

A Statistical Multi-Experts Approach to Image Classification and Segmentation

by


Lik Mui

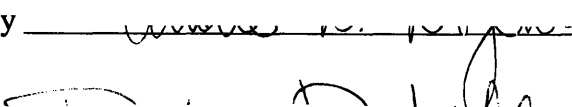
Submitted to the Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

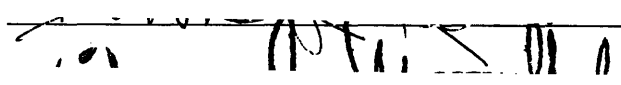
August, 1995

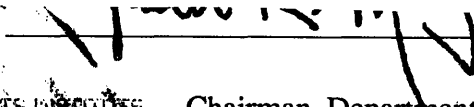
© Lik Mui, 1995. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce
and to distribute copies of this thesis document in whole or in part,
and to grant others the right to do so.

Author 
Department of Electrical Engineering and Computer Science
August 22, 1995

Certified by 
Andrew B. Dobrzeniecki
Thesis Supervisor

Certified by 
Darlene R. Ketten
Thesis Supervisor

Accepted by 
Frederick R. Morgenthaler
Chairman, Department Committee on Graduate Theses

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JAN 29 1996

Eng.

LIBRARIES

A STATISTICAL MULTI-EXPERTS APPROACH TO IMAGE CLASSIFICATION AND SEGMENTATION

by

Lik Mui

Submitted to the

Department of Electrical Engineering and Computer Science

August 22, 1995

In Partial Fulfillment of the Requirements for the Degrees of
Bachelor of Science in Electrical Engineering and Computer Science
and Master of Engineering in Electrical Engineering and Computer Science

Abstract

A real world image is a scene representation which captures many visual processes such as texture, shading, motion, etc. To accurately understand such an image, complex models usually have to be devised to account for all these different processes. For many machine vision systems, image segmentation (which can be considered as pixel classification) represents the first and perhaps the most important step in understanding an image because this procedure identifies the locations and types of objects in an image. Most of the current popular segmentation techniques address this problem from a single model viewpoint. This thesis presents an approach for image segmentation that considers the problem from a multiple models perspective. These models are supplied by different classification experts (or methods). The key problem that this thesis addresses is how to combine these different experts in a robust manner.

This thesis proposes two classifier designs for combining different methods (or experts) in the multi-experts approach. These two classifiers are based on two recently proposed statistical techniques. The first one is a gating network approach based on the work of Jordan and Jacobs (1994). The idea behind this approach is to weigh the outputs of different experts by appropriate priors. These priors are determined by a gating network which partitions the input space to allow the experts better suited for certain inputs to have higher weights for those inputs. At the same time, the experts which perform poorly on certain input patterns automatically receive lower weighting factors. The second proposed multi-experts classifier is based on Wolpert's (1992) idea of stacked generalization. The stacked generalizer makes generalization to yield the overall outputs by observing the predictions made by the classification experts.

Evaluations of these two classifiers are performed on synthetic as well as on real world images. Results from these evaluations are very encouraging for the multi-experts approach on many types of images. The multi-experts approach to image processing has many similarities to the distributed processing of the human visual system [Marr, 1982]. This approach naturally suggests a possible model for the computations in human vision.

Thesis Supervisors:

Dr. Andrew Dobrzeniecki
Research Scientist

Dr. Darlene Ketten
Assistant Professor, Harvard Medical School

Acknowledgements

I would like to thank my thesis advisors, Andrew Dobrzeniecki and Darlene Ketten, for all the encouragement and support throughout the past year while I have been working on my thesis. I am especially thankful for the freedom that they have granted me to define the direction of my thesis.

I am also very grateful to Michael Grunkin for many valuable conversations and advices on textures and politics. His comments on my thesis has greatly improved the presentation of this document.

To everyone in the Whitaker College Biomedical Image Computation Lab (WCBICL), I have to express my most sincere gratitude for making my stay in the lab very pleasant. Especially to Marie-Jose Belanger for reading many pages of my thesis and for supplying everything from the Visible Man images, to valuable cultural and political conversations, to advices on proper sleeping habits; to Bill Howard for providing a role model of a humble graduate student and a true conservative; to Jacqueline Yanch, the director of the lab, for welcoming me to her lab and many friendly conversations; to Ann Garvin for helping me with many secretarial tasks and for her dynamic presence.

Last but certainly not least, I thank my parents for their careful guidance, constant encouragement and support.

I sincerely hope that this thesis, inadequate as it is, would not disappoint all you amazing people.

CONTENTS

Abstract 2

Acknowledgements 3

Contents 4

1. Introduction 8

1.1 Image Classification and Segmentation Problems 10

1.2 Existing Approaches 12

1.3 Main Contributions of this Thesis 14

1.4 Overview 16

2. Background Materials 18

2.1 Applications of Image Classification and Segmentation 19

2.2 Images Used for Performance Evaluation 20

2.2.1 Synthetic Images 21

2.2.2 The Brodatz Album 25

2.2.3 Marine Mammalian Images 26

2.2.4 The NLM’s Visible Man 26

2.3 Neighborhoods and Vicinities 27

CONTENTS

2.4 Classification Methods	30
2.4.1 Feature-based Methods	30
2.4.2 Model-based Methods	34
2.4.3 Structural Methods	35
2.5 Segmentation Methods	36
2.5.1 Supervised Methods	36
2.5.2 Unsupervised Methods	40
2.6 Summary	42
3. The Theory of the Multi-Experts Approach	43
3.1 Introduction	44
3.2 Multi-Experts Classification	46
3.2.1 Overview of the MEC	47
3.2.2 A Probabilistic View of the Gating MEC	51
3.2.3 The Stacked Generalizer	53
3.3 Parameter Estimation for a MEC	54
3.3.1 Gating Network Parameter Estimation	54
3.3.2 Stacked Generalizer Parameter Estimation	57
3.3.3 Experts Parameter Estimation	57
3.4 Related Works	59
4. Model-based Classification Methods	61
4.1 Notations and Assumptions	62
4.2 Models Specifications	64
4.2.1 State Process	64
4.2.2 Intensity Process	67
4.3 Parameter Estimation	70
4.4 Some Experiments and Results	75
4.4.1 Creating a Phantom Image.	75
4.4.2 Maximum Likelihood Results	77
4.4.3 Maximum <i>A Posteriori</i> K-means Results	78
4.4.4 Maximum <i>A Posteriori</i> EM Results	79

5. Feature-based Classification Methods	81
5.1 Introduction	82
5.2 Features for Classification	83
5.2.1 First Order Statistics	84
5.2.2 Co-occurrence Matrix Features	84
5.2.3 Gaussian Markov Random Field Features	87
5.2.4 Simultaneous Auto-Regressive Features	90
5.3 Feature Selection and Classification	93
5.3.1 Selecting the Optimal Feature Set	94
5.3.2 Pixel Classification	96
5.3.3 Enhancing the Feature-based Classifier for MEC	98
5.4 Some Experiments and Results	100
5.4.1 Estimating Classifier Parameters	100
5.4.2 Image Classification and Segmentation Results	105
6. Other Classification Methods	115
6.1 K-Nearest Neighbors	115
6.1.1 KNN Density Estimation	116
6.1.2 KNN Spatial Estimation and Mrf Prior	116
6.2 Multilayer Perceptron Network	117
6.3 K-Means Based Classifier	119
6.3.1 Regularizing the K-Means Outputs	120
6.2 Probabilistic Co-Occurrence Classifier	120
7. Details of the Multi-Experts Classifier	122
7.1 Gating Network	122
7.1.1 Input Feature Sets	123
7.1.2 Network Architecture	124
7.2 Stacked Generalization	124
7.2.1 Inputs to a Stacked Generalizer	124
7.2.2 Selection of the REJECT Threshold	125
7.2.3 Post-Processing	126

8. Experiments and Results	127
8.1 Experimental Setups and Overview	128
8.2 Experimental Proof of Concepts: Quantitative Results	130
8.2.1 Problem Definition	130
8.2.2 Feature-based Classification and Scope	132
8.2.3 Segmentation by Other Experts	133
8.2.4 Gating Network Combination Methods and MEC Results	135
8.2.5 MEC Results Using Stacked Generalizers	139
8.2.6 Mrf State Phantom and Results	141
8.2.7 Comments on the Segmentation Results	149
8.3 Real World Image Segmentation	150
8.4 Discussions of Results	152
9. Conclusion	155
A. A Probabilistic Approach to Co-occurrence Matrix	159
B. The Incomplete Data Problem and Expectation Maximization	162
C. A Probabilistic Model for the Gating Network	164
D. Example of a Learning Set Input File	168
References	170

Chapter 1

INTRODUCTION

Image classification and segmentation are two important procedures in many image processing and machine vision systems. These tasks can be performed by human quickly, effortlessly, and in a robust fashion. Naturally, if researchers can find the “algorithms” that underlay the human visual system, image classifiers and segmenters would be able to perform just as well as human. For over thirty years now, much research has been devoted to find “solutions” to these two problems, but a technique with the capability approaching that of the human visual system is yet to be seen. However, many approaches have been found to achieve good results for various types of images.

