bölgeler, MPEG-7 standardı ile oluşturulan, 239 elemanlı bir öznitelik vektörü ile betimlenmektedir. Kullanıcı hangi nesnelerin sorgulanacağına karar vererek, geliştirilen bir arayüz aracılığı ile örnek bölgeler etiketlemekte ve sistemin eğitimi için kullanılacak kümeleri oluşturmaktadır. Sistem, bölütlerin sınıflandırılmasında bulanık ARTMAP sinir ağı mimarisini kullanmaktadır. Yapılan ön deneyler bu mimarinin erişim sistemlerinde kullanım için uygunluğunu ve bölge sınıflamasındaki başarısını kanıtlamıştır. Eğitim aşamasından sonra, bölütlenen veri tabanındaki tüm bölgelerden öznitelik vektörleri çıkartılarak, bu vektörler eğitilen bulanık ARTMAP modülü tarafından sınıflandırılır. Sorgu aşamasında erişimi hızlandırmak amacıyla her sınıfın bölge numaralarını kapsayan dizinler oluşturulur. Kullanıcı sistem arayüzü sayesinde bir görüntü açarak, sorguda kullanılacak bölgeyi ve etiketi belirler. Sorgu etiketine sahip sınıf bölgelerinin öznitelik vektörleri ile sorgu bölgesinin öznitelik vektörü arasındaki L_2 (Öklit) uzaklıkları hesaplanır ve sonuç görüntüleri bu uzaklığa göre sıralanarak kullanıcıya gösterilir. Bir sınıfın tüm görüntüleri kullanılarak yapılan erişim deneyleri sonucunda önerilen sistemin arama uzayını daraltmakta başarılı olduğu

tüm veritabanının sınıflandırma kullanılmadan aranması ile alınan sonuçlar göz önüne alındığında görülmektedir.

Anahtar Kelimeler: İçeriğe dayalı görüntü erişimi, bölge etiketleme, bölge sınıflama, bulanık ARTMAP, MPEG-7.

ACKNOWLEDGMENTS

It is a great pleasure for me to complete my Master thesis and have this opportunity to thank all who were "there" for me during this endeavor.

First of all, I would like to thank my advisor Fatoş Yarman - Vural who have been my idol since my first day in Middle East Technical University. I have become a computer engineer as the result of her sincere speech to a group of high school students as the Head of Dept. of Computer Engineering in METU in 1998. Her philosophy and relentless energy, her endurance to live a principled life as a member of academic community continue to enlight my whole life since then. It is such a big honor for me to be a student of hers.

I am also thankful to the wonderful faculty and staff at the Dept. of Computer Engineering, METU, for their endless support during my training. It was also a great pleasure for me to be a member of this crew as a teaching assistant. All the members of Image Processing and Pattern Recognition Laboratory at METU: Prof. Volkan Atalay, Prof. Sibel Tarı, Pınar Duygulu, Nafiz Arıca, Mutlu Uysal, Aykut Erdem, Erkut Erdem, Gülşah Tümüklü, Turan Yüksel and Ömer Önder Tola thanks for all those discussions and being patiently helpful at all stages. I should not forget to thank Erol Şahin for introducing me to fuzzy ARTMAP architecture which is an important part of this thesis.

My dear friends, Seda, Çiğdem, Ozan, Sadettin, Duru and the rest of CENG 2002, you made my life in METU so special and unforgettable.

Lastly, how can one forget the most important people of her life: the best father and captain Süleyman Özcanlı, the best mother and a butterfly İlknur Özcanlı, my great brother and friend İlke and my love Serhat. Thank you all my wonderful family.

To my captain and his butterfly.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Recent advances of the technology in digital imaging, broadband networking and digital storage devices make it possible to easily generate, transmit, manipulate and store large numbers of digital images and documents. As a result, image databases have become widespread in many areas such as art gallery and museum management, architectural and engineering design, interior design, remote sensing and management of earth resources, geographic information systems, medical imaging, scientific database management systems, weather forecasting, fabric and fashion design, trademark and copyright database management, law enforcement, criminal investigation, picture archiving and communication systems. Furthermore, the rapid growth of the World Wide Web has led to the formation of a very large but disorganized, publicly available image collection. Recent studies show that there are 180 million digital images on publicly indexable Web and millions of new images are being produced every day [41]. Thus, efficient image retrieval from digital image collections has been of great interest over the last decade and several systems have been developed for research and commercial purposes. (see: [37, 41, 44, 69, 88] for recent surveys.)

Retrieval problem is to select from a collection those images that are relevant to the user request specified either in visual or textual form. In this thesis, we are interested in using image content, i.e. only visual information, as a way of accessing the database. Despite the apparent success in appearance based querying, available systems are still far from retrieving images containing particular objects. Studies [7, 33, 34] reveal the need to query in the "things" domain, which suggests integration of object recognition

techniques with current retrieval architectures. However, object recognition is possible via the utilization of certain representation schemes, which usually require segmentation of the object from the rest of the image. Current automatic image segmentation methods are far from extracting objects. However, segmented regions coarsely correspond to objects or parts of objects in a natural image. In this regard, segmentation of images into coherent regions and representing them as collections of regions is a strong candidate as an intermediate level representation [22, 90]. A learning system that can efficiently discriminate objects under this representation is highly desirable. In this thesis, fuzzy ARTMAP algorithm is investigated as a promising tool and a system that can query objects in an inexactly segmented database is proposed.

Fuzzy ARTMAP is a supervised learning algorithm that can rapidly self-organize stable categorical mappings between input and output vectors [16, 19]. One of the major characteristics of fuzzy ARTMAP is that it can learn to attend to different parts of the feature vector to classify each category. This is an implicit feature selection, a major problem in CBIR applications, and is usually handled by automatic (via relevance feedback) or manual weight adjustment. Also, fuzzy ARTMAP has a many-to-one mapping capability, i.e. it can be trained to map regions that are dissimilar according to their features to the same class. These superiorities reveal that fuzzy ARTMAP can be successful at extracting complex objects in a database with a large variety of images. Fuzzy ARTMAP has been used in many applications of computer vision such as medical diagnosis [16], medical database analysis [42], land cover classification via remote sensing [17] and satellite imagery recognition [67]. In this study, fuzzy ARTMAP is used as the matching engine of our CBIR architecture.

CBIR applications transform pixel space to a feature space for efficient indexing and searching. Thus, extraction of *low-level* visual features from images has been an important aspect of retrieval research and led to the proposal of many schemes (see: [88] for a good summary of feature extraction schemes used by a variety of systems). Recently, this field of research is getting stabilized with the standardization of successful visual content descriptors under the name MPEG-7 from MPEG community [2, 4]. Our system uses these descriptors to extract color, texture and simple shape features of image regions and, at the lowest level, represents each region with a 239 dimensional feature vector.

It is relatively easy to match appearance of images automatically using low-level features extracted. Since this approach handles queries similar to "get me the images with high content of red at the middle", available quantitative features are well suited for the task. However, it is difficult to cope with high level semantic queries such as "the Governor X, kissing a baby". Such queries are to be handled with more sophisticated methods to extract semantic content.

The main contribution of this thesis is to extract qualitative information from the low-level quantitative representation in a hope to satisfy object level querying needs. For this purpose, a learning framework based on fuzzy ARTMAP algorithm is integrated into the CBIR architecture, where the user interferes with the system as a trainer. He/she determines relevant object classes to be learned according to querying needs and presents exemplars via a graphical interface. In this way, tedious training set preparation task becomes an easier region labeling stage before querying. User presence is an advantage of CBIR applications yet to be better utilized though there are systems with promising efforts such as getting relevance feedbacks from the user. In this perspective, our system is user-oriented and easily configurable according to specific application needs or the content of the database to be searched.

Another crucial aspect of CBIR is indexing. Various methods have been used [37, 69] for fast access to databases which typically contain thousands of images. However, it is well-known that conventional indexing schemes become to be infeasible for feature vectors with dimensions higher than twenty since the overhead of indexing operations exceeds the cost of an exhaustive search of the database. Our system proposes an efficient indexing via classification of regions and narrows down the search space without sacrificing precision in the retrieval stage.

In Chapter 2, representative CBIR systems will be briefly reviewed with emphasis on systems with learning capabilities that can be considered to be good efforts towards semantic querying. In Chapter 3, architectural details of our retrieval system will be summarized and fuzzy ARTMAP algorithm will be introduced. In Chapter 4, results of our preliminary experiments for region classification will be given. Retrieval performance of our system will also be presented in this chapter and the results are compared against exhaustive searching of the database. Lastly, Chapter 5 will conclude the thesis.

CHAPTER 2

BACKGROUND ON RELATED CBIR SYSTEMS

Since 1990's, Content Based Image Retrieval (CBIR) has become a very active research area and the literature is broad to review. The available systems can be basically classified as special and general purpose systems. The special purpose systems include: Xenomania from the University of Michigan [10] for face image retrieval based on query by example; Trademark image database system [45] for trademark retrieval using shape information; and Center of Excellence for Document Analysis and Recognition (CEDAR) system [83] from SUNY Buffalo for indexing and retrieving documents and newspaper articles by captions. Some of the general purpose systems are QBIC (IBM) [35], Photobook [65] and FourEyes [57] (MIT), VisualSEEK [80] and WebSEEK [79] (Columbia University), Chabot [62] and Blobworld [22] (UC Berkeley), Illustra/Virage [9], MARS [73] (University of Illinois at Urbana-Champaign), RetrievalWare from Excalibur Technologies [29], and Netra [53] (UC Santa Barbara).

These systems differ from each other in three respects [69]:

1. Visual feature extraction,
2. Indexing,
3. Retrieval system design.

Visual feature extraction is a necessary step to achieve efficient indexing and search with substantially reduced dimensions compared to raw pixel domain. Indexing is a usually by-passed aspect of retrieval systems though constituting the most critique

factor to increase retrieval performance in very large collections, such as WWW. Furthermore, it can be correlated with similarity measure and retrieval algorithms of the system to improve the efficiency. System design is about finding the optimum combinations of feature extraction and/or indexing methods coupled with the decisions regarding user interaction at various stages of the process, such as query preparation (e.g. sketch based or example based querying) or query refinement (e.g. giving relevance feedback). Most of the available systems basically achieve *appearance matching* which takes the global visual features of images into account without reference to the semantic content of images [37]. Appearance is particularly helpful when the composition of the image is important. For example, one could search for stock photos using a combination of appearance cues and keywords. Current state-of-the-art CBIR systems give satisfactory results for such purposes. Figure 2.1 shows a sample query result by a CBIR system making appearance matching. However, the problem is that images with the right composition but the wrong semantics are often retrieved. At this point, most of the systems with relevance feedback methods aim at excluding such images via parameter adjusting to satisfy the user. Unfortunately, an optimum parameter setting to catch the desired semantics may not always exist.

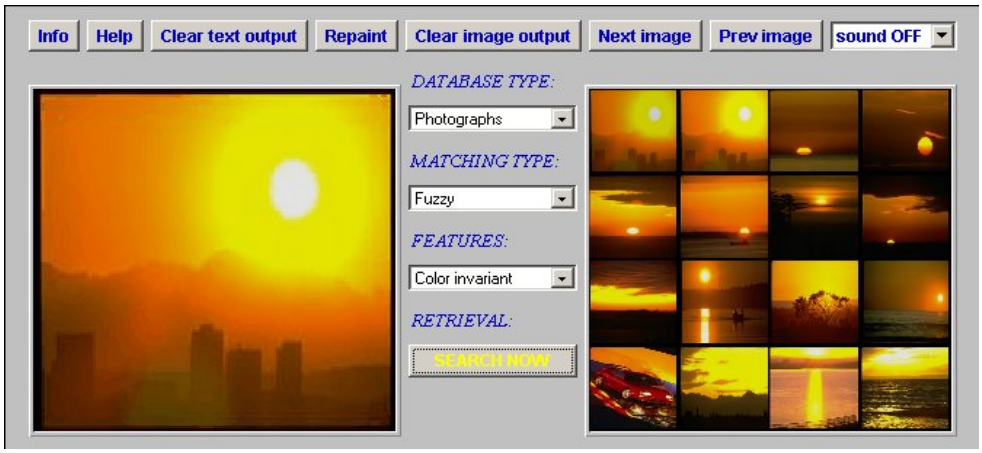

Figure 2.1: A sample appearance matching query from PicToSeek [40] system. The query image is on the left, and the images on the right are the results with global color feature matching.

An important step towards querying at object level is via localization of the global techniques and the introduction of region-based querying and/or indexing. The crucial question is how to form the regions. Malki *et al.* [54] propose a multiresolution

quadtree representation of the images, by-passing the difficult segmentation step to query the database of subimages formed at multi levels. This is the simplest approach for localization other then representing the image as a fixed sized grid, i.e. single level of resolution. Various segmentation-based approaches propose a better way of localization. These systems differ from each other at the choice of automatic segmentation algorithms, feature extraction schemes, and the matching algorithms used in retrieval. Segmentation is imperfect for the images in the unrestricted domain of CBIR. Some heuristics are proposed to overcome this drawback for different systems. Available systems either retrieve regions directly or they have a certain global matching algorithm utilizing region-wise similarity to retrieve images. Figure 2.2 shows a sample query result by a CBIR system making region-based retrieval. Some representatives

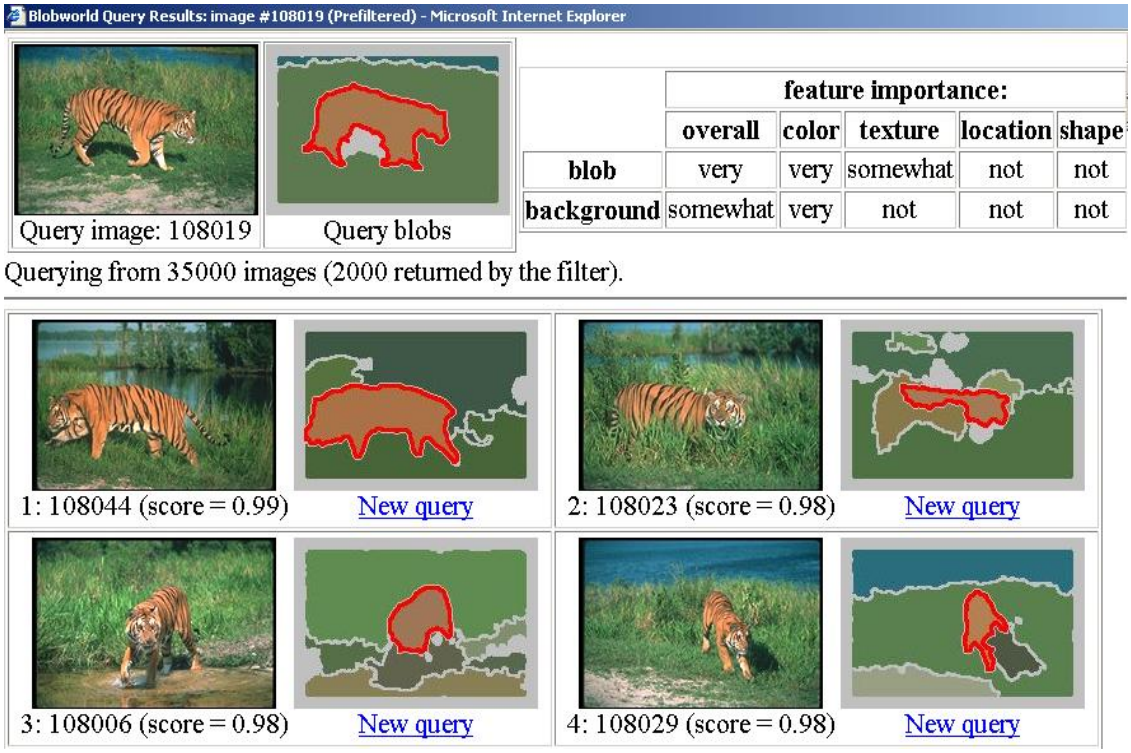

Figure 2.2: An example region-based query from Blobworld [22] system. Highlighted region is selected as the query region, and the system displays resultant images with matching regions highlighted similarly.

of region-based querying systems are given below:

- **Blobworld :** Blobworld [22], which is developed at UC Berkeley, is a system for image retrieval to find coherent image regions which roughly correspond

to objects. Each image is automatically segmented into regions (blobs) with associated color and texture descriptors using a statistical method based on Expectation-Maximization algorithm. Query is based on the attributes of one or two regions of interest, rather than a description of the entire image. The system allows the user to view the internal representation of the images to gain insight on why some "nonsimilar" images are returned and modify the query accordingly. A weighted distance scheme is used to combine different features to measure similarity. Figure 2.2 shows a sample query with results from this system. The on-line demo can be found at
`http://elib.cs.berkeley.edu/vision.html`.