This thesis is mainly concerned with developing multi-experts techniques for image segmentation. These segmentation techniques lend themselves naturally to image classification tasks. The underlying idea of this thesis is the multi-experts approach embedded in statistical framework. The performance of this proposed approach is evaluated using phantom images and real world images. Comparisons are made between the results achieved by the multi-experts approach and those achieved by commonly used techniques.

There are two main reasons for favoring the use of the multi-experts approach. First, segmenting real world images requires the use of methodologies that are robust for complex scenes. A real world image is usually the result of a variety of image formation processes such as illumination, texture, movement, etc. Most existing techniques approach image segmentation by assuming one prominent image formation process and ignoring the rest. These techniques usually operate well on images that are generated primarily by that process. However, their results would likely be miserable for images outside their expertise domains. When presented with real world images which are generated under multiple image formation processes, these techniques are likely not to have robust performance. If the goal of an image segmentation task is to obtain robust and quantitative results, these techniques are likely to be inadequate¹.

The multi-experts approach allows modeling of different processes in a single system. When a complex scene is presented for input, the Multi-Expert Classifier (MEC) (discussed in Chapter 3) automatically determines the likelihood that all the available *experts* can handle the input in a robust manner. In other words, the MEC determines the probability that an input pattern falls into the *expertise* area of each of the experts. This probability is assigned as the prior for combining the expert' outputs in deriving the final answer. Thus, the input space is partitioned, using *soft* boundaries², so that multiple segmenters can contribute to the final results using their respective *expertise*.

The second reason for using a multi-experts approach is that such an approach suggests a feasible model for the human visual system, especially with respect to scene segmentation. Vision researchers have found that the human visual system is a distributed system with many processes working together to yield higher level image understanding

¹ For some applications in which the image formation processes for the inputs are fairly predictable, such as in specific texture image analysis, algorithms that can handle different types of images might not be necessary. Nevertheless, one of the main goals of computer vision is to develop techniques that mimic the human visual ability, which definitely has this robust capability for handling multiple types of images.

² Soft boundaries are discussed in the first section of Chapter 3. It has been well known in information theory that soft boundaries are much more robust than hard boundaries. Essentially, soft boundaries allow a pattern to lie in multiple classes (or states), while hard boundaries restrict the membership to one class per pattern.

[Marr, 1982]. The distributed nature of the MEC, as will be discussed in Chapter 3, is a practical implementation of such a collaborative system for modeling interactions among different vision modules, of which the segmentation system is an important one. A real world object such as a television set usually has visible surfaces with different shading, textures, and other visual properties. Nevertheless, human observers can effortlessly identify the TV set as a whole. Clearly, some kind of mid-level vision or higher level processing must integrate the *features* extracted by simpler lower level modules to obtain the higher level understanding. In this light, the multi-experts framework discussed in this thesis provides practical *mid-level vision* schemes for combining lower level image segmentation results.

Section 1.1 considers the problems of image classification and segmentation. The following Section 1.2 discusses previous efforts in the fields related to this thesis. In section 1.3, contributions of this thesis are given in the context of current image classification and segmentation research. Section 1.4 gives an overview for the rest of the thesis.

1.1 Image Classification and Segmentation Problems

Image classification and segmentation are two very similar problems. Their distinctions are often blurred in many applications. Although this thesis is mainly concerned with image segmentation, techniques developed here for segmentation should easily be adapted for classification tasks. The following section first distinguishes the differences between these two procedures before discussing their similarities.

Strictly speaking, image classification is a recognition task, by which we mean an input image S is to be assigned a class label. For example, an image retrieval system is an example of an image classification task. The system attempts to *label* all the images in the database so that when a user enters an command such as “retrieve all images that look like this one on the screen”, the system can quickly retrieve appropriate images from its

database to match the user's demands. This is an image classification task, *not* an image segmentation task. Unlike text documents, images are difficult to label using vocabulary. A clear example of this difficulty can be seen through the above command: "images that look like this". Sometimes, the users do not even know what "this" (set of attributes) that characterizes an image really is ³. The challenge to an intelligent image retrieval system is to guess at the right set of attributes that the users want by using the contents of the image such as texture, brightness, entropy, etc. Recently, several researchers have attempted to tackle this problem and have achieved encouraging results. [Niblack, *et.al.*, 1993; Pentland, *et.al.*, 1993; Picard and Minka, 1995].

Many examples of the image classification task exist in the medical domain [Finette, *et.al.*, 1983; Garra, *et.al.*, 1989; Momenan, *et.al.*, 1994]. Image classification of medical images is not only a time saving procedure, it often outperforms human experts in identifying diagnostic indicators such as breast tissue calcification, abnormal tissue growth, tumor metastasis, among others. In an expensive health care environment such as ours today, these automatic procedures can cut costs while maintaining the quality of care.

Just like image classification, image segmentation also involves recognition, but generally at a much smaller scale than that of image classification. Usually, the scale of concern to this thesis, as well as to other image segmentation applications, is pixel-size⁴. The goal here is to group pixels into *homogeneous* regions (where the characteristics of homogeneity is context dependent). For example, given a chest X-ray, an image segmentation algorithm could aim to find the size of the lung [Duryea, *et.al.*, 1995]. Homogeneity in this case is the region associated with the lung on an X-ray film. Given some aerial photos, such as the LANDSAT images, the goal is to find the size of certain crop lands. Homogeneity is the brightness and textural properties of the crop lands.

³ For example, when one sees several variations of "brick wall", most people would not have the necessary knowledge and sophistication to tell the difference among them. Another example, how can one describe the "feeling" of festivity in a Christmas shot of the downtown? How about the empathy one has when a shot of an African famine is shown?

⁴ Sub-pixel size image segmentation problems and techniques also exist.

Clearly, the types of images in these two cases are very different. Even within a single image, such as in the later LANDSAT case, the region corresponding to forest could have very different image properties as opposed to that of the ocean. To accurately assign pixels to different classes in these real world images, a segmenter has to take into consideration different image features, and possibly different models. A point on language, since image segmentation can be thought of as pixel classification, and many segmentation methods can be applied for classification purposes, the usage of image “classification” and “segmentation” are sometimes interchanged through the rest of this thesis. “Classification” in this thesis refers to pixel classification, which involves classifying a sub-region surrounding the center pixel.

1.2 Existing Approaches

Most image classification approaches can be categorized into model-based, feature-based, or structural based methods. These methods are discussed in the background chapter, Chapter 2. In classifying pixels, these approaches provide the basis for many image segmentation algorithms. Usually, these algorithms assume a single (arbitrarily complex) model for the image. They might perform very well on those images generated by the image model they model after. However, most real world images are highly non-stationary and in general cannot easily be modeled by a single model alone. For example, a camera shot of a mountain during sunset would likely to have many different regions with different brightness, textures, and shadings. A magnetic resonance imaging (MRI) slice of the human inner ear region is another example. Such an image has different tissues and bones regions with different sizes, textural properties, brightness, and boundary distinctness. Accurate segmentation of these real world images is almost hopeless with a single model approach. Clearly, a multiple models approach would be much more appropriate for such real world images. The key problem is then, how can these different models be combined?