- **Netra :** Netra is a prototype image retrieval system, developed at the UC Santa Barbara [28, 53]. Color, texture, shape and spatial location information of segmented image regions are used to search and retrieve similar regions from the database. It allows the user to compose queries like "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 one-third of the image" where the individual objects A, B and C could be regions belonging to different images. The on-line demo is at
`http://maya.ece.ucsb.edu/Netra`.

- **Ikona :** Ikona [13] is a prototype software for the IMEDIA (Image and multimedia indexing, browsing and retrieval) project, developed at INRIA. Ikona uses the query-by-example approach for retrieving images and integrates advanced features such as image signature combination and face detection. It supplies hybrid text-image retrieval mode and query refinement with relevance feedback, together with a region-based mode where the user can select a part of an image and the system searches images (or parts of images) that are visually similar to the selected part. The on-line demo can be found at
`http://www-rocq.inria.fr/imedia/ikona/index.html`.

- **SIMPLIcity :** SIMPLIcity (Semantics-sensitive Integrated Matching for Picture LIbraries) is an image retrieval system developed at Stanford University [90]. In this system, images are represented by sets of regions, roughly corre-

sponding to objects, that are characterized by color, texture, shape and location properties. Segmentation is achieved by a simple algorithm based on k-means clustering in feature space. As opposed to region-wise retrieval (as in Blobworld for instance), images are retrieved as a whole by the help of a region matching scheme that incorporates properties of all the regions in an image to measure similarity. This overall similarity approach is to reduce the influence of inaccurate segmentation. The on-line demo can be found at
`http://wang.ist.psu.edu/IMAGE/`.

There are other systems for region-based retrieval in the literature, such as VisualSEEK [80] and Amore [60].

The region-based retrieval systems suffer from the main problem of "global" appearance matching systems in a "localized" manner. In this case, "regions" with the right composition but the wrong semantics may be retrieved. Objects are often oversegmented and their regions have dissimilar feature vectors. Simple distance measures used in these systems are not capable of retrieving all the regions in a database from the same class. Many complicated features are used in combination and it is difficult for users to judge their relative importance for different queries. These combinations are likely to be highly class-specific requiring automatic detection. In this thesis, a learning framework is proposed to address these issues in a region-based CBIR system.

2.1 Towards High Level Semantics

Humans are accustomed to utilize high level concepts, like objects, people, places, etc., to navigate through daily quests. These concepts come naturally to the users and querying at this level of abstraction is required for *successful* CBIR systems if success is defined to be user satisfaction. At this point, learning is ought to be an inevitable part of a CBIR system. Thus, some off-line and/or on-line processing is required to narrow down the semantic gap between the user needs and the current system replies [69]. There are some CBIR systems that are built with a learning framework (e.g. [12, 91, 15]) though this line of research is in its infancy stages with many breakthroughs yet to be made.

First of all, *finding tools* in the current literature should be mentioned. These tools use *template matching* techniques from object recognition as a way of estimating

object-level semantics. Template matching refers to finding objects by matching image patches with sample templates. A natural application of template matching is to construct whole-image templates that correspond to particular semantic categories [24, 77]. These templates can be constructed off-line and used to simplify querying by allowing a user to use an existing template. If object appearance is steady and suitable for template matching then specific object finding tools can be built such as systems for finding faces [68, 66], pedestrians [63], naked people [39] and horses [38]. However, building a template matching system to retrieve objects from various classes remains to be a difficult problem [37]. Preparation of generic templates is nontrivial and requires significant effort for each object class to be searched. Method also suffers from high variation of objects in appearance, pose, scale, articulation and occlusion, in natural image domain of CBIR.

A natural step in determining image semantics is to label the type of material image patches represent; for example, "grass", "buildings", etc., as opposed to "green", "grey". At this point, human assistance is inevitable. A method is required to establish classes and provide exemplars. Automatic annotation is one possible, useful tool to be utilized for speed and efficiency purposes and there is a recent promising work for automatic annotation and region labeling using statistical methods on large annotated databases [31, 11]. However, some form of human intervention is still required in the form of prior annotation..

Image or region classification is a separate problem that is to be studied on its own. However, CBIR is a good test field for various classification and/or learning algorithms at this stage of low-level image processing which transforms raw image pixels into new feature space or spaces where classification becomes feasible. In [82], Soysal *et al.* propose to use different types of classifiers in combination to extract semantic class information from standardized low level features and successfully classify images into certain classes like "indoor", "crowd", "sky", "forest", *etc.* In this thesis, an intermediate level region-based representation on top of low-level features is used which enables images containing distinctive objects like "cheetah", "flower", "horse", *etc.* to be classified and a CBIR system which utilizes this classification to query images at the level of objects is proposed.

Another advantage of CBIR applications in terms of integration of learning archi-

tectures is that human assistance is available at various levels. For instance, a recent trend is to utilize relevance feedbacks of users to refine queries according to user preferences. In the following, some systems which can be considered as good efforts in narrowing the gap between low-level feature extraction and higher level image semantics will be presented. These systems basically differ at the classification methods used and the type of learning adopted, i.e. on-line via relevance feedbacks or off-line via user supplied training data. Athitsos *et al.* [8] propose a system to classify an image as photograph or graphics to be used as a part of WebSeer [85], an image search engine for the Web. They used color tests for classification combined with multiple decision trees constructed using a training set of hand-labeled images. Belongie *et al.* [12] propose a Bayesian approach to classify image regions. In the work of Wood *et al.* [91], radial-basis function neural networks are used to classify regions with on-line training using relevance feedbacks from the user, and in a related work of Campbell *et al.* [15] multi layer perceptrons have been used for the same purpose but with a training phase instead of learning from relevance feedbacks. All these systems utilize classification to satisfy high level querying needs of users and/or for efficient indexing and retrieval purposes.

The system proposed in this thesis adopts a neurocomputing approach to learn and classify image regions. It is a region-based CBIR system with off-line supervised learning using fuzzy ARTMAP neural network architecture. Prior region classification avoids exhaustive similarity search in the retrieval phase in which only the regions from the query object class are ranked and displayed to the user.

2.2 Fuzzy ARTMAP Neural Network Architecture

Adaptive Resonance Theory (ART) is proposed by Stephan Grossberg in 1976 and a family of learning architectures followed in the last two decades that utilize ART approach. These architectures are:

- ART 1 [18] for unsupervised clustering of binary input patterns.
- ART 2 [18] for unsupervised clustering of analog (continuous-valued) input patterns.
- ARTMAP [18] for supervised classification of input patterns.

- fuzzy ARTMAP [19] generalization of ARTMAP using fuzzy set operations.

Fuzzy ARTMAP is a promising architecture that has evolved from the biological theory of cognitive information processing [16]. It has been used in many applications from diverse fields such as industrial design and manufacturing, the control of mobile robots, face recognition, remote sensing land cover classification, target recognition, medical diagnosis, electrocardiogram analysis, signature verification, tool failure monitoring, chemical analysis, circuit design, protein/DNA analysis, 3D visual object recognition, musical analysis, and seismic, sonar, and radar recognition (e.g., [16, 17, 23, 25, 42, 67, 76, 81]). Applications utilize the ability of ART systems to rapidly learn to classify large databases in a stable fashion and to focus attention upon feature groupings that are found to be important for each class. CBIR is a new application domain for ART systems to be introduced. Details of fuzzy ARTMAP algorithm will be given in Section 3.2.1

2.3 Visual Features and MPEG-7

MPEG-7 [2, 4] of ISO MPEG, also known as "Multimedia Content Description Interface", aims at providing standardized core technologies allowing description of audiovisual data content in multimedia environments. This is a challenging task, given the broad spectrum of requirements and targeted multimedia applications, and the broad number of audiovisual features of importance in such context. In order to achieve this goal, MPEG-7 standardizes: **Descriptors (D)** that define the syntax and semantics of each feature representation, **Description Schemes (DS)** that specify the structure and semantics of the relationships between their components, which may be both Ds and DSs, a **Description Definition Language (DDL)** to allow the creation of new DSs and possibly Ds, and to allow the extension and modification of DSs, **System Tools** to support multiplexing of description, synchronization issues, transmission mechanics, file format, etc. In this framework, DDL provides the mechanism to build a description scheme which in turn forms the basis for the generation of a description. The standard has provided a reference software, namely the eXperimentation Model (XM), which is the simulation platform for the descriptors and description schemes, coding schemes, and DDL. MPEG-7 Visual description tools included in the XM consist of basic structures and descriptors that cover visual features

such as color, texture, shape, motion and localization. Basically, there are eight color descriptors:

- Color space,
- Dominant Colors,
- Color Quantization,
- GoF/GoP Color ,
- Color Structure,
- Color Layout and
- Scalable Color Histogram;

three texture descriptors:

- Edge Direction Histogram,
- Homogeneous Texture and
- Texture Browsing;

and three shape descriptors:

- region-based shape,
- contour-based shape and
- 3D shape.

In Section 3.1.1, descriptors used in our application to form feature vectors are explained briefly. In [5], each of the above descriptors are explained in detail with the underlying algorithms and the supplementary issues such as color spaces, quantization methods and coding schemes used. MPEG-7 enables the development of efficient, configurable visual storage and access tools by supplying such content descriptors.

Low-level visual features can be extracted either globally from the whole image or locally as in region-based systems. A particular difficulty is the choice of feature extraction schemes. (feature selection problem). If more than one scheme is to be used, which is the usual case in most of the applications, it is not easy to determine how

to combine similarities measured according to different features. Most of the systems use a linear weighted summation of the distance values (heterogenous features) [77]. Weight adjustment may be handled manually by the user of the CBIR system or it may be automized via afore mentioned relevance feedback methods. However, it is not certain that a linear combination suitable for a certain class of images is also suitable for other classes of images. Minka *et al.* use techniques from machine learning in FourEyes [57] system to infer appropriate features from user annotation practices, using across-image groupings (which patches have been classified as "sky" in the past?) and in-image groupings (which patches are classified as "sky" in this image?). As a result a user annotating an image can benefit from past experience. In [87], Uysal *et al.* propose a similarity based learning method to determine the feature that best discriminates each class from MPEG-7 set and increase retrieval precision of the CBIR system using this class-specific best feature. In [77], Sheikholeslami *et al.* train a multi layer perceptron to find a nonlinear composite similarity value from the similarities measured using different feature extraction methods. Similarly, in [51] Lee *et al.* use radial basis function neural network for the same purpose of composing heterogenous features in a nonlinear way, in this system user feedbacks are used for incremental learning of feature importance values instead of off-line training.

In this thesis, a large feature vector with 239 dimensions is formed using MPEG-7 descriptors. Feature selection is handled implicitly via fuzzy ARTMAP architecture which is capable of attending to salient features of classes during classification as explained in Section 3.2.1.

2.4 Chapter Summary

In this chapter, some representative CBIR systems with region-based querying capabilities are overviewed as promising alternatives to global appearance matching of classical CBIR systems. Though successful in certain aspects, these systems are far from satisfying high level user needs during querying. Recent work that focuses on image/region classification or clustering is summarized as promising efforts towards high level semantics in CBIR applications. MPEG-7 content descriptors which are used as low-level feature extraction schemes in this thesis are introduced and fuzzy ARTMAP neural network architecture is presented shortly as the matching engine of

the proposed system.

CHAPTER 3

A CBIR SYSTEM BASED ON REGION CLASSIFICATION

In this chapter, we will introduce a new CBIR system that works on a presegmented image database. Users are able to query the database by sample regions. Our system classifies database regions prior to the retrieval in order to narrow down the search space. Fuzzy ARTMAP supervised learning algorithm [20] is used to classify the image regions. In the training phase, sample regions from user-defined classes are hand-labeled and feature vectors of these regions are fed into the fuzzy ARTMAP as input vectors. Fuzzy ARTMAP learns the mapping between these vectors and binary coded class identification numbers used as output vectors. In the database importing phase, all regions in the database are classified by the trained fuzzy ARTMAP system. In the querying phase, user selects a region and a label whose index is used for fast access. The regions with the associated label, are ranked with L_2 distance and displayed to the user.

Figure 3.1 presents the overview of the proposed system, which will be explained in detail in the following sections. In Section 3.1 preprocessing of images before being used by the system is explained. Sections 3.2, 3.3 and 3.4 present main processing stages together with their submodules. A detailed explanation of fuzzy ARTMAP is given in Section 3.2.1 for completeness and the motivation to choose this neural architecture is clarified as its key properties are presented.