For many years, statisticians have also been trying to solve a similar problem. Many of them have pointed out that using a single model to make predictions and inferences, no matter how elaborate, is not optimal [Howard, 1970; Self and Cheeseman, 1987; Kwok and Carter, 1990]. Several attempts have been made to average over many prediction models using decision trees [Self and Cheeseman, 1987; Buntine, 1989; Kwok and Carter, 1990]. The general result is that by averaging over different models, better results are obtained than results from using any single model.

In the machine vision community, the researchers working on the Photobook project in MIT for image database retrieval have recently proposed a multiple texture models system for annotating images [Pentland, *et.al.*, 1993; Picard and Minka, 1994]. Rather than using one elaborate texture model, their system chooses from a set of available models the one that “best explains” an input image region. This approach corresponds to making a hard decision in selecting the right model, and using that model to make predictions. In the words of the statistician Breiman, such strategies is like “wearing the less worn of two old suits” [Breiman, 1992]. As Breiman has shown in his 1992 paper, such an approach is inferior to those approaches that take all models into considerations when making decisions.

Wolpert’s *stacked generalization* was proposed in 1992 to reduce the generalization error of a single or multiple generalizers [Wolpert, 1992]. A *generalizer* is learning system that makes inferences or predictions based on learning done using a separate learning set. The basic idea behind this approach can be illustrated by an example. If a city C’s weather is always sunny or cloudy, and if meteorologist A can correctly predict the weather 90% of the time while meteorologist B always gets the prediction wrong, which meteorologist gives a better weather prediction at any given day? Clearly, whenever meteorologist B predicts sunny, the weather would be cloudy and vice versa. In fact, B’s prediction, if *reversed*, is 100% correct. Learning from the outputs of predictors when given certain input patterns, instead of directly learning from those input patterns is called *first level* learning [Wolpert, 1992]. Stacked generalization is uses this learning

approach for making inferences or predictions. This thesis takes stacked generalization and applies it to image segmentation. Discussions of our approach are given in Chapter 3.

Stacked generalization is a serial approach for combining different models in making decisions. An alternate approach is a parallel approach that combines all these models' outputs by some weighting factors. This is the approach taken by Jordan and Jacobs in their Hierarchical Mixtures of Experts (HME) architecture [Jordan and Jacobs, 1994]. This thesis applies this parallel approach by using a *gating network*, which is a learning system that determines the priors for individual models' outputs when presented with a given input. The overall decision is a weighted sum of all the models' outputs. The learning of this gating network is accomplished through a supervised learning scheme derived in Chapter 3.

1.3 Main Contributions of this Thesis

To the best knowledge of the author, this thesis is the first piece of work that applies the multi-experts approach in a statistical framework to image segmentation. These techniques can also be adapted for image classification tasks. Real world images are often results of different image formation processes. To analyze these images using a single model is extremely difficult. The multi-experts framework advocated by this thesis provides a possible solution to this difficulty.

This thesis considers two techniques for combining *expert* knowledge -- the gating network method and the stacked generalization method. Comparisons are made on the effectiveness of these two *higher level* generalizers⁵. What is meant by *higher level* learning is that these two methods learn to generalize based on information provided by

⁵ We have borrowd the term *generalizer* from [Wolpert, 1992]. A *generalizer* is essentially any system that can make predictions or inferences based on learning performed on a learning set of patterns.

other lower level processes, such as motion, texture, shading, and contours. In the context of this thesis, these lower level generalizers are the individual image segmentation methods. Implemented in a vision system, the gating network and the stacked generalizer can be considered as *mid-level* visual processors that combine information from lower level processes.

The structure of the Multi-Experts Classifier (MEC) presented in Chapter 3 is very general. It can be applied to many other types of vision or pattern recognition tasks. The MEC approach suggests a possible model for the human visual system. Vision scientists have recognized that the complex human vision processing has to be distributed over a series of stages [Marr, 1982]. Initial processing is handled by several different specialized processes that provide information about visual properties such as texture, motion, shading, and brightness of an input scene. The information provided by these lower level vision modules has to be combined appropriately in order to yield meaningful higher level understanding. The multi-experts approach advocated by this thesis provides a good framework for modeling the serial and well as the parallel nature of the human visual system.

In comparing the performance of the MEC with traditional techniques, this thesis has also evaluated the performances of several commonly used image segmentation methods for different types of images. These methods include both supervised and unsupervised ones, which consist of K-nearest neighbors (KNN), multilayer perceptron (MLP), K-means clustering, mixtures of Gaussians with parameters estimated by expectation maximization, Lohmann's approach (1995) to co-occurrence matrices, and various feature-based approaches. In addition, different types of features have been evaluated, which include first and second order statistics, co-occurrence matrix based features, Simultaneous Autoregressive features (SAR) features, and Markov random field (MRF) features.

1.4 Overview

The general flow of this thesis is from theory to experiments to results. The beginning chapters are mainly concerned with the theoretical aspects of traditional methods and the multi-experts approach. Experiments and results are presented later. Many of the traditional methods can potentially be incorporated into the Multi-Experts Classifier (MEC) construction -- as the *experts*. Therefore, extensions of some of these methods are suggested so as to conform to the MEC architecture.

The main ideas of this thesis are presented in Chapter 3. This chapter describes the theory of the MEC construction. It includes discussions on both the stacked generalization approach and the gating network approach. Further discussions on how the MEC is related to other works are also given in this chapter.

To understand the difficulty with many traditional methods in performing image segmentation, this thesis spends several chapters discussing their theories and evaluating their performances. The first type of such methods is discussed in Chapter 4, which is about model-based techniques. Falling under this category includes Markov random field methods, auto-regressive methods, and density estimation methods. Chapter 4 also presents some of the recent development in parameter estimations for the models modeled by these methods. Specifically, this chapter discusses a recently developed technique for robust parameter estimation based on expectation maximization (EM) [Kashyap and Chellappa, 1983; Geman and Geman, 1984; Derin, *et.al.*, 1986; Besag, 1986; Liang, *et.al.*, 1994; Zhang, *et.al.*, 1994].

In Chapter 5, we consider feature-based approaches. As will be discussed in that chapter, almost all existing classification and segmentation algorithms can be considered in some way to be “feature-based”. What differentiates the feature-based classifiers discussed in Chapter 5 from all of the other methods is the emphasis that these classifiers have on choosing the *optimal* feature set for classification. A feature set can be used for other purposes such as for simulating images, or for restoring degraded images [Hassner and

Sklansky, 1980; Cross and Jain, 1983; Geman and Geman, 1984]. In fact, by visually evaluating the simulated images, one can check the robustness of the feature parameter estimation routines [Cross and Jain, 1983; Lohmann, 1994]. This chapter also describes extensions proposed for augmenting traditional feature-based classifiers to become *expert* classifiers for the MEC. Some experiments and results are reported for these methods.

The following chapter, Chapter 6, is a brief overview of other relevant methods. Some of these methods actually fall under the model-based or feature-based techniques, but are placed in this chapter because their approaches differ considerably from the more traditional model-based or feature-based techniques considered in Chapter 4 and 5. These methods include K-Nearest Neighbors, a novel probabilistic co-occurrence approach, a K-means clustering based classifier, and a neural network approach.

Some of the details of the MEC implementation are described in Chapter 7. In Chapter 8, experiments and results on applying the MEC to segmenting images are reported. Image classification and segmentation are performed on a variety of images, including both synthetic and real world images. Chapter 9 concludes the thesis with suggestions on future works using the multi-experts approach.

Chapter 2

BACKGROUND

There has been an enormous amount of research in both image classification and image segmentation in the past thirty years. The breath and depth of this research makes detailing all published techniques very difficult. More methods and descriptions of these methods can be found in the many review articles already written, such as [Weszka, *et.al.*, 1976; Haralick., 1979, 1985; Ohanian, *et.al.*, 1992; Bezdek, *et.al.*, 1993; Reed, *et.al.*, 1993]. This chapter only intends to touch on those concepts and techniques relevant to this thesis. This chapter also establishes some of the notations used throughout the thesis. In addition, important background concepts are reviewed. Finally, brief descriptions are given on the types of images used for testing and evaluating different classification and segmentation techniques.

The first section gives some examples of how image classification and segmentation are used in practice. Image classification and segmentation tasks permeate many aspects of everyday life. This section aims to provide a glimpse of the possibilities. In Section 2.2, the types of images used in this thesis for evaluating segmentation results are described. Both synthetic and real world images are used. This section also describes the methods used for generating the synthetic images. Concepts of neighborhoods and vicinities are discussed next, which include discussions on the notations and basic concepts used for the rest of this thesis. In the following section, relevant techniques for image

classification and segmentation are then reviewed in the context of this thesis. Finally, a brief summary concludes this chapter.

2.1 Applications of Image Classification and Segmentation

There are many applications of image classification and segmentation techniques. The application area ranges from the medical, military, to scientific domains. Their importance in any image understanding task is great because these two techniques answer two central questions about any vision system. These two questions are: what objects are in an input image *and* where are they?.

Robot vision is perhaps the first application that comes to most people's minds when they think of image segmentation. Recently, auto-pilot cars are being tested in several research labs, most famous of which is the ALVINN in Carnegie Mellon [Pomerleau, 1989]. Navigating these auto-pilot cars requires much image classification and segmentation techniques.

In the medical community, image classification and segmentation methods are used on a day to day basis for various diagnostic purposes. For example, in planning surgeries, modern surgeons often need to review CT and MRI data sets, usually with segmented parts, to help them plan their operations accurately. For radiologists and oncologists, scanning mammograms for breast cancer are aided by image classification and segmentation techniques [Dengler, *et.al.*, 1994; Kilday, *et.al.*, 1994].

Before the Berlin Wall fell, and before the collapse of the Soviet Union, image classification and segmentation techniques developed for defense purposes were highly guarded secrets, some of which are probably still so today. For quick and robust object identification of tanks or nuclear warheads, or for analyzing landscapes from aerial photographs or satellite images, image classification and segmentation techniques are indispensable. During peaceful times, these techniques are still extremely useful for

identifying wildfires, typhoons, tornadoes, and other natural phenomenon so that people can be warned of any potential danger before the danger approaches.

This section considers one final application of image classification and segmentation techniques. Marine mammals are known to have very good underwater hearing capability. However, the exact mechanism of their great auditory competence over a wide range of sound frequency have eluded many biologists and hearing experts for many years [Ketten, *et.al.*, 1992]. Currently, research efforts are underway to use image classification and segmentation techniques to understand how sound is transmitted to the inner ears of these animals through 3-D imaging studies of their auditory systems. The proposed channel of transduction for marine mammals is a fatty tissue channel which requires much more sophisticated techniques to detect than the commonly used intensity thresholding for segmentation. This difficulty is caused by the extensive intensity overlap among neighboring tissues.

The main point of this section is that currently, there exists many applications for image classification and segmentation. The need is going to be even greater in the future. In the medical field, for example, as nations are trying to cut health care spending and upgrade their medical diagnostic capabilities, image classification and segmentation techniques will be in increasing demand for providing fast, accurate, and inexpensive means for analyzing and interpreting images.

2.2 Images Used for Performance Evaluation

This section describes the four kinds of images used for testing and evaluating the performance of various image classification and segmentation techniques. Both synthetic images and real world images are included. The real world images come from a variety of sources, which include the Brodatz texture album [Brodatz, 1963], marine mammalian images [Ketten, *et.al.*, 1992], and the National Health Institute (NIH)'s Visible Man images.

2.2.1 Synthetic Images

Three types of synthetic images are generated for quantitative evaluation of different image segmentation techniques. These synthetic images are based on three different assumptions on the image formation process. These assumptions are: Gaussian distributed intensities, Auto-regressive model based intensities, and Markov random field based intensities. This subsection briefly describes the steps that have been used to generate these images.

(i) *Gaussian distributed intensities*

Gaussian density is a common intensity characteristics of many images. An example of a synthetic image produced by a Gaussian intensity process is shown in Figure 2.1. The intensity values are generated using the following familiar equation for a Gaussian density [Fukunaga, 1990]:

$$f(y_i) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{\{-(y_i - \mu_k)^2 / (2\sigma_k^2)\}} \quad (2.1)$$

where μ_k and σ_k are the mean and standard deviation of region k ; y_i is the intensity value at pixel i . For all four quadrants in Figure 2.1, different means and variances are used to

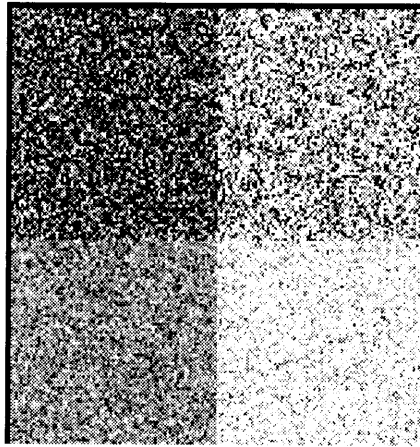


Figure 2.1 An example of a synthetic image with Gaussian distributed intensity values. The means and variances of the four quadrants are shown in Table 2.1.

generate those regions. For this figure, the means and standard deviations of the intensity values are shown below in Table 2.1.

Region Location	Top-left	Top-right	Bottom-left	Bottom-right
Mean (μ)	120.09	139.80	159.97	179.87
Std. Dev. (σ)	40.63	20.42	50.67	30.19

Table 2.1 The mean and standard deviation of the intensity values in the Gaussian mosaic image 1 of Figure 2.1. Note the large overlap among the different regions.

(ii) Auto-regressive process based images

In a finite image lattice S , the intensity values generated by an AR process can be described by the following generative equation for all intensity values $\{y_i, i \in S\}$ [Kashyap and Chellappa, 1983]:

$$y_i = \sum_{j \in \eta_i} d_j y_j + \sqrt{\rho} w(i) \quad (2.2)$$

where ρ and d_j 's are unique parameters for each AR textural images. η_i is the local neighbors of pixel i whose intensity is y_i . Chapter 5 will discuss in detail the image generation process. Figure 2.2 shows several simulated images by applying this equation with a four neighbors non-symmetrical half plane (NSHP) model. (The NSHP model will also be explained in a later section -- see Figure 5.4.) [Lim, 1990; Grunkin, 1992].

(iii) Markov random field process based images

MRF has been used successfully for modeling both intensity and state processes of various types of images [Hassner and Sklansky, 1980; Jain and Cross, 1983; Geman and Geman, 1984; Besag, 1986; Zhang, *et.al.*, 1994]. The intensity generation process using

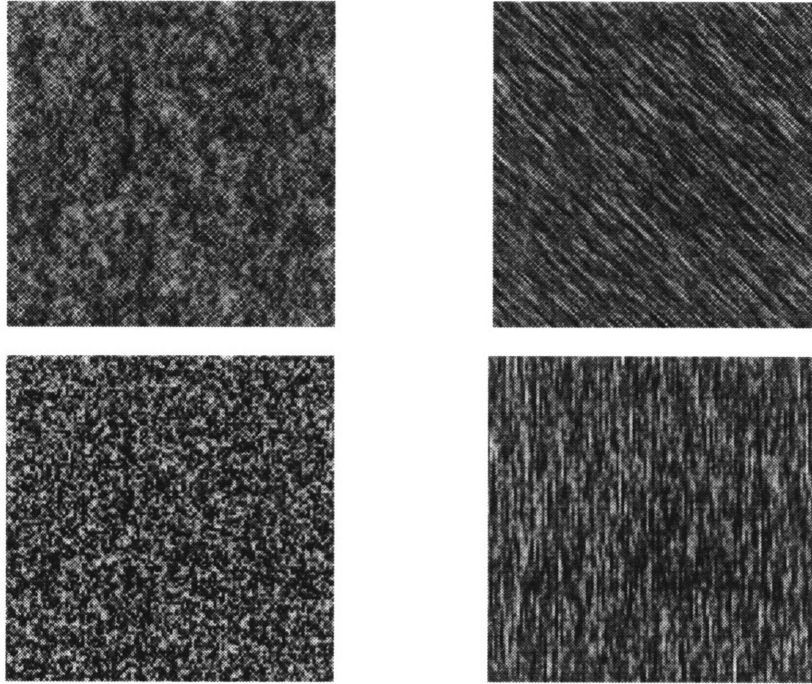


Figure 2.2 Synthetic images simulated using AR generative equation (2.2). The parameters of these generative processes will be discussed in Chapter 5.