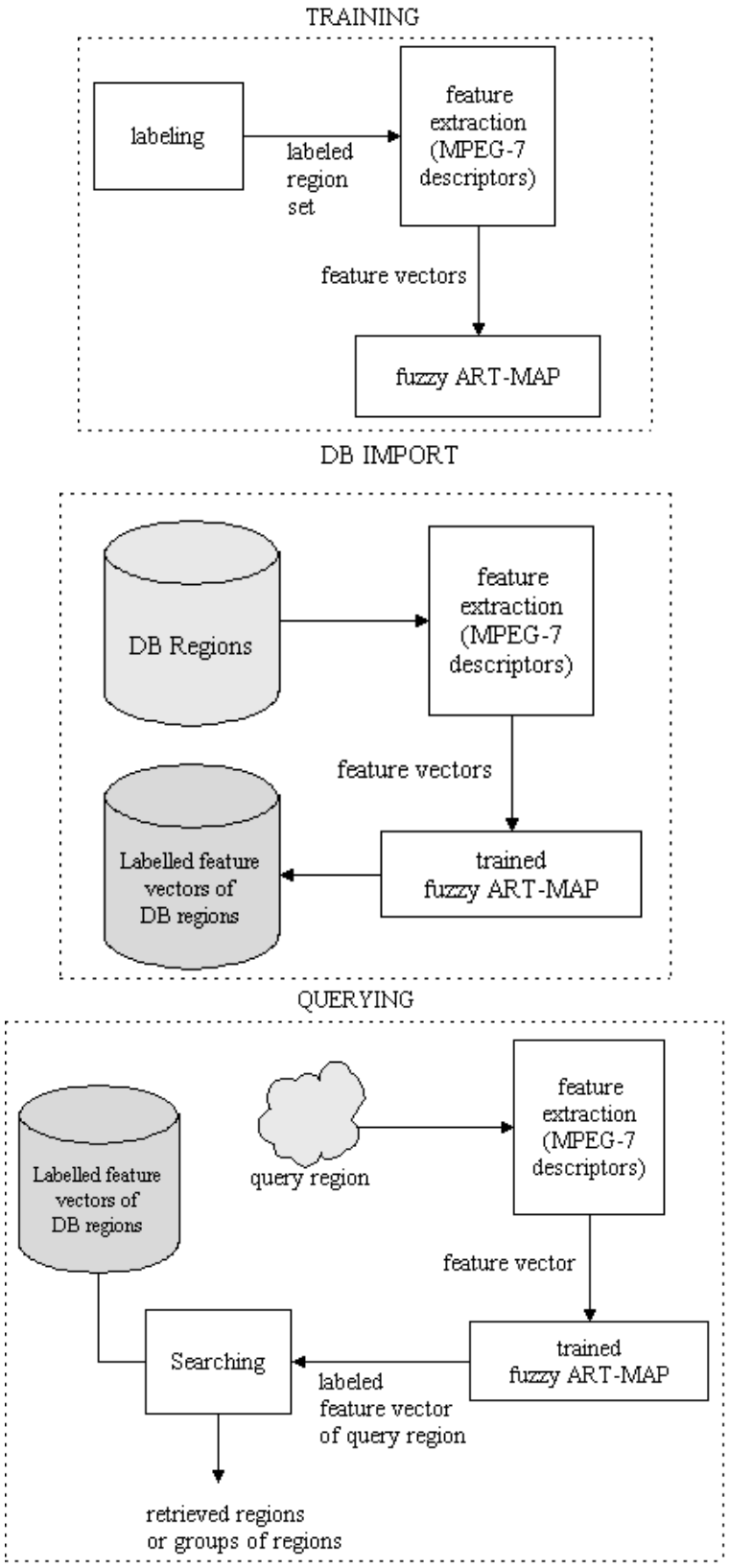

Figure 3.1: Overview of the proposed CBIR system.

Figure 3.2: Sample images from Corel data set. Each row consists of images from classes: bear, cheetah, elephant, penguin, plane.

3.1 Input Representation

Our system represents images as a collection of regions and works in the region domain in all of its phases. Images are segmented automatically using Normalized Cuts [78] algorithm to form regions. Figures 3.2, 3.3 show some sample images from the data set. Figures 3.4 and 3.5 display segmented forms of the images.

Similar to most of the segmentation algorithms, Normalized Cuts has the tendency to produce small regions. However, regions are coherent with respect to color and texture features and they roughly correspond to objects or parts of objects. 8 largest

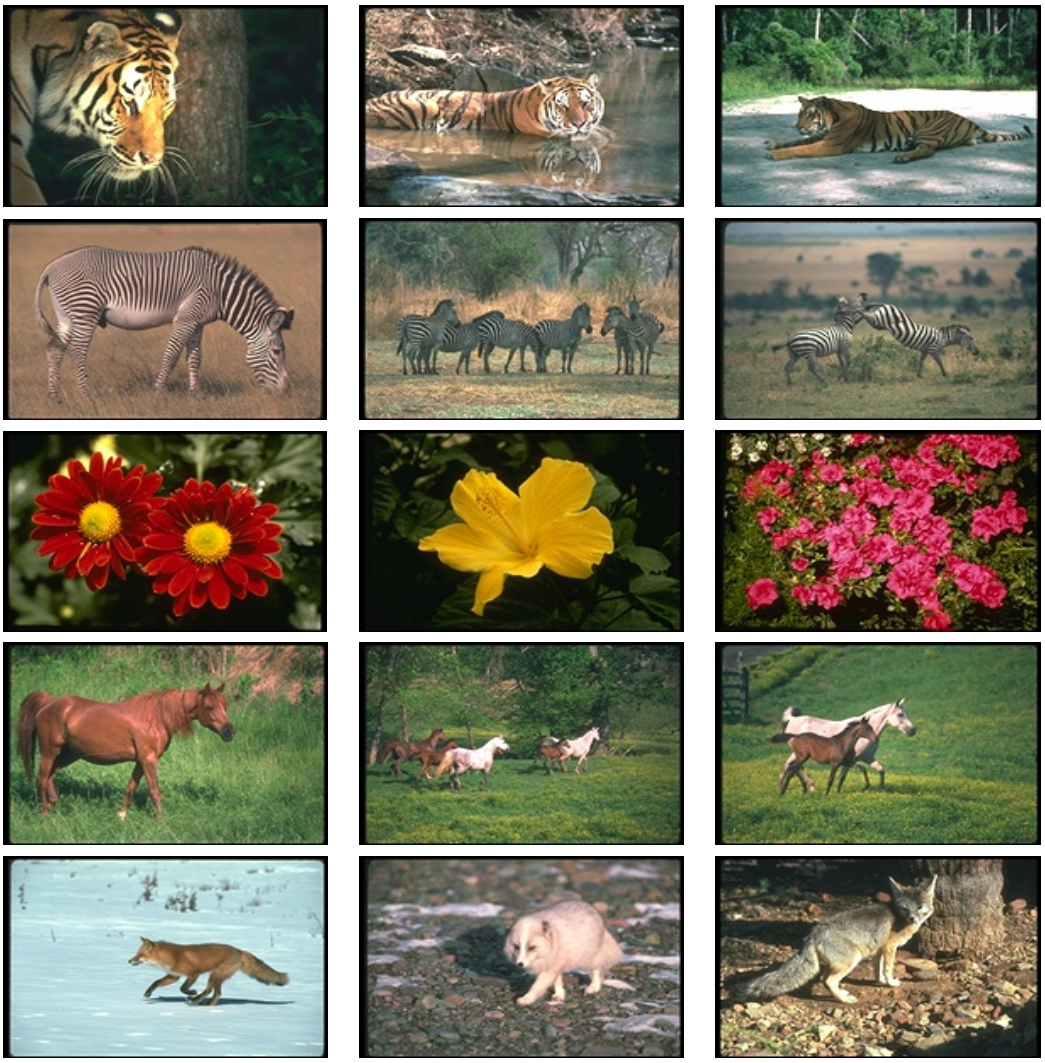

Figure 3.3: Sample images from Corel data set. Each row consists of images from classes: tiger, zebra, flower, horse, fox.

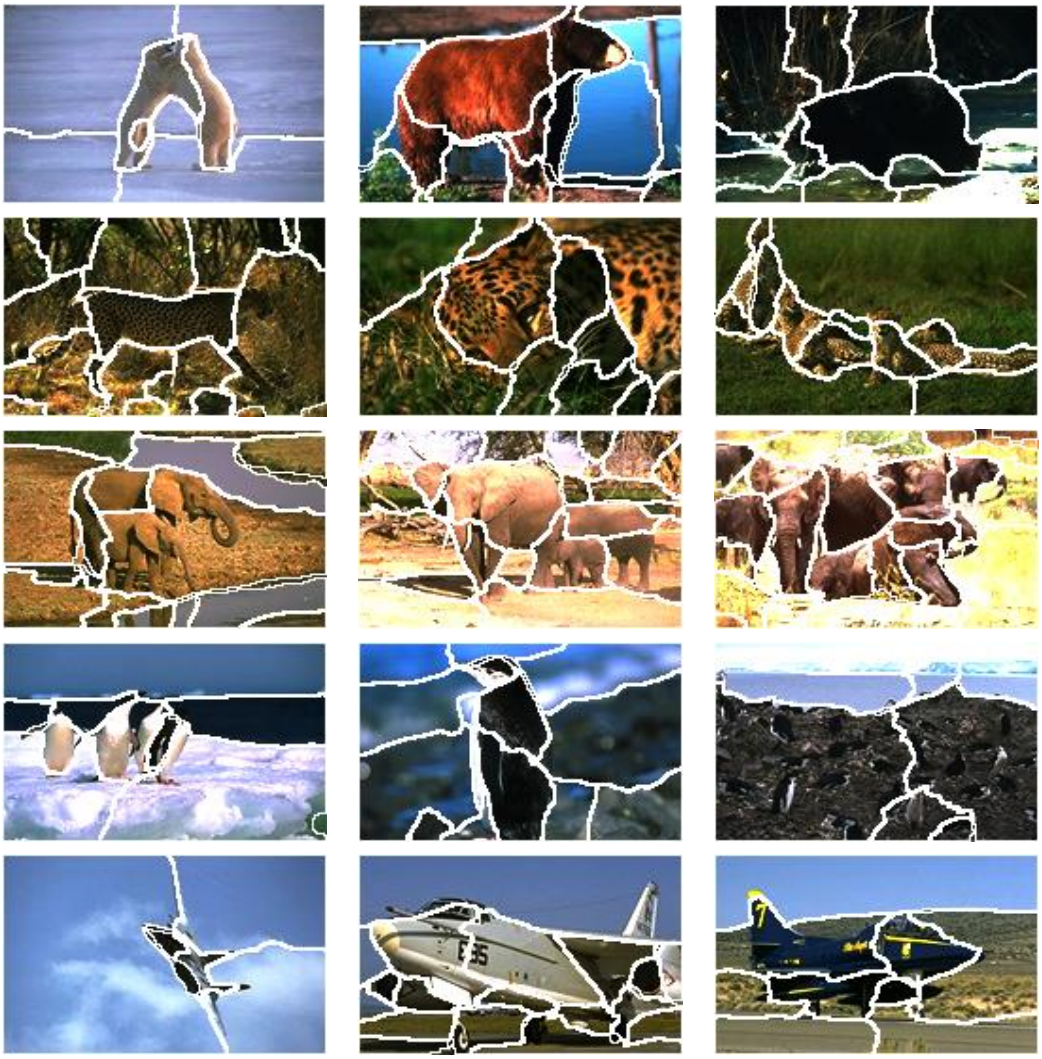

Figure 3.4: Sample outputs of Normalized Cuts segmentation for images in Figure 3.2.

Figure 3.5: Sample outputs of Normalized Cuts segmentation for images in Figure 3.3.

regions are selected from each image since regions smaller than a certain size become problematic during feature extraction process. So each image is represented as a collection of 3 to 8 regions.

3.1.1 Feature Extraction Module

At the lowest level, each region is represented by a feature vector extracted using MPEG-7 visual content descriptors. To include a large variety of features extracted with different computational methods, all the descriptors in MPEG-7 set that are suitable for our application are used. For this purpose, the color descriptors:

- Color Layout (12 features)
- Color Structure (32 features)
- Dominant Color (4 features)
- Scalable Color (16 features)

are used. The remaining descriptor in the set, GoF/GoP (Group of Frames/Group of Pictures), extends Scalable Color descriptor for still images to color description of video segments or a collection of still images, so the proposed CBIR system is out of the intended scope of this descriptor. The following texture descriptors are used:

- Edge Direction Histogram (80 features)
- Homogeneous Texture (60 features).

Texture Browsing descriptor is left out to avoid redundancy since it is only a compressed version of Homogeneous Texture descriptor designed for fast browsing applications. From shape descriptors, only Region Shape (35 features) is used. Contour Shape descriptor requires a clean segmentation of the object making it very unsuitable for our application since region contours that we get are usually far from exact contours of the objects they contain (see Figures 3.4 and 3.5). And Shape 3D descriptor is designed for intrinsic description of mesh models in 3D databases so the proposed CBIR system is out of the intended scope of this descriptor as well.

Feature extraction schemes of each descriptor are explained in detail in [5]. **Color Layout** descriptor specifies the spatial distribution of colors in Y, Cr, Cb color space

by the help of a DCT transformation. Local representative colors are determined by dividing the image into 64 blocks and averaging 3 channels on these blocks. After transformation, 12 of its coefficients are put into the feature vector. **Color Structure** descriptor captures both color content (similar to that of a color histogram) and the structure of this content in HMMD color space. This descriptor can distinguish between two images containing identical amounts of a given color in their histograms but with different structural groupings of the pixels with this color. **Dominant Color** descriptor is configured to specify a set of dominant colors in each region using RGB color space. In this thesis, the color value with the highest percentage of pixels is found and put into the feature vector together with its percentage. **Scalable Color** descriptor is a color histogram in the HSV color space, which is encoded by a Haar transform. The number of coefficients used in the scalable representation is chosen to be 16 among alternatives of 32 and 64 not to increase feature vector dimension. **Edge Direction Histogram** represents the spatial distribution of five types of edges, namely four directional edges (vertical, horizontal, 45 degree and 135 degree edges) and one non-directional edge; the descriptor divides the image into a 4 by 4 grid and finds the number of edges from each type in 16 regions. As a result, it encodes the histogram in 80 bins. The **Region Shape** descriptor utilizes a set of Angular Radial Transform coefficients. Angular Radial Transform is a 2D complex transform defined on a unit disk in polar coordinates, and this descriptor uses twelve angular and three radial functions to extract 35 features which are put directly into the feature vector. Lastly, **Homogeneous Texture** descriptor is actually the Gabor filter, one of the most commonly used texture feature in CBIR applications. Due to the inconvenience of the scheme used in the eXperimentation Model software for modification to make region-based filtering of images, Gabor Filter is implemented separately as described in [55]. Scale and orientation parameters are selected properly to be compliant with the standardized Homogeneous Texture descriptor as explained in [5]. In this way, Gabor features of individual regions are extracted separately by finding their mean oriented energies and energy variances. Resulting feature vector is of size 60 for 5 scales and 6 orientations (30 means and 30 variances).

All features are mapped to analog [0-1] scale and concatenated forming the final feature vector of size 239 to represent each region. This mapping is required by the

fuzzy ARTMAP module which will be explained next.

3.2 Training the System

Our system is trained off-line through the efficient graphical user interface, developed in this study. Initially, user is expected to determine classes with distinctive objects, which will be searched from the database. Appropriate labels for the regions of these classes should be chosen and a set of regions from each object class should be hand-labeled to form the training data. Some background regions might also be labeled though not required. In Chapter 4, the minimum number of regions to be labeled for satisfactory performance is found by experiments.