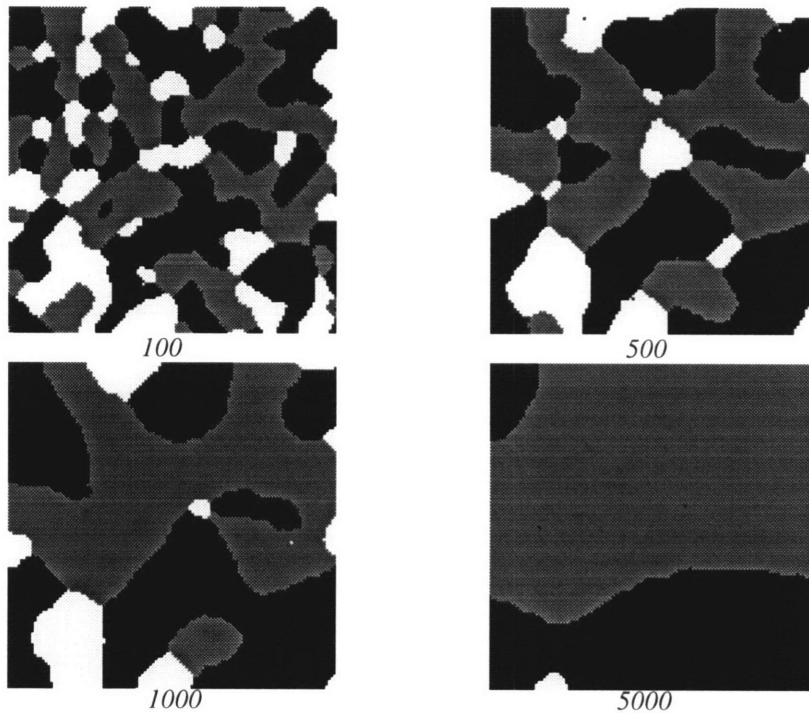


Figure 2.3 Synthetic three-level Markov random field images generated by a Gibbs sampler with a $\beta = 1.5$. The number below each figure represents the number of iterations that the Gibbs sampler underwent. Notice that although a MRF used here is a local field, it has long range interactions -- if the iteration continues, the entire map would be of a constant color.

the so-called Gaussian Markov random field (GMRF) is discussed in detail in Chapter 5. This section considers a general three-level MRF process. An image generated by a MRF satisfies the following Markov condition:

$$\textbf{Markov condition: } f(z_i|z_j, j \neq i) = f(z_i|z_j, j \in \eta_i). \quad (2.3)$$

where η_i is the local neighborhood of pixel i with state z_i . MRF turns out to be the right conditional structure to model any images provided that an arbitrarily large MRF is used [Besag, 1972; Geman and Geman, 1984]. For most images, a second order MRF with a local characteristics determined by a pixel's eight nearest neighbors suffices (refer to Chapter 5 for more discussion on neighborhoods).

A MRF is related to a statistical mechanics energy called the Gibbs energy [Geman and Geman, 1986]. Most researchers these days apply MRF modeling using the Gibbs energy framework rather than directly with MRF because parameter estimation of the MRF coefficients turns out to be computationally demanding [Hassner and Sklansky, 1980; Cross and Jain, 1983]. In dealing with the Gibbs energy equivalent expression for a MRF, the implementation of algorithms using MRF becomes much more lucid and requires less computational resources. This nice relationship between a MRF and a Gibbs energy is made possible through a famous theorem proved by Hammersley and Clifford [Besag, 1972]. Because of the general applicability of MRF to the intensities and the states (or classes) of images, this theorem is explained in detail in a separate section (2.3). This theorem will be used several times throughout this thesis.

Figure 2.3 shows several examples of MRF with three states. These images are generated using a Gibbs sampler [Geman and Geman, 1986] with a β value of 1.5 (refer to Chapter 5 for discussion of β). The number of iterations used for each simulation is indicated below each figure. Unlike Monte Carlo which uniformly samples from the configuration space, a Gibbs sampler samples values from the local characteristics. This locally dependent sampling idea was first used in the Metropolis algorithm for studying the equilibrium states of dynamical systems [Metropolis, *et.al.*, 1953]. This sampling technique is much more efficient than uniform sampling because the state distribution of

interest (Gibbs distribution, see equation 2.5) tends to have most of its *energy* near the most likely states. As indicated by Figure 2.3, although the MRF used is a second order locally dependent field, the effect of such a field is long range -- the tendency shown in Figure 2.3 is toward uniformity.

2.2.2 Brodatz Album

The Brodatz album is a collection of 112 digitized real world textural images that has become a standard set of images for performance comparisons [Brodatz, 1966]¹. The images were scanned with an 8-bit, 300 dpi scanner. The output is a 2400 x 1800 pixels image, which is then reduced by a two steps Gaussian pyramid procedure to 600 x 450. Some example Brodatz images are shown in Figure 2.4.

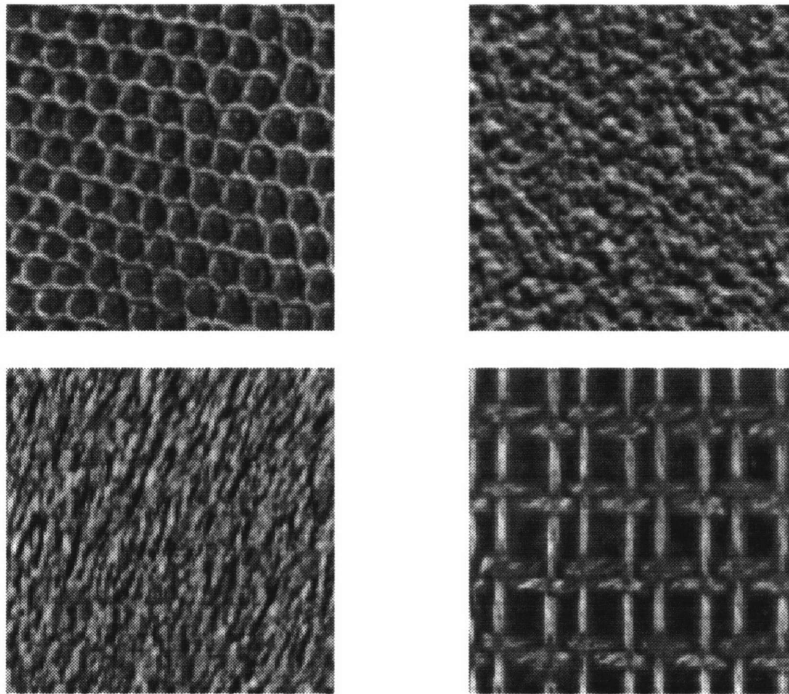


Figure 2.4 Example images from the Brodatz Album. The upper left hand image is D3, reptile skin; the upper right hand image is D57, handmade paper; the lower left hand image is D93, fur; the lower right hand image is D20, French canvas.

¹ The set of digitized Brodatz images used were graciously provided by Professor Michael Grunkin of the Institute of Mathematical Statistics and Operations Research (IMSOR), Technical University of Denmark.

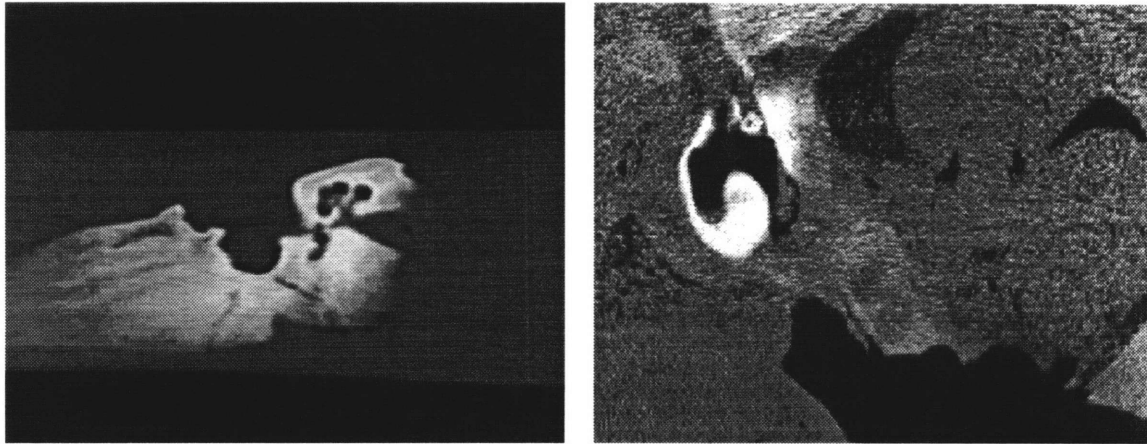


Figure 2.5 Sample images from the marine mammalian image set. The left hand side is a CT slice of a seal temporal bone immersed in a beaker of solution. The right hand side is another slice of a grey whale auditory system (middle ear and inner ear region).

2.2.3 Marine Mammalian Images

Two sets of marine mammalian images are used in this thesis. The first set is an X-ray CT scan of a seal temporal bone isolated in a beaker of solution. The second set belongs to a grey whale CT head scan. An example of each of these two sets is shown in Figure 2.5. These images are acquired for the purpose of studying the marine mammals' unique auditory capability [Ketten, *et.al.*, 1992]. The goal is to understand how marine mammals such as seals and grey whales can listen in various frequencies.

2.2.4 NLM's Visible Man

The Visible Man (VM) is the first of a series of ambitious projects initiated by the United States National Library of Medicine (NLM) in 1989. The goal is to create a digital atlas of the human anatomy [Lorensen, 1994]. The VM data set is composed of three types of whole body images -- X-ray CT, MRI, and histology scans. This thesis considers only several slices from the CT data set, although plans are underway to process some of the MRI and histology scans. An example image from the VM is shown in Figure 2.6. That image is a head CT slice showing part of the external auditory canal and the middle ear region.

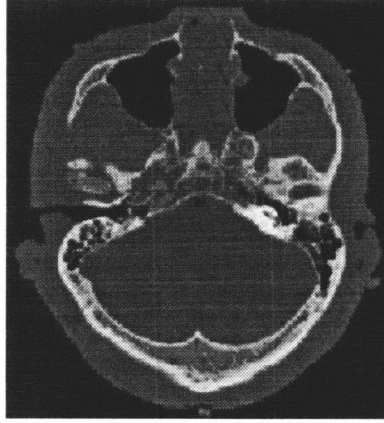


Figure 2.6 An example head slice from the Visible Man fresh CT data set. The main area of interest for this thesis is areas related to the auditory system, such as the external auditory canals, the middle ear, and inner ear.

2.3 Neighborhoods and Vicinities

This thesis considers different spatial constraints in classification and segmentation using different neighborhood and vicinity systems. In this section, definitions of neighborhood and vicinity systems are given. The important Hammersly-Clifford Theorem concerning spatial systems described by a MRF is also discussed.

Consider an image lattice $S = \{s_1, s_2, \dots, s_N\}$, where there are N lattice points, the neighborhood of each of the lattice point s_i is denoted by η_i , where $\eta = \{\eta_1, \eta_2, \dots, \eta_N\}$ is the whole neighborhood set of the image. A neighborhood system for pixel i satisfies the following conditions:

Conditions for neighborhood

$$\eta_i \text{ of pixel } i: \begin{cases} s_i \notin \eta_i \\ \text{if } s_i \in \eta_j, \text{ then } s_j \in \eta_i \end{cases} \quad (2.4)$$

The simplest explanations for what a neighborhood system is probably through figures². Figure 2.7 shows three types of neighborhood, a first order neighborhood with 4 neighbors, a second order neighborhood with 8 neighbors, and a third order neighborhood

² We consider only the lattice points neighborhoods and refer the readers to other literatures on other types of neighborhood [Geman and Geman, 1984].

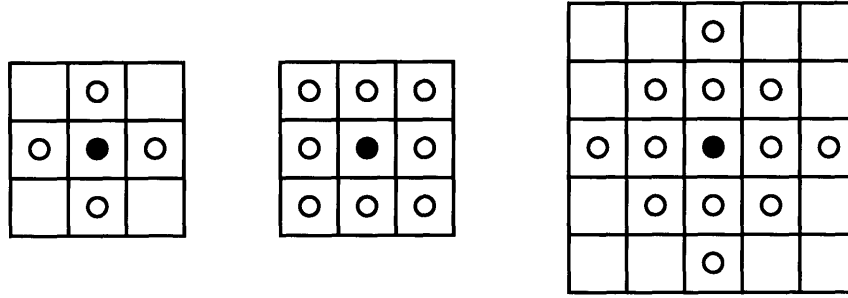
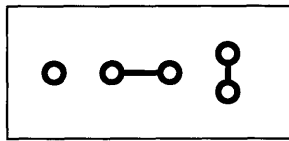


Figure 2. 7 Symbolic diagram of several neighborhood systems. The one on the left is the familiar nearest neighbor system (or first order neighborhood). The middle one is a second order neighborhood. The one on the right is the third order neighborhood.

1st Order Cliques



2nd Order Cliques

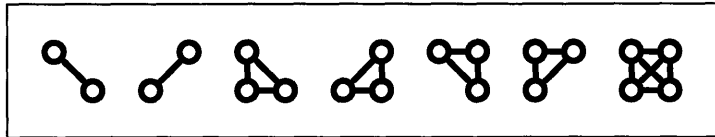


Figure 2. 8 Illustrations of first and second order cliques. Every pixel in a clique is a neighbor to every other one in the same clique.

with 12 neighbors. (The first order system is also known as the nearest neighbor system.) In a group of pixels, if every pixel in that group is a neighbor to each other, then this group is called a *clique*, C . Shown in Figure 2.8 are examples of cliques associated with first and second order neighborhood systems.

The concept of vicinities is used later in Chapter 4 to describe the intensity process of an image. The definition of a vicinity is almost exactly the same as that of a neighborhood, with the extension of membership to the center pixel also. For example, a first order vicinity v_i for a given pixel i has five members, the four nearest neighbors of pixel i , plus the pixel i itself. A second order vicinity has nine members, and so on.

A final preliminary concept before the discussion of the Hammersley-Clifford Theorem is the Gibbs distribution, which is a statistical mechanics concept related to the

energy of a system. Let U be defined as the system energy function, then the Gibbs distribution of the energy is expressed as:

$$f(\omega) = \frac{1}{Z_n} e^{-U(\omega)/T} \quad (2.5)$$

where Z_n is the partition function that normalizes the function (2.5) to sum to one over all ω 's. ω represents a particular configuration of states in the system. For example, in a greyscale image of with 256 greylevels, the space of all possible images is $\Omega = \{y_1, y_2, \dots, y_N\}$ where every y_i can take 256 different values. ω is a particular instance of Ω . Discussions of the energy function $U(\omega)$ is deferred to Chapter 4, where several variations of $U(\omega)$ are discussed. There is one more quantity -- the temperature T -- in (2.5) that needs to be explained. T shapes the Gibbs distribution by either making the probability distribution (2.5) more concentrated near the most likely configuration ω or spreading the distribution more evenly over all configurations. To make the likelihood of sampling a configuration near the mode of the Gibbs density higher, the system temperature has to be lowered -- this is the process of *simulated annealing* [Geman and Geman, 1986]. Conversely, to spread the distribution, T has to be raised. In a statistical mechanics system, higher temperature means that the particles have higher mean velocities and tend not to settle down to any particular energy state as easily as a lower temperature system. By controlling T appropriately, the lowest energy states can be isolated within finite iterations. *i.e.*, the most likely state (with the lowest possible energy) can be found.