Figure 3.6 presents a snapshot of the GUI provided for region labeling. User opens an image which is displayed in segmented form and selects a region together with a label. This label is assigned to the selected region and the labeled region is added to the list of the selected training set. If a region is labeled the second time, the initial record is overwritten. Labels are automatically assigned unique identification numbers on their first presentation and all the training sets are assured to use the same id numbers for the same labels. It is relatively easy to label desired amounts of regions from each class using this GUI which is opened from main querying GUI of the system (Section 3.4).

3.2.1 Fuzzy ARTMAP Module

Once training sets are formed, by the help of another GUI which is also developed for this study, all the regions in a training set are cropped to their bounding boxes and features are extracted from these cropped regions. Feature vectors of a training set constitute input vectors and binary coded class identification numbers constitute output vectors to be mapped by fuzzy ARTMAP module.

ART stands for "Adaptive Resonance Theory", proposed by Stephen Grossberg in 1976. ART encompasses a wide variety of Neural Networks based explicitly on neurophysiology. ART networks are defined algorithmically in terms of a set of differential equations intended as plausible models of biological neurons. In practice, ART networks are implemented using analytical solutions or approximations to these differential equations.

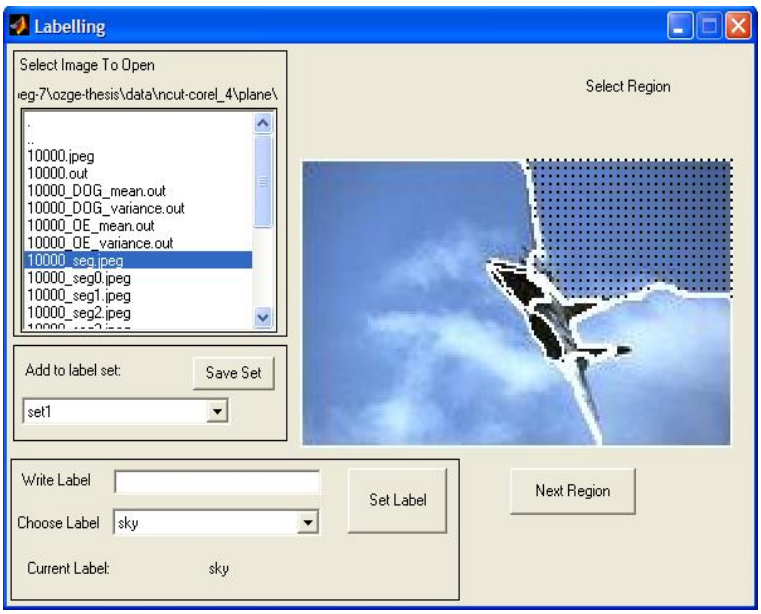

Figure 3.6: A snapshot from the region labeling GUI. Next Region button selects the regions one by one for label setting. The top-right region of the image is shown to be selected in this snapshot.

As discussed by Moore [59], the unsupervised ARTs are basically similar to many iterative clustering algorithms in which each case is processed by:

1. finding the "nearest" cluster seed (prototype or template) to that case

2. updating that cluster seed to be "closer" to the case,

where "nearest" and "closer" can be defined in several different ways depending on input representation and similarity measures. In ART, the framework is modified by introducing the concept of "resonance" so that each case is processed by:

1. finding the "nearest" cluster seed that "resonates" with the case

2. updating that cluster seed to be "closer" to the case.

"Resonance" is actually a matter of being within a certain threshold of a second similarity measure which will be explained in detail later in this section. A crucial feature of ART is that if no seed resonates with the case, a new cluster is created. This feature is said to solve the "stability-plasticity dilemma" which is to learn large and evolving databases quickly and stably without catastrophically forgetting the past knowledge.

ARTMAP [21, 18] is a class of Neural Network architectures that perform incremental "supervised" learning of recognition categories and multidimensional maps in response to input vectors. ARTMAP was initially proposed to classify input patterns represented as binary values. Carpenter *et al.* [19] refined the system to a general one 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 architecture contains two fuzzy ART units ART_a and ART_b with a MAP field in between (see Fig 3.7). The key operation takes place in ART_a which categorizes the input patterns 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. Figure 3.8 shows the notation used in the rest of the section.

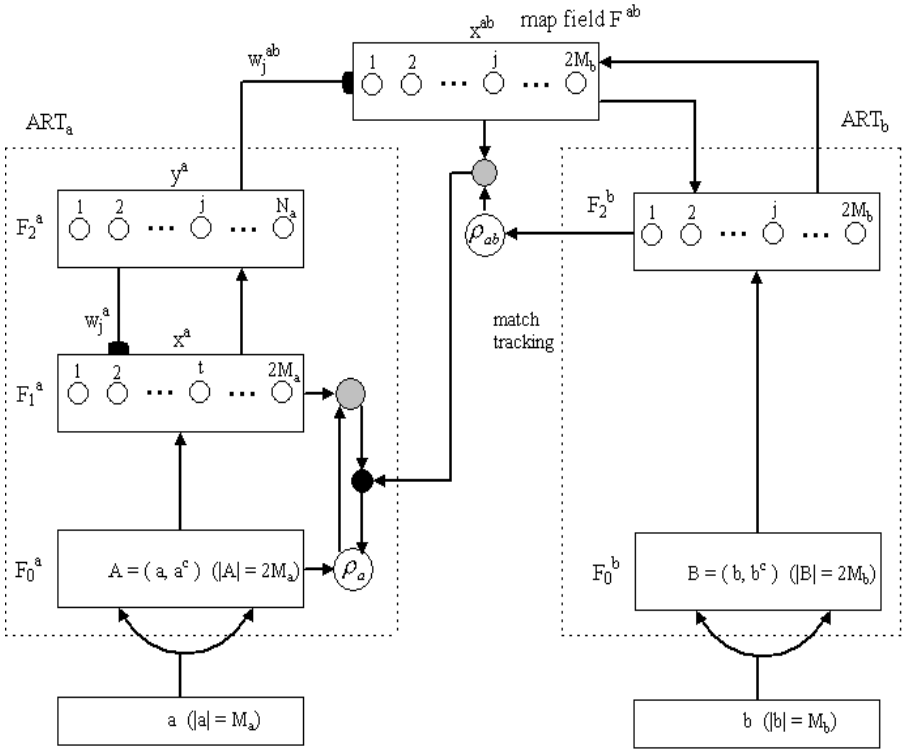

Figure 3.7: Simplified fuzzy ARTMAP architecture. Compared to the original architecture in [19], ART_b layer is modified not to contain the F_1^b layer of ART_b module. The output of F_0^b is directly sent to F_2^b layer. And the clusters of ART_b are directly the complemented class codes with this simplified architecture.

To solve category proliferation problem observed in noisy data, which is the ten-

dency of cluster seeds to degenerate to zero vector and the continual creation of new cluster seeds, Carpenter *et al.* [19] proposed to normalize the input vector for fuzzy ARTMAP architecture. Normalization is achieved when $\| \mathbf{A} \| \equiv \gamma$ for some $\gamma > 0$, for all inputs $\mathbf{A}$. In [19], *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. The complement coded input $\mathbf{A}$ is defined to be

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

where $a_i^c \equiv 1 - a_i$. Note that

$$\begin{aligned} \| \mathbf{A} \| &= \| (a, a^c) \| \\ &= \sum_{i=1}^{M_a} a_i + \left(M_a - \sum_{i=1}^{M_a} a_i \right) \\ &= M_a, \end{aligned}$$

so inputs are automatically normalized when they are complement coded.

| $\mathbf{v}$ | : size of vector $\mathbf{v}$
$\| \mathbf{v} \|$: norm of vector $\mathbf{v}$, where $\| \mathbf{v} \| \equiv \sum_{i=1}^{|\mathbf{v}|} v_i$
$\mathbf{a}$: input vector to F_0^a
$\mathbf{A}$: input vector to F_1^a
$\mathbf{b}$: input vector to F_0^b
$\mathbf{B}$: input vector to F_2^b
M_a, M_b : size of input vectors (i.e. | $\mathbf{a}$ |, | $\mathbf{b}$ | respectively)
N_a, N_b : final number of category nodes in F_2^a and F_2^b respectively ($N_b = 2M_b$)
$\mathbf{x^a}$: F_1^a activity vector
$\mathbf{x^{ab}}$: map field activity vector
α : choice parameter
β : learning rate parameter
ρ_a : vigilance for ART_a
$\bar{\rho_a}$: base-line vigilance for ART_a
ρ_{ab} : vigilance for match tracking in the map field

Figure 3.8: Nomenclature.

Operation in ART_a: There are three fields of nodes in ART_a (see Fig 3.7): F_0^a represents the current input pattern in complemented form, F_1^a receives both bottom-up input from F_0^a and top-down input from a field, F_2^a, that represents the active

code or category. Each F_2^a category node j ($j = 1, \ldots, N_a$) is associated with a vector $\mathbf{w}_j^a \equiv (w_{j\ 1}, \ldots, w_{j\ 2M_a})$ of adaptive weights, or so-called *long term memory* (LTM) traces. Initially all the category nodes are said to be *uncommitted* and

$$w_{j\ 1}(0) = \ldots = w_{j\ 2M_a} = 1. \tag{3.1}$$

A category node becomes *committed*, when its weight vector is updated for the first time. The weight vector of a category node constitute the *learned expectancy* of the system for that category. This expectancy is updated for each member of a category during training and at the end of training it becomes a sort of template representing the patterns of that category. Each LTM trace w_{ji} is monotonically nonincreasing through time and hence converges to a limit [19].

There are three parameters that determine the dynamics of the system: choice parameter $\alpha > 0$, learning rate parameter $\beta \in [0, 1]$, and a vigilance parameter $\rho_a \in [0, 1]$. For each input $\mathbf{A}$ and F_2^a category node j, the *choice function*, T_j, is defined by

$$T_j(\mathbf{A}) = \frac{\| \mathbf{A} \wedge \mathbf{w}_j^a \|}{\alpha + \| \mathbf{w}_j^a \|},$$

where the fuzzy AND operator is defined by

$$(\mathbf{p} \wedge \mathbf{q})_i = min(\mathbf{p}_i, \mathbf{q}_i)$$

for any same-dimensional vectors $\mathbf{p}$ and $\mathbf{q}$. The *category choice* of the system is indexed by J, where

$$T_J = max\{T_j : j = 1, \ldots, N_a\}. \tag{3.2}$$

In this case, the J^{th} node of F_2^a is said to become active. At any time, only one of the category nodes can be active so if more than one of T_j's are maximal, the category j with the smallest index is chosen. When a category node J becomes active, its top-down weights are sent as input to F_1^a layer and the activation $\mathbf{x}^a$ at this layer becomes

$$\mathbf{x}^a = \mathbf{A} \wedge \mathbf{w}_J^a. \tag{3.3}$$

If the *match function*, $\| \mathbf{x}^a \| / \| \mathbf{A} \|$ meets the vigilance criterion, i.e. if

$$\frac{\| \mathbf{A} \wedge \mathbf{w}_J^a \|}{\| \mathbf{A} \|} \geq \rho_a, \tag{3.4}$$

then the system enters into a *resonance* state. This is the state where input pattern matches the expectancy for that category well enough (to the degree vigilance determines) to cause a resonance between the two activation fields of ART_a. Learning can only occur when this resonance state is achieved. Otherwise, i.e. when

$$\frac{\| \mathbf{A} \wedge \mathbf{w}_J^a \|}{\| \mathbf{A} \|} < \rho_a,$$

a *mismatch reset* occurs. Then the value of the choice function T_J is set to 0 for the duration of the presentation of the current input $\mathbf{A}$ to prevent the persistent selection of the same category during search. A new index J is then chosen, by Equation 3.2. The search process continues until a chosen category satisfies Equation 3.4. Once search ends, the weight vector $\mathbf{w}_J^a$ is updated according to the equation

$$\mathbf{w}_J^{a\ (new)} = \beta(\mathbf{A} \wedge \mathbf{w}_J^{a\ (old)}) + (1 - \beta)\mathbf{w}_J^{a\ (old)}. \tag{3.5}$$

When β is set to 1, this is equivalent to the *fast learning* option where the weight vector is directly equated to $\mathbf{x}^a$, the activation at F_2^a field (Equation 3.3) when the resonance occurs. It is notable that not the input pattern but the attended portions of it are learned. This causes detection of relevant feature groupings of the categories and focusing attention on these portions while trying to match new input patterns. If the search ends at a new code, the code's active memory representation begins by learning the current input itself, i.e. the expectation for this new category is set to to the current input pattern (fast learning).

For analog vectors, the degree to which $\mathbf{q}$ is a fuzzy subset of $\mathbf{p}$ is given by the term

$$\frac{\| \mathbf{p} \wedge \mathbf{q} \|}{\| \mathbf{q} \|}$$

[19]. When $\alpha \cong 0$, the choice function T_j primarily reflects the degree to which the weight vector $\mathbf{w}_j^a$ is a fuzzy subset of the input vector $\mathbf{A}$. If

$$\frac{\| \mathbf{A} \wedge \mathbf{w}_j^a \|}{\| \mathbf{w}_j^a \|} = 1,$$

then $\mathbf{w}_j^a$ is a fuzzy subset of $\mathbf{A}$ and category j is said to be a *fuzzy subset choice* for input $\mathbf{A}$. When a fuzzy subset category choice exists, it is always selected over other choices. In this case, by Equation 3.5, no recoding occurs if j is selected since $\mathbf{A} \wedge \mathbf{w}_j^a = \mathbf{w}_j^a$. If more than one category is a fuzzy subset choice, the small but

positive parameter α breaks the tie by choosing J that maximizes $\| \mathbf{w}_j^a \|$ among the fuzzy subset choices.

Resonance depends on the degree to which $\mathbf{A}$ is a fuzzy subset of $\mathbf{w}_J^a$, by Equation 3.4. If category j is a fuzzy subset choice, then the match function value is given by

$$\frac{\| \mathbf{A} \wedge \mathbf{w}_j^a \|}{\| \mathbf{A} \|} = \frac{\| \mathbf{w}_j^a \|}{\| \mathbf{A} \|}.$$

Thus, choosing J to maximize $\| \mathbf{w}_j^a \|$ among fuzzy subset choices also maximizes the opportunity for resonance in Equation 3.4. There is a close linkage between fuzzy subsethood and ART choice/resonance/learning that forms the foundation of the computational properties of fuzzy ART [20].

Operation in ART_b: In the original fuzzy ARTMAP algorithm both ART_b and ART_a are proposed to be fuzzy ART modules; however, in this thesis operation in ART_b module is simplified for efficiency purposes eliminating the redundancy in this module. Classes are represented with binary coding so if there are k classes in the training set than $M_b = log_2 k$ bits are necessary to form the class representation. The class code of the current input is complemented and fed into the input layer F_0^b having $2M_b$ nodes to represent the class (see Fig 3.7). Nodes of this layer are directly connected to F_2^b, so categories of ART_b are directly class codes, denoted by $\mathbf{B}$. This would still be the case if this module were to work with exact fuzzy ART algorithm and with identical architecture to ART_a due to the nature of inputs to this module.