A lucid form of the Hammersley-Clifford Theorem can be stated as follows [Besag, 1972; Geman and Geman, 1984]:

Hammersley-Clifford Theorem: If η is a neighborhood system, then ω is a MRF with respect to η if and only if $f(\omega)$ is a Gibbs distribution of the form 2.5.

This above theorem makes calculation of the joint distribution $f(\omega)$ fairly easy to compute. With $f(\omega)$, modeling images with MRF becomes quite straightforward, as will be done in Chapter 4.

2.4 Classification Methods

There are three main groups of image classification methods: feature-based methods, model-based methods, and structural methods. Sub-section 2.4.1 discusses the commonly used features for feature-based classification. Feature-based methods group pixels with similar features together and separate pixels with dissimilar features. Model-based methods estimate the underlying models of an image and use these models for pixel classification. Model-based methods can be considered a subset of feature-based methods since classification is performed over the model parameters³, which can also be considered as features. Nevertheless, model-based methods emphasize the modeling part much more than the feature extraction part of classification. Structural methods assume a primitive which is repeated throughout an image. Classification is performed by identifying the right primitive for an image. These methods are only suitable for images with repeating patterns.

The meaning of classification in this section is pixel classification, which is the first step in most image segmentation procedures. Further segmentation rules are applied to this initial segmentation. ML and other procedures are discussed in section 2.5.

2.4.1 Feature-based Methods

The most important aspect in a feature-based technique is obviously the feature set that is used for classification. This sub-section describes a set of commonly used features that have been found to be useful in many applications. One way these features can be used for pixel classification is by simply grouping pixels with similar features. This technique, as will be discussed in Section 2.5, is like a maximum likelihood (ML) region-based method since pixels are grouped according to how likely their feature sets can be

³ Sometimes, the features are transformed to other domain for classification purposes.

described by the prototype feature sets of different classes. (A contrasting viewpoint to classify pixels by separating the pixels with dissimilar feature sets. This corresponds to boundary-based segmentation method discussed in 2.5.)

(i) first order intensity statistics

First order statistics include mean, variance, skewness, kurtosis, and other histogram based features [Lim, 1990; Gonzalez and Wood, 1992; Pitas, 1993]. Although very useful for simple images, these features have been found by the author to be fairly ineffective for natural textures, complex scenes, and other “not-so-simple” images. Below is a list of the commonly used first order features:

$$\begin{aligned}
 \textbf{Mean:} \quad \mu &= \sum_y y p(y) & \textbf{Variance:} \quad \sigma^2 &= \sum_y (y - \mu)^2 p(y) \\
 \textbf{Entropy:} \quad s &= -\sum_y p(y) \log p(y) & \textbf{Energy:} \quad E &= \sum_y (p(y))^2 \\
 \textbf{Skewness:} \quad \mu_3 &= \frac{1}{\sigma^3} \sum_y (y - \mu)^3 p(y) \\
 \textbf{Kurtosis:} \quad \mu_4 &= \frac{1}{\sigma^4} \sum_y (y - \mu)^4 p(y) - 3
 \end{aligned} \tag{2.6}$$

where $p(y)$ is the normalized image histogram, which sums up to 1.0 for all y values. These features represent the most commonly used of the first order statistics.

(ii) co-occurrence matrix features

Since early 1970's, researchers have been using the co-occurrence matrices and the related gray-level difference statistics for image classification and segmentation, especially for textural images. Review articles [Haralick, 1979] and [Weszka, *et.al.*, 1976] give fairly extensive accounts of various attempts. Briefly, a co-occurrence matrix of an image is defined for a given displacement d and a given orientation angle θ . The usual choices of (d, θ) are : $(1, 0^\circ)$, $(1, 45^\circ)$, $(1, 90^\circ)$, and $(1, 135^\circ)$. Assume that a greylevel image has m greylevels. Each choice of (d, θ) results in a co-occurrence matrix of size $m \times m$. For a

usual image with 256 greylevels, the sheer sizes of the co-occurrence matrices prohibit one from using their coefficients as features for classification -- this feature space is of extremely high dimension. Usually, one derives much smaller dimensional features from the co-occurrence matrices and, at the same time, attempts to capture as much of the original image information as possible.

A few of the often used features from a normalized co-occurrence matrix are (the co-occurrence matrix coefficients satisfy the conditions $0.0 \leq c_{ij} \leq 1.0$ and $0 \leq i, j \leq m$, where m is the number of greylevels):

$$\begin{aligned}
 \text{Entropy:} & \quad -\sum_{i,j} c_{ij} \log(c_{ij}) & \text{Max. probability:} & \quad \max_{i,j}(c_{ij}) \\
 m^{\text{th-order contrast:}} & & \sum_{i,j} |i-j|^m (c_{ij}) & \\
 m^{\text{th-order inverse difference:}} & & \sum_{i,j} \frac{c_{ij}}{|i-j|^m} & \\
 \text{Correlation:} & & \sum_{i,j} \frac{(i-\mu)(j-\mu)c_{ij}}{\sigma^2} & \\
 & & (2.7) &
 \end{aligned}$$

These derived features reduce the information in the original texture to several numbers. Desirably, they would contain the necessary information for classifying a given region to the right class. Unfortunately, they often fail to retain all the information originally in the co-occurrence matrices. Nevertheless, co-occurrences matrix based features often rank among the best textural features available [Weszka, *et.al.*, 1976; Haralick, 1979; Ohanian, *et.al.*, 1994]. Recently, Lohmann (1994) proposed a new approach to extraction features from the co-occurrence matrices. These features seem to retain all the information in the original co-occurrence matrices. We discuss his novel method in Chapter 7 as well as in Appendix A.

(iii) grey-level difference and grey-level sum histograms

For a displacement $\mathbf{d}=(d1, d2)$, the grey-difference between two image pixels is defined as follows [Pitas, 1993]:

$$D(\mathbf{d}) = |y_{i,j} - y_{i+d1,j+d2}| \quad (2.8)$$

Grey-level difference histogram (GLDH) is essentially computed by summing over a co-occurrence matrix over constant grey-level difference $|m-n|$. As Carstensen (1992) has pointed out, greylevel difference histogram is a measure of the “distance” to the co-occurrence matrix diagonal. So, GLDH can be considered as a derived feature from a co-occurrence matrix.

Grey-level sum histogram (GLSH) is defined similarly as follows:

$$S(\mathbf{d}) = |y_{i,j} + y_{i+d1,j+d2}| \quad (2.9)$$

Again as pointed out by Carstensen, (2.9) is really computing the co-occurrence matrix coefficients over constant grey-level of $(m+n)$. GLSH is also a derived feature from the corresponding co-occurrence matrix.

Some commonly used GLDH and GLSH features are defined as follows (where for avoiding redundancy, the generic G represents either $D(\mathbf{d})$ or $S(\mathbf{d})$):

$$\begin{array}{ll}
 \textbf{Energy:} & E = \sum_k G_k^2 \quad \textbf{Entropy:} \quad s = -\sum_k G_k \log G_k \\
 \textbf{GLDH Inertia:} & I = \sum_k k^2 G_k \quad \textbf{Local Homogeneity:} \quad H = \sum_k \frac{G_k}{1+k^2} \\
 \textbf{GLSH Sum Avg:} & A = \sum_k k G_k \quad \textbf{Cluster Shade:} \quad A = \sum_k (k-A)^3 G_k
 \end{array} \quad (2.10)$$

where k stands for the greylevel sum or difference index for calculations done using a given \mathbf{d} [Carstensen, 1992]. As this list of features shows, there are many variations of features that can be computed from GLDH and GLSH. Since GLDH and GLSH are derived features from the co-occurrence matrix, these features in (2.10) can also be

considered as features of the co-occurrence matrix. If the right classification methods using the co-occurrence matrices are used, these methods should be a superset of other methods using features in (2.10). In fact, Weszka (1976) has found that co-occurrence matrix features usually perform better than GLDH and GLSH features. The reason behind his observation could be that the later features are derived from the former co-occurrence matrices.