Match tracking in the MAP field: F^{ab} is the map field between ART_a and ART_b modules of the system. This field is used to form associations between categories of each module via the *match tracking* rule. According to this rule, the vigilance parameter of ART_a increases in response to a predictive mismatch at ART_b and the category structure of ART_a is reorganized not to repeat the error on subsequent presentations of the same input. Nodes in the F_2^a field of ART_a are connected to nodes in F^{ab} with adaptive weights, thus each node j in F_2^a is associated with a weight vector $\mathbf{w}_j^{ab} \equiv (w_1^{ab}, \ldots, w_{N_b}^{ab})$. If node J of F_2^a resonates for a given input $\mathbf{A}$, then the activation at F^{ab} is given by

$$\mathbf{x}^{ab} = \mathbf{B} \wedge \mathbf{w}_J^{ab}.$$

where $\mathbf{B}$ is the category coded at ART_b for the current input $\mathbf{A}$. Nodes in the F_2^b field of ART_b are connected to nodes in F^{ab} with 1-to-1 pathways as opposed to adaptive

weights, so the category in ART_b directly affects the activation in F^{ab}. If a mismatch occurs between the prediction at ART_b and $\mathbf{w}_J^{ab}$, then a search for a better category in ART_a is initiated, as follows. At the start of each input presentation, ART_a vigilance parameter ρ_a is equated to a baseline vigilance, $\bar{\rho_a}$. The map field vigilance parameter is ρ_{ab}. If

$$\frac{\| \mathbf{x}^{ab} \|}{\| \mathbf{B} \|} < \rho_{ab},$$

then ρ_a is increased until it is slightly larger than match function for $\mathbf{A}$, i.e.

$$\rho_a = \frac{\| \mathbf{x}^a \|}{\| \mathbf{A} \|} + \epsilon,$$

where ϵ denotes a small positive number. Then

$$\| \mathbf{x}^a \| = \| \mathbf{A} \wedge \mathbf{w}_J^a \| < \rho_a \| \mathbf{A} \|,$$

where J is the index of the active node in F_2^a. When this occurs node J is inhibited (i.e. $T_J = 0$ for subsequent presentations of input $\mathbf{A}$) and the search for a new category J^* that satisfies both

$$\| \mathbf{x}^a \| = \| \mathbf{A} \wedge \mathbf{w}_{J^*}^a \| \geq \rho_a \| \mathbf{A} \|$$

and

$$\| \mathbf{x}^{ab} \| = \| \mathbf{B} \wedge \mathbf{w}_{J^*}^{ab} \| \geq \rho_{ab} \| \mathbf{B} \|,$$

continues. The search looks for possible categories among committed nodes in F_2^a initially and if none satisfy above conditions then the next uncommitted node is chosen since an uncommitted node always satisfies them. Hence search is assured to end at least by committing a previously uncommitted node. When the search ends the following learning rule associates the chosen category J with the class coded by $\mathbf{B}$:

$$\mathbf{w}_J^{ab\ (new)} = \beta(\mathbf{B} \wedge \mathbf{w}_J^{ab\ (old)}) + (1 - \beta)\mathbf{w}_J^{ab\ (old)}. \tag{3.6}$$

Since $\mathbf{w}_j^{ab}$ are all initialized to be vectors of one as in Equation 3.1, and if β is set to one as in fast learning option then $\mathbf{w}_J^{ab}$ is equated to the class coded by $\mathbf{B}$. Thus adaptive weights from F_2^a category nodes to the map field learn classes corresponding to these categories. In the recall phase, when an input is presented to ART_a module, $\mathbf{w}_J^{ab}$ vector of the chosen category J in F_2^a determines the class of the input.

Algorithms that are used to train and test the fuzzy ARTMAP module used in this thesis are given by Algorithm 1 and Algorithm 2. In Algorithm 1, all inputs in

the training set are presented once to the network and each input is assured to be assigned to a category node in F_2^a field and mapped to the corresponding class via F^{ab} field. Algorithm 2 simply returns the class mapped by the node that becomes active in F_2^a layer when the input is presented.

One well-known drawback of ART systems is that they are vulnerable to the order of pattern presentation during training. Categories are formed differently depending on this order. Carpenter *et al.* [19] propose a voting scheme in which more than one classifiers are trained with randomly ordered versions of the same training set and the class of an unknown pattern is taken to be the majority vote. This scheme is also implemented and in Chapter 4, it has been showed that our system's retrieval performance is not effected significantly from pattern ordering during training.

3.2.2 Fuzzy ARTMAP as a Classifier in CBIR

In this thesis, fuzzy ARTMAP is chosen as a classifer regarding some of its key characteristics that well-suit our application. First of all, it achieves a many-to-one mapping, i.e. the number of category nodes in F_2^a field may exceed the number of classes in the training data. Input patterns are categorized as much as needed for them to be recognized by the fuzzy ARTMAP module and more than one category can be mapped to the same class. For example, black and white regions of a penguin which are labeled with the same keyword "penguin" can be categorized by different nodes in F_2^a field while being associated to the same class. This is a very nice property since in this way, regions can be grouped under proper classes even if their feature vectors are dissimilar. Such a property cannot be achieved by classical distance-based similarity measuring CBIR systems. For instance, our "bear" class (see Figure 3.2 and Chapter 4) having polar, black and brown types has been split into 3 different classes in Carson *et al.* 's experiments [22].

Another key characteristic of fuzzy ARTMAP is that it learns top-down expectations (weights between F_2^a and F_1^a) that can bias the system to ignore masses of irrelevant data. The system selectively searches for recognition categories whose top-down expectations provide an acceptable match to bottom-up data. Each top-down expectation begins to focus attention upon, and bind, that cluster of input features that are part of the prototype which it has already learned, while suppressing fea-

Algorithm 1 train fuzzy ARTMAP module

initialize F_2^a layer with one uncommitted node $\{N_a = 1$ initially$\}$

initialize $\bar{\rho_a}$, ρ_{ab} $\{\simeq 1\}$, α $\{> 0\}$, β

for each input $(\mathbf{a}, classid)$ in the training set **do**

$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$

form class code $\mathbf{b}$ from *classid* using binary coding

$\mathbf{B} = (\mathbf{b}, \mathbf{b}^c)$

$\rho_a = \bar{\rho_a}$

deinhibit all nodes in F_2^a layer

repeat

repeat

for each node j in F_2^a **do**

if j is not inhibited **then**

$T_j = \frac{\|\mathbf{A} \wedge \mathbf{w}_j^a\|}{\alpha + \|\mathbf{w}_j^a\|}$

else

$T_j = 0$

end if

find J with $T_J = max\{T_j\}$

end for

$\mathbf{x}^a = \mathbf{A} \wedge \mathbf{w}_J^a$

if $\frac{\|\mathbf{x}^a\|}{\|\mathbf{A}\|} < \rho_a$ **then**

inhibit J

if number of inhibited nodes $= N_a$ **then**

add an uncommitted node to F_2^a and increase N_a by 1

end if

end if

until $\frac{\|\mathbf{x}^a\|}{\|\mathbf{A}\|} \geq \rho_a$ {resonance between F_1^a and F_2^a}

$\mathbf{x}^{ab} = \mathbf{B} \wedge \mathbf{w}_J^{ab}$

if $\frac{\|\mathbf{x}^{ab}\|}{\|\mathbf{B}\|} < \rho_{ab}$ **then**

$\rho_a = \frac{\|\mathbf{x}^a\|}{\|\mathbf{A}\|} + \epsilon$

end if {go back to ART_a category search loop}

until $\frac{\|\mathbf{x}^{ab}\|}{\|\mathbf{B}\|} \geq \rho_{ab}$ {resonance between F_2^a and F^{ab}}

$\mathbf{w}_J^{a\ (new)} = \beta(\mathbf{x}^a) + (1 - \beta)\mathbf{w}_J^{a\ (old)}$

$\mathbf{w}_J^{ab\ (new)} = \beta(\mathbf{x}^{ab}) + (1 - \beta)\mathbf{w}_J^{ab\ (old)}$

end for

Algorithm 2 find the class of input vector **a** with the trained fuzzy ARTMAP module

$\mathbf{A} = (\mathbf{a}, \mathbf{a}^c)$

for each node j in F_2^a **do**

$T_j = \frac{\|\mathbf{A} \wedge \mathbf{w}_j^a\|}{\alpha + \|\mathbf{w}_j^a\|}$

end for

find J with $T_J = max\{T_j\}$

if $T_J > 0$ **then**

return the *classid* coded with normalized $\mathbf{w}_J^{ab}$

else

return *nochoice* {fuzzy ARTMAP could not match this input, no response}

end if

tures that are not. In the recall phase, only the group of features that are attended by the expectation of each class determine the classification. This process can be regarded as a salient feature detection in the form of expectations for each class. For instance, the system can learn to suppress texture and shape features to categorize "sky" regions, or it can learn to suppress color features and attend to Gabor filter responses at certain scales to categorize "flowers". In this way, a large variety of features extracted with different computational methods are used in combination and their relative importance to discriminate different classes are detected.

A confidence measure, called *vigilance*, calibrates how well an input pattern needs to match the expectation of a category in order for the corresponding category to resonate with it and be chosen. High vigilance values result in finer categorization and with low vigilance a broader categorization is achieved. In the extreme cases, when the vigilance is 1, all the input patterns are put into different categories and final number of category nodes in the F_2^a field equals 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 in the F_2^a field equals 1. In fuzzy ARTMAP, the match tracking rule sets the vigilance for ART_a module. Thus according to training data, fine or broad categorization can be achieved to map the input vectors to classes properly. For instance, if the user labels a particular dog kind with its name, e.g. "shepard", but all other dogs with only the keyword "dog", then the system raises vigilance to achieve finer categorization for shepard category which would otherwise be put into one of dog categories. In this way, various levels of abstraction or classification of

input patterns can be learned concurrently. This property well-suits the user guided category determination phase proposed in this thesis.

ART based systems are proposed as a solution to the *stability-plasticity dilemma* which is to learn large and evolving databases quickly and stably without catastrophically forgetting the past knowledge. 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 [16] as desirable for CBIR systems. Proposed system can be modified in the future to continue learning across querying sessions as in relevance feedback methods.

3.3 Importing a Database

In the database importing phase, by the help of a simple GUI, all the regions in the database are cropped to their bounding boxes as in training phase and their features are extracted. These feature vectors are presented to the trained fuzzy ARTMAP module and index files are formed. The index file of each class simply contains region identification numbers of the patterns of that class in the database.

3.4 Querying by Example Regions

In the querying phase, user determines the class of distinctive objects to query by selecting its label and presents the system a query region which is used to rank the results. If the query region is from the database, then the label it has been assigned in the database importing phase is also displayed to the user. This gives the user a chance to supply a different query region for better ranking if the current one is already misclassified. L_2 distance between each region in the query class index file and the query region is found and results are displayed to the user in decreasing similarity. If more than one regions are retrieved from the same image then it is displayed only once in the first place. Figure 3.9 is a snapshot taken from a querying session.

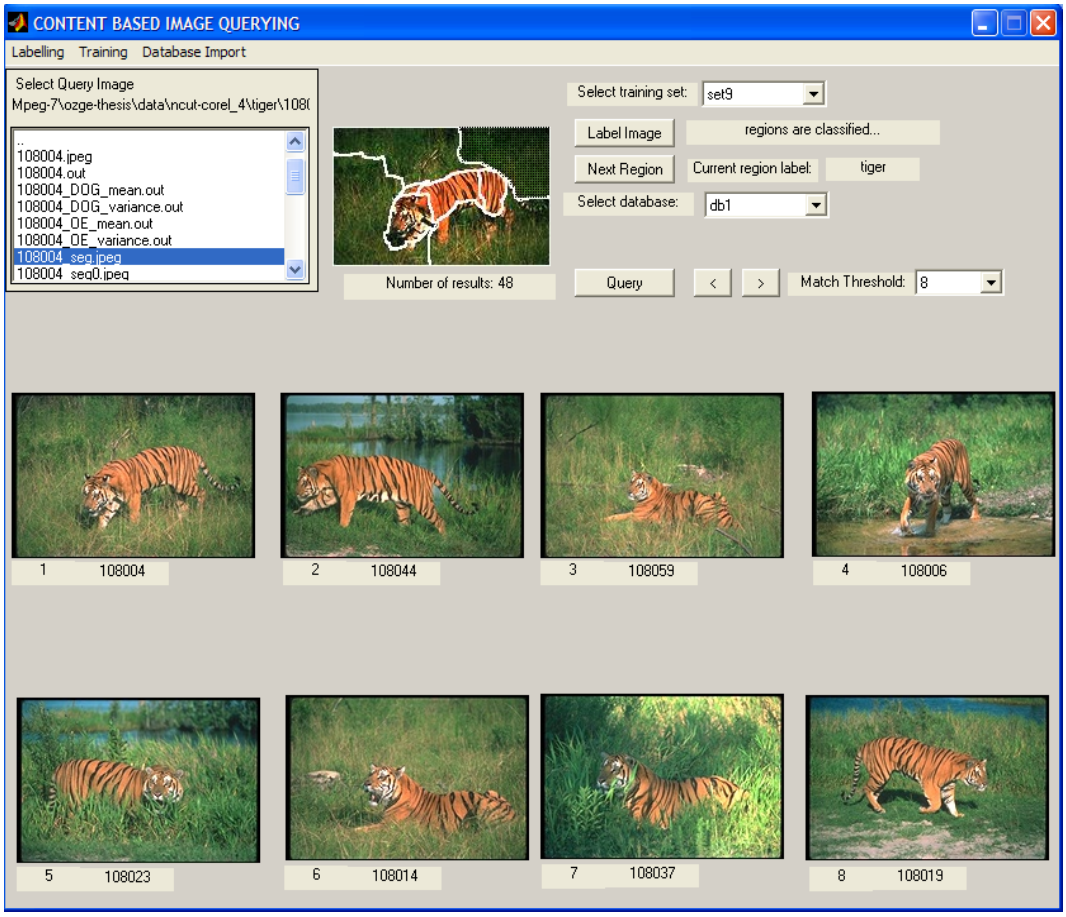

Figure 3.9: A snapshot from the region querying GUI. Query image is displayed in segmented form for region selection.

CHAPTER 4

Experiments

In this chapter, we present results of our experiments designed to analyze the effectiveness of the proposed CBIR system based on fuzzy ARTMAP algorithm. Specifically, the power of fuzzy ARTMAP is tested in classification of image regions prior to the retrieval. Experiments are also prepared to measure size of the search space during retrieval of images containing distinctive objects from a database.

4.1 Database

In our experiments, we used images from Corel data set [1] which is comprised of CD's with 100 images each. Each CD contains images that are grouped according to some semantic or visual criteria, such as images from France, images of colored patterns, images containing certain animals like cheetahs, eagles, elephants, etc. We have formed a database from images of 10 classes with distinctive objects: *tiger*, *cheetah*, *elephant*, *bear* (with polar bears, brown bears and black bears), *penguin*, *plane*, *flower*, *zebra*, *fox* and *horse*. Each row in Figures 3.2 and 3.3 contain three sample images from a class to give an idea about the amount of variation among members. Our database contains 938 images with 100 images from each class except for zebra class having 38 images. After the database importing phase, the number of regions in the database is 6661.

Table 4.1: Hand-labeled region counts from different classes in each training set.

class	set 1	set 2	set 3	set 4	set 5
tiger	5	10	23	33	33
bear	5	10	20	30	30
plane	5	10	20	30	30
elephant	5	10	21	31	31
penguin	5	10	22	32	32
cheetah	5	10	20	30	30
horse	5	10	20	30	30
fox	5	10	20	30	30
zebra	5	10	19	29	30
flower	5	10	20	30	30
grass	5	10	6	16	0
river	0	0	6	9	0
sky	5	10	6	15	0
tree	5	10	6	16	0
snow	5	10	6	16	0
rocks	0	0	6	6	0
water	5	10	5	12	0
total	75	150	246	396	306

4.2 Training Sets

Five training sets with varying sizes and regions are formed in order to observe the effect of the number of training data. These sets contain labeled samples of regions from ten major classes and also some labeled background regions for completeness. Images are chosen randomly from each class and the regions they contain are labeled to form three sets. A fourth set is formed by merging second and third sets and a fifth set is formed by eliminating background regions from fourth set to observe the effect on classification and retrieval. The number of labeled samples from each class are given in Table 4.1 for each set. With the labeling GUI developed, an image is selected and desired regions are labeled in less than a minute.

4.3 Preliminary Classification Experiments

To demonstrate effectiveness of fuzzy ARTMAP in region classification, some preliminary experiments are performed. Five test sets are formed from only object regions, i.e. by eliminating background regions from training sets. Table 4.2 gives the number of regions in test sets.

Table 4.2: Number of regions in test sets prepared from training sets of Table 4.1.

class	set1-test	set2-test	set3-test	set4-test (set5)
tiger	5	10	23	33
bear	5	10	20	30
plane	5	10	20	30
elephant	5	10	21	31
penguin	5	10	22	32
cheetah	5	10	20	30
horse	5	10	20	30
fox	5	10	20	30
zebra	5	10	19	30
flower	5	10	20	30
total	50	100	205	306

Table 4.3: Number of ART_a nodes formed during training with presentation of the patterns in their original form. Recoding rate (β) is 0.9 .

	set 1	set 2	set 3	set 4	set 5
# of nodes	21	21	34	35	25

During labeling operation training sets are prepared such that exemplars of each class are nearly in succession. Initial experiments are performed by presenting patterns to the network in this original order during training. It is observed that number of nodes allocated in categorization module, ART_a, of ARTMAP is larger than the number of classes (see Table 4.3). This is due to the many-to-one mapping capability of fuzzy ARTMAP. Classes like "penguin" have both white and black regions with dissimilar feature vectors which are represented by different nodes but mapped to the same class. Similarly for "bear" class having polar, black and brown types all with the same label. (This class has been split into 3 different classes in Carson *et al.* 's experiments [22].) Number of nodes allocated is increased with the number of patterns in training sets to capture within-class variations that are not obvious as in examples given above.

Performance results with different combinations of train and test sets are given in Table 4.4 for trained networks of Table 4.3. Table 4.4 demonstrates an expected increase in classification performance with the increasing number of training patterns. Users of our retrieval system are supposed to label sufficient amount of exemplars from each class they wish to query. Performances are satisfactory for sets 3 and 4. So from Table 4.1, we see that 15 to 30 regions should be labeled for each class. Usually

Table 4.4: Test performances with different combinations of train and test sets ($\beta = 0.9$) with 10 classes.

Training Set	Test Set	Total Correct	Total Error
set1	set1-test	50 (100.00 %)	0 (0.00 %)
	set2-test	56 (56.00 %)	44 (44.00 %)
	set3-test	91 (44.39 %)	114 (55.61 %)
	set4-test	143 (46.73 %)	163 (53.27 %)
set2	set1-test	41 (82.00 %)	9 (18.00 %)
	set2-test	100 (100.00 %)	0 (0.00 %)
	set3-test	109 (53.17 %)	96 (46.83 %)
	set4-test	176 (57.52 %)	130 (42.48 %)
set3	set1-test	29 (58.00 %)	21 (42.00 %)
	set2-test	52 (52.00 %)	48 (48.00 %)
	set3-test	205 (100.00 %)	0 (0.00 %)
	set4-test	216 (70.59 %)	90 (29.41 %)
set4	set1-test	40 (80.00 %)	10 (20.00 %)
	set2-test	77 (77.00 %)	23 (23.00 %)
	set3-test	170 (82.93 %)	35 (17.07 %)
	set4-test	306 (100.00 %)	0 (0.00 %)
set5	set1-test	40 (80.00 %)	10 (20.00 %)
	set2-test	78 (78.00 %)	22 (22.00 %)
	set3-test	171 (83.41 %)	34 (16.59 %)
	set4-test	306 (100.00 %)	0 (0.00 %)

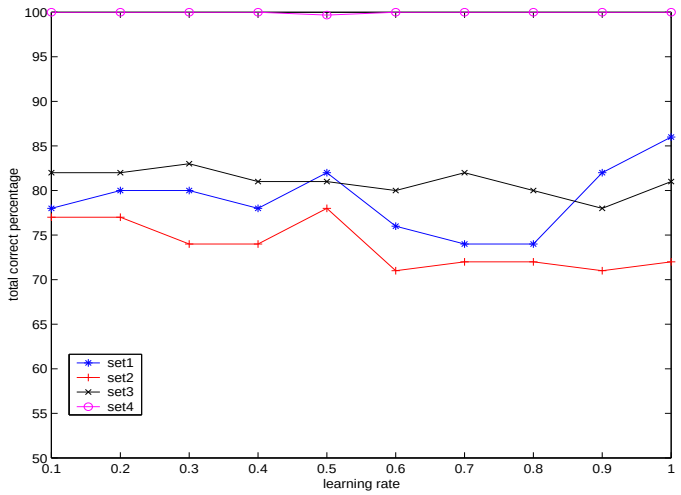

Figure 4.1: Effect of learning rate (β) on test performances of training set 4.

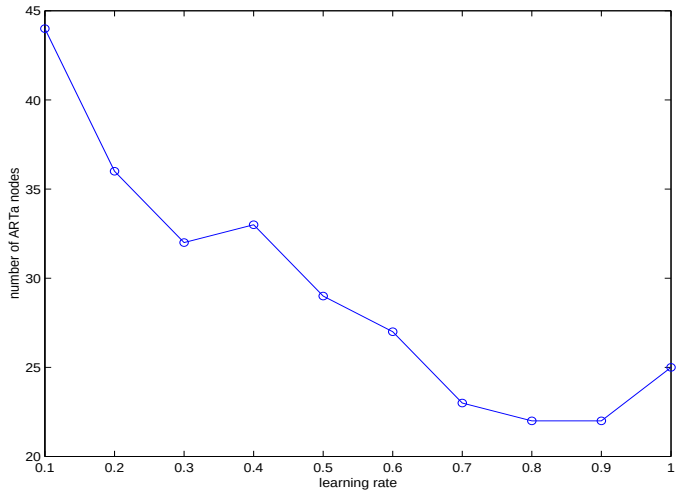

Figure 4.2: Effect of learning rate (β) on number of nodes formed in ART_a after training with set 4.

there are more than one regions from a single object in images, this means that 10-15 images should be chosen and labeled from each class using our system interface. Also, it is observed from Table 4.4 that classification performances of sets 4 and 5 are almost the same, remember that set 5 is formed by removing all background regions from set 4.

Figure 4.1 is a plot of total correct percentages of the network, when trained with varying recoding rates (β) using set 4 and Figure 4.2 shows the number of nodes formed in ART_a after training. We observe that while test performances stay nearly the same, number of nodes formed in ART_a increases with decreasing recoding rate.

This is expected since low recoding rates cause templates to be updated very slowly. Hence, entering into resonance with more specific templates becomes difficult which increases number of category nodes formed in ART_a. We used β value of 0.9 for the rest of our experiments as significant performance differences are not observed for other values.

There are no other parameters of fuzzy ARTMAP to be set before training, however, it is well-known that fuzzy-ARTMAP is vulnerable to the order of pattern presentation during training. Different clusters are formed with different presentations of the inputs during training. Carpenter *et al.* [19] propose a voting scheme to overcome this vulnerability which is demonstrated to increase the classification performance as well. Table 4.5 shows new performance rates when a classifier module with 10 voters is used. Each voter is trained with a different random presentation of the same input set and the category of a test pattern is taken to be the majority vote. If Tables 4.5 and 4.4 are compared, a performance increase though not significant is observed for each training set. During our experiments, we also observed that increasing number of voters does not affect the classification performance significantly as demonstrated in Figure 4.3 for set 4.

These preliminary experiments show that sets 1 and 2 do not contain sufficient amount of training patterns regarding their pure recognition performances. However, about 80% of the test patterns are classified correctly when training sets contain sufficient number of class regions as in sets 4 and 5. Still these sets contain around 30 regions which is a small number when compared to typical training set sizes of learning systems. This shows that with relatively less training fuzzy ARTMAP is effective at classifying image regions. Sets 3, 4 and 5 will be used in the next section which demonstrates the retrieval performance of our system. The effect of voting scheme on retrieval will also be shown.

4.4 Retrieval Experiments

All the regions in the database are treated as unknown patterns and classified using our trained fuzzy ARTMAP modules. The training is performed in 6 different ways resulting in 6 different modules. Three 1-voter systems are trained with sets 3, 4 and 5 with their original ordering of input patterns. Three 10-voter systems are trained

Table 4.5: Test performances with different combinations of train and test sets ($\beta = 0.9$). 10 voters are used each of which is trained with a different random presentation of the input set.

Training Set	Test Set	Total Correct	Total Error
set1	set1-test	50 (100.00 %)	0 (0.00 %)
	set2-test	67 (67.00 %)	33 (33.00 %)
	set3-test	104 (50.73 %)	101 (49.27 %)
	set4-test	159 (51.96 %)	147 (48.04 %)
set2	set1-test	41 (82.00 %)	9 (18.00 %)
	set2-test	100 (100.00 %)	0 (0.00 %)
	set3-test	114 (55.61 %)	91 (44.39 %)
	set4-test	186 (60.78 %)	120 (39.22 %)
set3	set1-test	32 (64.00 %)	18 (36.00 %)
	set2-test	54 (54.00 %)	46 (46.00 %)
	set3-test	205 (100.00 %)	0 (0.00 %)
	set4-test	233 (76.14 %)	73 (23.86 %)
set4	set1-test	41 (82.00 %)	9 (18.00 %)
	set2-test	78 (78.00 %)	22 (22.00 %)
	set3-test	171 (83.41 %)	34 (16.59 %)
	set4-test	306 (100.00 %)	0 (0.00 %)
set5	set1-test	41 (82.00 %)	9 (18.00 %)
	set2-test	76 (76.00 %)	24 (24.00 %)
	set3-test	174 (84.88 %)	31 (15.12 %)
	set4-test	306 (100.00 %)	0 (0.00 %)

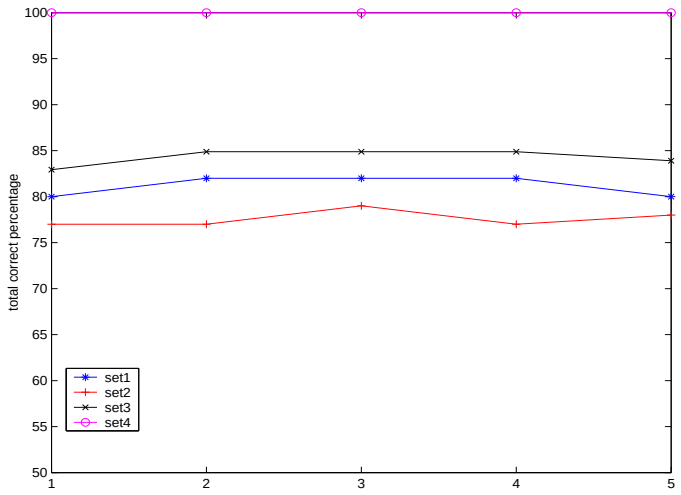

Figure 4.3: Effect of increasing number of voters on test performances of training set 4. Number of voters are 3, 5, 10, 30 and 50.

Table 4.6: The number of labeled database regions from each class after classification with 1 voter systems.

class	set 3	set 4	set 5
tiger	1016	392	406
bear	162	319	1569
plane	1134	1401	1471
elephant	267	230	172
penguin	1687	392	743
cheetah	393	404	424
horse	365	1091	723
fox	275	185	168
zebra	211	233	407
flower	865	516	578
grass	66	269	0
river	35	125	0
water	4	285	0
tree	153	486	0
snow	16	219	0
sky	8	110	0
rocks	4	4	0
total	6661	6661	6661

with 10 random orderings the same sets. There is no practical way of knowing exact number of regions from each class in the database, but the number of regions assigned to each class is shown in Tables 4.6 and 4.7.

It is observed from Tables 4.6 and 4.7 that the number of training patterns effect the results substantially. Set 3 contains fewer number of labeled background regions, compared to the distinctive object regions, as a result fewer number of background regions are labeled in the database. This is partly due to fuzzy ARTMAP's representing classes with more exemplars with more generic templates since the template gets more and more generic by "fuzzy-or"ing with more and more patterns. And partly due to the fact that there are generally fewer number of background regions compared to distinctive object regions in the images. Although we do not know the exact number of regions from each class, regions from *rocks*, *water*, *river*, etc. classes do not appear as frequently as distinctive object regions. (The latter are assured to be present in all relevant images of their class.) Obviously, no region is assigned to a class when it is not learned by the system during training. This is observed for training set 5 in Tables 4.6 and 4.7. Tables contain the value 0 for all background classes as they are

Table 4.7: The number of labeled database regions from each class after classification with 10 voter systems.

class	set8	set9	set10
tiger	586	365	408
bear	146	357	1488
plane	1166	1512	1449
elephant	111	211	162
penguin	1000	467	945
cheetah	410	393	457
horse	415	907	778
fox	1260	298	237
zebra	643	257	243
flower	829	663	494
grass	45	258	0
river	15	113	0
sky	4	217	0
tree	7	344	0
snow	12	196	0
rocks	8	99	0
water	4	4	0
total	6661	6661	6661

not present in this training set (Table 4.1). All the regions are forced to be classified to be one of the object regions in this case. Since primary aim of our application is object retrieval, wrong classification of background regions does not constitute a big drawback. Indeed, experiments presented in the rest of this section will demonstrate that retrieval performances with sets 4 and 5 are almost the same.

To see the querying performance of the system, Algorithm 3 is run for each class. This algorithm finds average recall and precision for each class by automatically querying with all regions of each image in the class. The recall and precision for an image is set to be the values of one of its regions giving best recall and precision. Since one of these regions is assured to belong to the distinctive object of the class, it is assumed that no other region can give better recall and precision values for that class. The motivation is that if a user runs a query with this image, he/she would select one of the distinctive object regions and at best get the precision and recall found with this automatic method. In this way, the burden of preparing a query manually for each image of the class is avoided. The algorithm initially finds the set of regions assigned to this class using index file of the class. This set is used for querying and the search

space is reduced depending on the size of this set for each class.

Algorithm 3 Find the average recall and precision for a given class

t = total number of relevant images of this class in the database
m = match threshold
find **SLR** where **SLR** = set of regions labeled with the label of this class
for each image **i** of the class
    for each region **r** of **i**
        for each region **rr** in **SLR**
            find the Euclidean distance between **r** and **rr**
        sort **SLR** according to the distance values
        take 1 region per image and find the sorted image list **LSI**
        correct = the relevant images of this class in top **m** images in **LSI**
        recall_region = **correct** / **t**
        precision_region = **correct** / **m**
    recall_image = best of all **recall_region** values
    precision_image = best of all **precision_region** values
recall_class = average of all **recall_image** values
precision_class = average of all **precision_image** values

To determine the effectiveness of the system, a baseline is needed for comparison. This baseline is obtained by querying the database without using the classification. An automatic method similar to Algorithm 3 is used which simply finds the Euclidean distance of each region in the database to all other regions (Algorithm 4) instead of to the labeled set of the class (**SLR** in Algorithm 3). This method is feasible in these experiments since the database size is not so large (6661 regions). However, the complexity is $O(n^2)$ where n is the number of regions in the database and as the database grows this method loses its feasibility. Our system reduces the size of the search space by labeling regions in the database, and it not only speeds up the querying process but also enables larger databases to be searched effectively. Table 4.8 shows the CPU time that each session of querying takes, where the speed up achieved

Table 4.8: The CPU time in seconds for each querying session.

class	base	set 3	set 4	set 5
tiger	215.87	32.11	12.44	12.98
bear	207.24	4.91	9.42	45.95
plane	143.3	24.27	29.9	31.66
elephant	211.7	8.48	7.23	5.47
penguin	190.94	48.75	11.32	21.27
cheetah	193.99	11.72	11.86	12.53
horse	205.95	11.33	33.03	22.22
fox	205.05	8.49	5.83	5.31
zebra	67.33	2.24	2.4	4.22
flower	220.61	26.81	16.01	18.09

is clearly observed.

Algorithm 4 Find the baseline average recall and precision for a given class.

t = total number of relevant images of this class in the database

m = match threshold

for each image **i** of the class

    for each region **r** of **i**

        for each region **rr** in the database

            find the Euclidean distance between **r** and **rr**

        sort the database regions according to the distance values

        take 1 region per image and find the sorted image list **LSI**

        correct = the relevant images of this class in top **m** images in **LSI**

        recall_region = **correct** / **t**

        precision_region = **correct** / **m**

    recall_image = best of all **recall_region** values

    precision_image = best of all **precision_region** values

recall_class = average of all **recall_image** values

precision_class = average of all **precision_image** values

Algorithms 3 and 4 are run for 5 different match threshold values: 10, 20, 30, 40 and 50. Performance of our systems are shown in Figures 4.4, 4.5 and 4.6. Figures are the recall/precision plots of the class averages. Both baseline values and 1-voter and 10-voter systems' values are plotted for comparison.

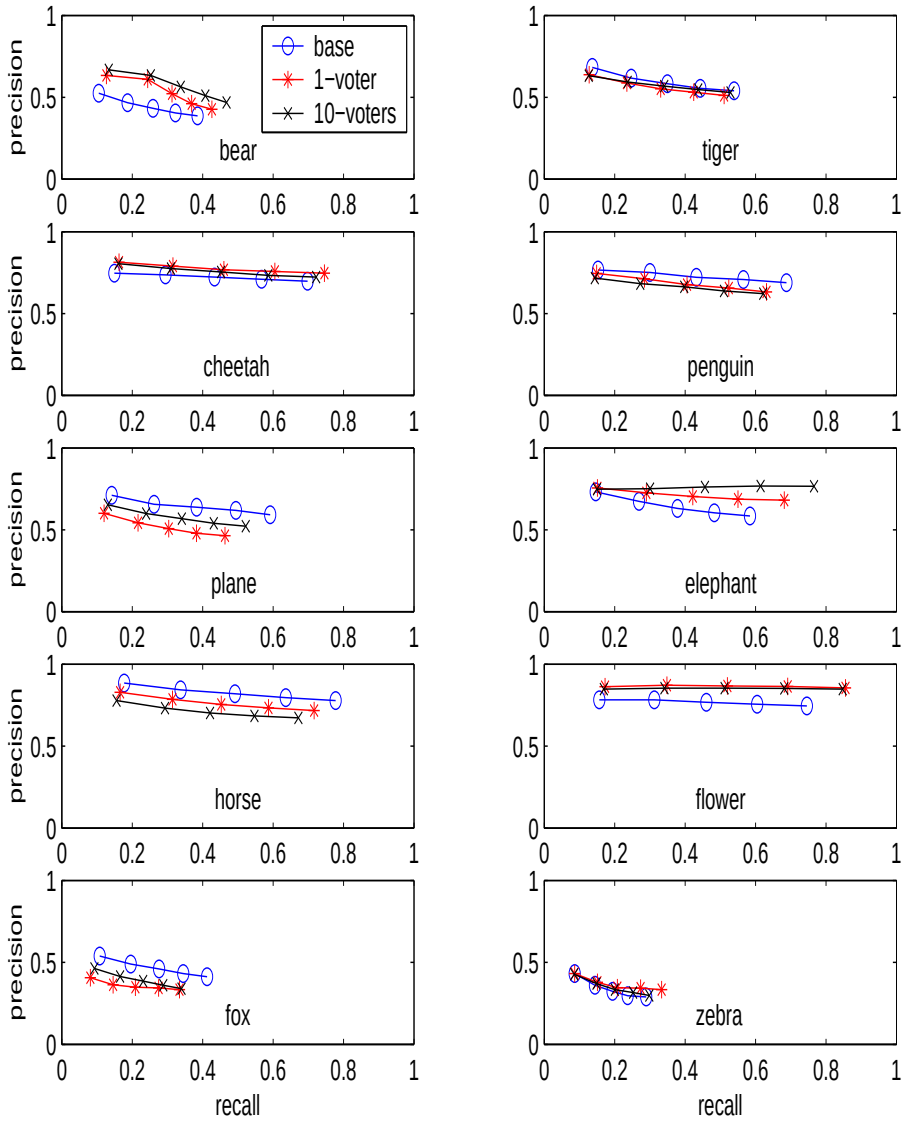

Figure 4.4: Average recall and precision values for each class with training set 3.

For most of the classes, precision rates are comparable to the baseline values, which demonstrates the effectiveness of our system in narrowing the search space. Also, for certain classes like "flowers" precision is highly improved as exhaustive searching of the database creates many false alarms. Only "fox" and "plane" classes seem to suffer from our method, which might be due to the high variability within members

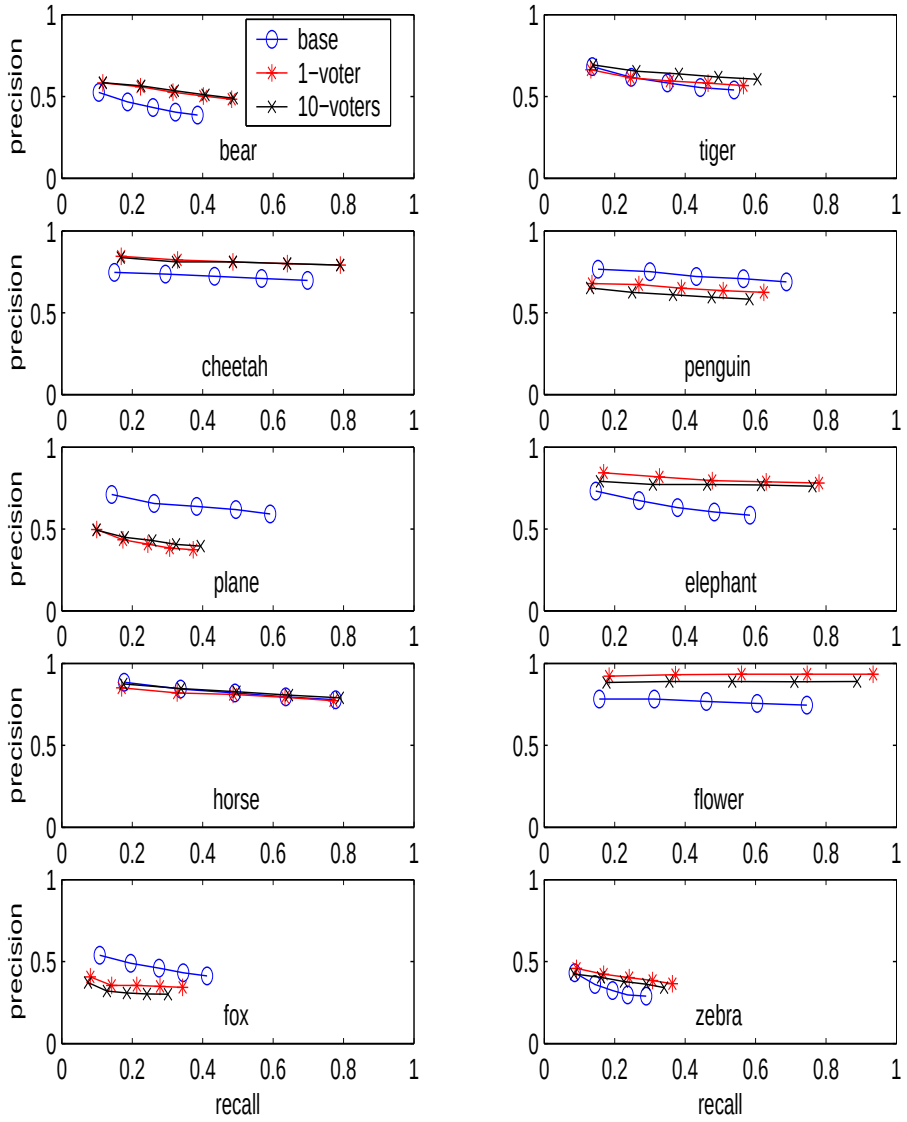

Figure 4.5: Average recall and precision values for each class with training set 4.

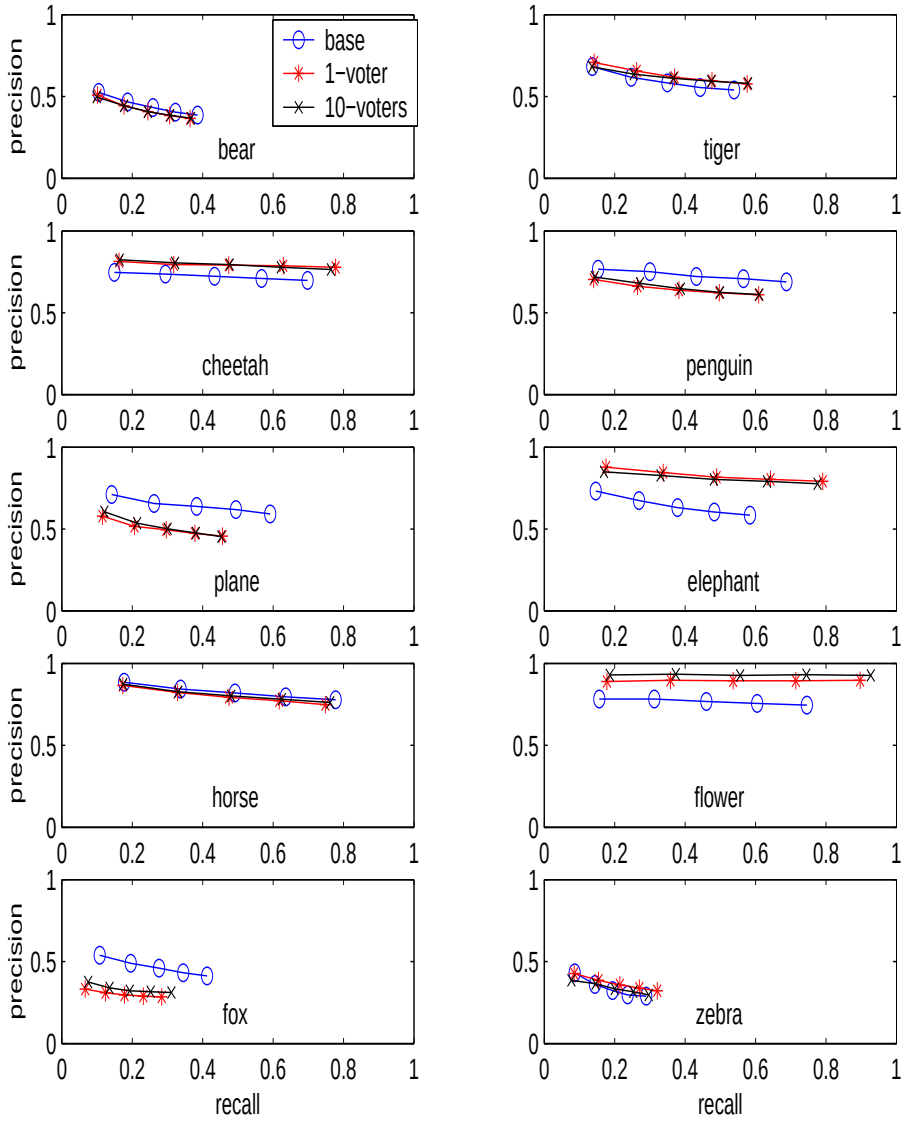

Figure 4.6: Average recall and precision values for each class with training set 5.

of these classes (see Figures 3.2 and 3.3 for examples). Apparently, fuzzy ARTMAP is insufficient to group input patterns of these classes and find salient features to attend to for each group. Although a large number of database regions are labeled as "plane", most of these are false alarms regarding low precision values. Also it should be noted that voting mechanism is not effective in increasing precision values proving the relative unimportance of the presentation of the patterns during training for this task.

CHAPTER 5

CONCLUSIONS AND FUTURE DIRECTIONS

In this thesis, a CBIR system with learning capabilities is proposed to query object content of databases. Images are represented as collections of coherent regions with respect to color and texture which are segmented via Normalized Cuts algorithm. At the lowest level, MPEG-7 content descriptors are used to form a feature vector to represent each region. It has become possible to attach meaning to these image regions via classification of their feature vectors. A user-friendly graphical interface has been developed for integrating a training module into the system. Tedious training set preparation task has become an easier labeling operation via this interface.

Fuzzy ARTMAP Neural Network architecture is used as the matching engine of our system. Preliminary experiments show that ARTMAP is effective at attending to different salient features for each class and classifying regions successfully despite the high dimension of the feature space. Comparably high precision values in retrieval with respect to exhaustive searching further prove this effectiveness. Correct classification of certain classes, like planes, due to their high within-class variation remains as a challenge. This difficulty might be overcome in a retrieval system by the use of certain background regions to support the query. As a future study, we plan to improve our system in this direction to allow querying via combinations of different regions from both object and background classes to increase retrieval precision.

Fuzzy ARTMAP systems have the capability of learning to classify databases in a stable fashion without catastrophically forgetting their past knowledge. This property makes them especially suitable to continue learning during querying similar to sys-

tems using relevance feedbacks of users. These feedbacks can be used to present new patterns to the fuzzy ARTMAP module after each querying session and continously update its current experience. Performance achieved via the off-line training stage proposed in this thesis can be improved by this on-line training strategy.

The system can also be utilized to organize image databases, since unknown regions can be labeled and/or auto-annotated. It is capable of attaching meaning to regions after being trained by the user. This serves for the purpose of accessing object content and semantic organization of databases, the ultimate aim of retrieval systems.

REFERENCES

[1] Corel Data Set. http://www.corel.com/products/clipartandphotos.

[2] MPEG7 Home page. http://ipsi.fraunhofer.de/delite/Projects/MPEG7/.

[3] Web archive. http://www.archive.org.

[4] Mpeg-7 context, objectives and technical roadmap. Technical Report ISO/IEC JTC1/SC29/WG11/N2861, International Organization for Standardisation, Coding of Moving Pictures and Audio, 1999.

[5] Multimedia content description interface, part 3 visual. Technical Report ISO/IEC JTC1/SC29/WG11/N4062, International Organization for Standardisation, Coding of Moving Pictures and Audio, 2001.

[6] N. Arica and F. T. Yarman Vural. BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recognition Letters*, 24:1627–1639, 2003.

[7] L. H. Armitage and P. G. B. Enser. Analysis of user need in image archives. *Journal of Information Science*, 23(4):287–299, 1997.

[8] V. Athitsos, M. J. Swain, and C. Frankel. Distinguishing photographs and graphics on the world wide web. In *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.

[9] J. Bach, C. Fuller, A. Gupta, A. Hampapur, B. Gorowitz, R. Humphrey, R. Jain, and C. Shu. Virage image search engine: An open framework for image management. In *Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases IV*, 1996.

[10] J. R. Bach, S. Paul, and R. Jain. A visual information management system for the interactive retrieval of faces. *IEEE Transactions on Knowledge and Data Engineering*, 5(4):619–628, 1993.

[11] K. Barnard, P. Duygulu, N. de Freitas, D. A. Forsyth, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.

[12] S. Belongie, C. Carson, H. Greenspan, and J. Malik. Recognition of images in large databases using a learning framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[13] N. Boujemaa, J. Fauqueur, M. Ferecatu, F. Fleuret, V. Gouet, B. Le Saux, and H. Sahbi. IKONA: Interactive generic and specific image retrieval. In *Proceedings of International workshop on Multimedia Content-Based Indexing and Retrieval (MMCBIR'2001)*, 2001.

[14] A. C. Bovik, M. Clark, and W. S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

[15] N. W. Campbell, W. P. J. Mackeown, B. T. Thomas, and T. Troscianko. Interpreting image databases by region classification. *Pattern Recognition (Special Edition on Image Databases)*, 30(4):555–563, 1997.

[16] G. A. Carpenter. ART neural networks: Distributed coding and ARTMAP applications. Technical Report CAS/CNS TR-2000-005, MA: Boston University, 2000.

[17] G. A. Carpenter, M. Gjaja, S. Gopal, and C. Woodcock. ART networks in remote sensing. *IEEE Transactions on Geoscience and Remote Sensing*, 35(2):308–325, 1997.

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

[19] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3(5):698–713, 1992.

[20] G. A. Carpenter, S. Grossberg, N. Markuzon, and D. B. Rosen. Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks*, 4:759–771, 1991.

[21] G. A. Carpenter, S. Grossberg, and J. H. Reynolds. ARTMAP: A self-organizing neural network architecture for fast supervised learning and pattern recognition. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN-91)*, 1991.

[22] C. Carson, S. Belongie, H. Greenspan, and Jitendra Malik. Blobworld: Color and texture based image segmentation using EM and its application to image querying and classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, 2002.

[23] T. P. Caudell, S. D. G. Smith, R. Escobedo, and et al. Nirs: Large scale ART-1 neural architectures for engineering design retrieval. *Neural Networks*, 7:1339–1350, 1994.

[24] S.-F. Chang, W. Chen, and H. Sundaram. Semantic visual templates - linking visual features to semantics. In *Proceedings of IEEE International Conference on Image Processing*, 1998.

[25] C. G. Christodoulou, J. Huang, M. Georgiopoulos, and et al. Design of gratings and frequency selective surfaces using fuzzy ARTMAP neural networks. *Journal of Electromagnetic Waves and Applications*, 9:17–36, 1995.

[26] G. Chuang and C.-C. Kuo. Wavelet descriptor of planar curves: Theory and applications. *IEEE Transactions on Image Processing*, 5:56–70, 1996.

[27] J. G. Daugman. Complete discrete 2-d gabor transforms by neural networks for image analysis and compression. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(7), 1988.

[28] Y. Deng and B. S. Manjunanth. An efficient low-dimensional color indexing scheme for region-based image retrieval. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[29] J. Dowe. Content-based retrieval in multimedia imaging. In *Proceedings SPIE Storage and Retrieval For Image and Video Databases*, 1993.

[30] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification.* John Wiley and Sons, Inc., New York, 2001.

[31] P. Duygulu, K. Barnard, N.d. Freitas, and D. A. Forsyth. Object recognition as machine translation: learning a lexicon for a fixed image vocabulary. In *Seventh European Conference on Computer Vision (ECCV)*, volume 4, pages 97–112, 2002.

[32] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks-a review. *Pattern Recognition*, 35:2279–2301, 2002.

[33] P. G. B. Enser. Query analysis in a visual information retrieval context. *Journal of Document and Text Management*, 1(1):25–39, 1993.

[34] P. G. B. Enser. Progress in documentation pictorial information retrieval. *Journal of Documentation*, 51(2):126–170, June 1995.

[35] C. Faloutsos, R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3(3-4):231–262, 1994.

[36] M.M. Fleck, D.A. Forsyth, and C.Bregler. Finding naked people. In *4th European Conference on Computer vision*, volume 2, pages 591–602, 1996.

[37] D. A. Forsyth and J. Ponce. *Computer Vision: a modern approach.* Prentice-Hall, 2001.

[38] D.A. Forsyth and M.M. Fleck. Body plans. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[39] D.A. Forsyth, M.M. Fleck, and C. Bregler. Finding naked people. In *Proceedings of European Conference on Computer Vision*, 1996.

[40] T. Gevers and A.W.M. Smeulders. PicToSeek: Combining color and shape invariant features for image retrieval. *IEEE Transactions on Image Processing*, 9(1):102–119, January 2000.

[41] A. A. Goodrum. Image information retrieval: An overview of current research. *Informing Science*, 3(2):63–66, 2000.

[42] F. M. Ham and S. Han. Classification of cardiac arrhythmias using fuzzy ARTMAP. *IEEE Transactions on Biomedical Engineering*, 43(4):425–430, 1996.

[43] R. M. Haralick, K. Shanmugam, and I. Dinstein. Texture features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 3(6), 1973.

[44] F. Idris and S. Panchanathan. Review of image and video indexing techniques. *Journal of Visual Communication and Image Representation*, 8(2):146–166, 1997.

[45] A. Jain and A. Vailaya. Shape based retrieval: A case study with trademark image databases. *Pattern Recognition*, 31(9):1369–1390, 1998.

[46] F. Jing, M. Li, H.-J. Zhang, and B. Zhang. An effective region-based image retrieval framework. In *Proceedings of ACM Multimedia*, 2002.

[47] F. Jing, M. Li, L. Zhang, H.-J. Zhang, and B. Zhang. Learning in region-based image retrieval. In *Proceedings of International Conference on Image and Video Retrieval*, 2003.

[48] A. Khotanzan and Y. H. Hong. Invariant image recognition by zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:489–497, 1990.

[49] L. J. Latecki and R. Lakamper. Shape similarity measure based on correspondence of visual parts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1185–1190, 2000.

[50] L. J. Latecki, R. Lakamper, and U. Eckhardt. Shape descriptors for nonrigid shapes with a single closed contour. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2000.

[51] H. K. Lee and S. I. Yoo. A neural network-based image retrieval using nonlinear combination of heterogeneous features. *International Journal of Computational Intelligence and Applications*, 1(2):137–149, 2001.

[52] I.-J. Lin and S. Y. Kung. Coding and comparison of Dags as a novel neural structure with applications to online handwritten recognition. *IEEE Transactions on Signal Processing*, 45(11):2701–2708, 1997.

[53] W. Y. Ma and B. S. Manjunanth. Netra: A toolbox for nevigating large image databases. In *Proceedings of International Conference On Image Processing*, 1997.

[54] J. Malki, N. Boujemaa, C. Nastar, and A. Winter. Region queries without segmentation for image retrieval by content. In *Proceedings of 3rd International Conference on Visual Information Systems (Visual'99)*, 1999.

[55] B. S. Manjunath and W. Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):837–847, 1996.

[56] C. S. McCamy, H. Marcus, and J. G. Davidson. A color-rendition chart. *Journal of Applied Photographic Engineering*, 2(3), 1976.

[57] T. P. Minka and R. W. Picard. Interactive learning using a society of models. *Pattern Recognition*, 30(4):565–581, 1997.

[58] F. Mokhtarian, S. Abbasi, and J. Kittler. *Image Databases and Multimedia Search*. A. W. M. Smeulders and R. Jain ed., World Scientific Publication, 1997.

[59] B. Moore. ART 1 and pattern clustering. In *Proceedings of the 1988 Connectionist Models Summer School*, pages 174–185, 1989.

[60] S. Mukherjea, K. Hirata, and Y. Hara. Amore: A world wide web image retrieval engine. *The WWW Journal*, 2(3):115–132, 1999.

[61] C. Nastar, M. Mitschke, C. Meilhac, and N. Boujemaa. Surfimage: A flexible content-based image retrieval system. In *Proceedings of the 6th ACM International Conference on Multimedia*, pages 339–344, September 1998.

[62] V.E. Ogle and M. Stonebraker. Chabot: Retrieval from relational database of images. *Computer*, 28(9):40–48, 1995.

[63] M. Oren, C. Papageorgiou, P. Sinha, and E. Osuna. Pedestrian detection using wavelet templates. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR'97)*, 1997.

[64] M. Oren, C. Papageorgiou, P. Sinha, and E. Osuna. Pedestrian detection using wavelet templates. In *Computer vision and pattern recognition*, 1997.

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

[66] T. Poggio and Kah-Kay Sung. Finding human faces with a gaussian mixture distribution-based face model. In *Proceedings of Asian Conference on Computer Vision*, 1995.

[67] S. Rajasekaran and G. A. Vijayalakshmi Pai. Image recognition using simplified fuzzy ARTMAP augmented with a moment based feature extractor. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(8):1081–1095, 2000.

[68] H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, 1998.

[69] Y. Rui, T. S. Huang, and S. Chang. Image retrieval: Current techniques, promising directions, and open issues. *Journal of Visual Communication and Image Representation*, 10(4):39–62, 1999.

[70] Y. Rui, T. S. Huang, and S. Mehrotra. Content-based image retrieval with relevance feedback in MARS. In *Proceedings of IEEE International Conference on Image Processing*, 1997.

[71] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega. Automatic matching tool selection using relevance feedback in MARS. In *Proceedings of 2nd International Conference on Visual Information Systems*, 1997.

[72] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture in content-based multimedia information retrieval systems. In *Proceedings of IEEE Workshop on Content Based Access of Image and Video Libraries, in conjunction with IEEE CVPR'97*, 1997.

[73] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions On Circ. Sys. Video Tech*, September 1998.

[74] H. Schneiderman and T.Kanade. A statistical approach to 3d object recognition applied to faces and cars. In *IEEE Conference on Computer Vision and Pattern Recognition*, page 100, 2000.

[75] S. Sclaroff, L. Taycher, and M. La Cascia. ImageRover: A content-based image browser for the world wide web. In *Proceedings of IEEE Workshop on Content-based Access of Image and Video Libraries*, 1997.

[76] T. Serrano-Gotarredona, B. Linares-Barranco, and A. G. Andreou. *Adaptive Resonance Theory Microchips: Circuit Design Techniques*. Kluwer Academic Publishers, Boston, 1998.

[77] G. Sheikholeslami, W. Chang, and A. Zhang. Semquery: Semantic clustering and querying on heterogeneous features for visual data. *IEEE Transactions on Knowledge and Data Engineering*, 14(5):988–1002, 2002.

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

[79] J. R. Smith and S. F. Chang. Visually searching the web for the content. *IEEE Multimedia Magazine*, 4(3):12–20, 1997.

[80] J. R. Smith and S.F. Chang. VisualSEEK: A fully automated content-based image query system. In *Proceedings of ACM Multimedia 96*, 1996.

[81] P. Soliz and G. W. Donohoe. Adaptive resonance theory neural network for fundus image segmentation. In *Proceedings of the World Congress on Neural Networks (WCNN'96)*, pages 1180–1183, 1996.

[82] M. Soysal and A. A. Alatan. Combining MPEG-7 based visual experts for reaching semantics. In *Proceedings of 8th International Workshop on Visual Content Processing and Representation*, 2003.

[83] R. K. Srihari. Automatic indexing and content-based retrieval of captioned images. *Computer*, 28(9):49–56, 1995.

[84] W. D. Stromberg and T. G. Farr. A fourier-based textural feature extraction procedure. *IEEE Transactions on Geoscience and Remote Sensing*, 24(5):722–732, 1986.

[85] M. J. Swain, C. Frankel, and V. Athitsos. WebSeer: An image search engine for the world wide web. Technical Report TR-96-14, Computer Science Department, University of Chicago, 1996.

[86] H. Tamura, S. Mori, and T. Yamawaki. Texture features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, 8(6), 1978.

[87] M. Uysal and F. T. Yarman Vural. Selection of the best representative feature and membership assignment for content-based fuzzy image database. In *Proceedings of International Conference on Image and Video Retrieval (CIVR'2003)*, 2003.

[88] R. C. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Utrecht University, 2000.

[89] J. Wang, W.-J. Yang, and R. Acharya. Color clustering techniques for color-content-based image retrieval from image databases. In *Proceedings of IEEE Conference on Multimedia Computing and Systems*, 1997.

[90] 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.

[91] M. E. J. Wood, N. W. Campbell, and B. T. Thomas. Searching large image databases using radial-basis function neural networks. In *Proceedings of the Sixth International Conference on Image Processing and its Applications*, 1997.

[92] D. Yu and A. Zhang. Acq: An automatic clustering and querying approach for large image databases. In *Proceedings of ACM Multimedia*, 1999.

[93] A. Zhang, B. Cheng, and R. Acharya. A fractal-based clustering approach in large visual database systems. *The International Journal on Multimedia Tools and Applications*, 3(3):225–244, 1996.