(iv) *Fourier spectral features*

Over the past three decades, the effectiveness of spectral features have frequently been debated. The pros and cons of these features are not discussed in this sub-section. These discussions can be found in [Weszka, *et.al.*, 1976; Chen, 1972; Wilson and Spann, 1988; Grunkin, 1993]. If the power spectrum is estimated correctly, and if the right set of spectral features are used, many researchers have found these features to be quite useful and theoretically sound. The quantities in (2.11) represent a few of the commonly used features (given the power spectrum of an image is $P(r, \phi)$ [Lim, 1990], where r and ϕ are the polar units in the frequency domain):

$$\begin{aligned}
 \text{Avg. Power:} \quad & AP = \int_{\phi=0}^{2\pi} d\phi \int_{r=0}^{r_{\max}} P(r, \phi) r dr \\
 \text{Ring from } r_1 \text{ to } r_2: \quad & R(r_1, r_2) = \int_{r_1}^{r_2} p(r, \phi) dr \\
 \text{Wedge from } \phi_1 \text{ to } \phi_2: \quad & W(\phi_1, \phi_2) = \int_{\phi_1}^{\phi_2} p(r, \phi) r d\phi
 \end{aligned} \tag{2.11}$$

From (i) to (iv), this sub-section has discussed a few of the most common and useful features. There are many other features that have been invented for various applications. Interested readers should refer to the references quoted in the beginning of this chapter for further details.

2.4.2 *Model-based Methods*

The model-based methods use features derived from modeling parameters of a given image. The commonly used models include Gaussian models, Simultaneous Autoregressive (SAR) models, and Markov random field models. These are discussed in the introductory section on model-based classification experts (Chapter 4) and are skipped here to avoid redundancy. The parameters extracted from these models are used for classification just like the features described in the previous sub-section. The general classification rules for this operation are discussed later in this thesis.

2.4.3 Structural Methods

Structural methods assume repeating primitive patterns over a given image with a specific placement rule. Therefore, these methods apply only to specific types of textural images because of the tight spatial constraints assumed by these methods. Due to this restriction, these methods are of limited utility.

The main ideas behind structural methods can be understood from a spatial-frequency point of view [Jayaramamurthy, 1980; Reed, *et.al.*, 1993]. Consider a given textural primitive $h(x, y)$ and a placement rule $d(x, y)$ composing of a set of delta functions defined as follows:

$$d(x, y) = \sum_{x_m, y_m} \delta(x - x_m, y - y_m) \quad (2.12)$$

The entire texture image is then the convolution of $h(x, y)$ and $d(x, y)$:

$$\begin{aligned} S(x, y) &= h(x, y) * d(x, y) \\ S(u, v) &= H(u, v) D(u, v) \end{aligned} \quad (2.13)$$

The lower equation in (2.13) is the frequency domain equivalent of a convolution process and u and v are the frequency coordinate variables. Clearly, we can find the placement rule by taking the inverse transform of the following deconvolution process in frequency:

$$D(u, v) = H^{-1}(u, v) S(u, v) \quad (2.14)$$

The deconvolution filter $H^1(u,v)$ is simply the inverse of the texture primitive frequency transform. So, to classify a given pixel and region is nothing more than finding the class with prototype pixels and regions with similar placement rule and the textural primitive.

Lee (1983) describes a pyramid averaging scheme. His key idea is that by averaging over the size of a texture primitive, the texture regions in the image would be turned into uniform sections. By carefully manipulating the pyramid structure, regions that are not textures would be unaffected. On texture portions, classification of different regions can be performed by locating the different constant regions.

2.5 Segmentation Methods

There are several different ways to describe segmentation methods such as supervised versus unsupervised methods, region-based versus boundary-based, maximum likelihood (ML) versus maximum *a posteriori* (MAP) or other maximization procedures, among others. This section attempts to use the supervised versus unsupervised paradigm to give an overview of many of the existing segmentation methods. Note that this paradigm can also be used to *classify* classification methods.

2.5.1 Supervised Methods

(i) Bayes' methods

This sub-section considers a parametric method, which means that assumptions are made about the underlying data generation models. Consider an image that can be modeled by a mixture of several densities $f(y|z_k)$ where $k = \{1, 2, \dots, K\}$ represents the class labels. The image has the following joint probability density function (pdf):

$$f(y, z|\Phi) = \sum_{k=1}^K g_k f(y|z_k, \Phi) \quad (2.15)$$

where z represents the class label, Φ represents all the parameters of the mixture model and g_k is the prior for class k . Consider a learning set $\chi_L = \{y_j, z_{jk}\}$ where $j = \{1, 2, \dots, L\}$ for a total of L learning pairs, modeling the underlying densities is performed using χ_L . The estimation of the model parameters Φ is often performed by either maximum likelihood (ML) or maximum *a posteriori* (MAP) methods [Besag, 1986; Derin and Elliott, 1988; Silvermann and Cooper, 1988; Zhang, *et.al.*, 1994; LaValle and Hutchinson, 1995].

In particular, the feature-based methods discussed in the last section could assume a multivariate normal density function for estimating the feature parameters, which is the approach taken in Chapter 5. In this case, the parameters of interest: $\Phi = \{\mu_k, \Sigma_k\}$ where $k = \{1, 2, \dots, K\}$ and μ_k and Σ_k are the mean and co-variance for class k . From the learning set χ_L , the parameters for each class Φ_k can be estimated directly without any iterative procedures. A maximum likelihood (ML) classifier would classify a given pixel with its local region map by maximizing the following quantity:

$$z_i^{*(ML)} = \arg \max_{z_i} f(y|z_i, \Phi) \quad (2.16)$$

As Bezdek, *et.al.*, (1993) have pointed out, this supervised ML procedure is unstable in the sense that slight changes in the input would yield outputs that are very different from the learning set χ_L . Therefore, in the case of a 3D data set, parameter estimations have to be done on every slice of the image. This problem is ameliorated by using the maximum *a posteriori* (MAP) approach instead. Using Bayes's theorem for the posterior probability of the state z_i , the following expresses the MAP classification of pixel i :

$$\begin{aligned} z_i^{*(MAP)} &= \arg \max_{z_i} f(z_i|y, \Phi) \\ &= \arg \max_{z_i} \frac{g_i f(y|z_i, \Phi)}{\sum_j g_j f(y|z_j, \Phi)} \end{aligned} \quad (2.17)$$

The denominator of (2.17) can be ignored because it is a constant that is independent of the state z_i . The prior g_i is problem dependent and can be obtained in various ways, some of which are discussed in Chapter 4 for the model-based *expert*. Both the ML and MAP

classification procedures are applied to the feature-based classification expert of Chapter 5. The density models used for the model parameters are the multivariate Gaussian densities.

(ii) *K-nearest neighbor (KNN)*

The previous section has discussed a parametric approach, this section considers a non-parametric method known as K-nearest neighbor (KNN) [Fukunaga, 1990; Bezdek, *et.al.*, 1993]. KNN is non-parametric in the sense that no prior assumptions are made on the underlying data. The only parameter in KNN is the K value, which is usually determined through some kind of cross-validation methods (Breiman, *et.al.*, 1984). This method's success relies on a large set of correctly labeled learning sets χ_L . To classify an input pattern, KNN collects the class labels belonging to that input's K closest neighbors. The class that predominates in this collection is assigned to be the class of the input.

The usual distance metric for measuring *similarities* between an input to its neighbors is the Euclidean distance $d(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j)}$, where \mathbf{x}_i is the input feature vector and \mathbf{x}_j is the feature vector of a neighbor. However, there exists many other more appropriate distance metric for many applications, such as the Mahalanobis distance discussed in Chapter 5.

It is well known in the pattern recognition field that KNN has an *upper bound* on its classification error. This upper bound is twice the optimal Bayes' error [Fukunaga, 1990]. Wolpert (1992) experimentally shows that KNN is a stable method that does not have widely oscillating outputs as its inputs or parameters are perturbed. Because of the simplicity of implementation and the robust classification capability, KNN is often used for various classification and segmentation tasks.

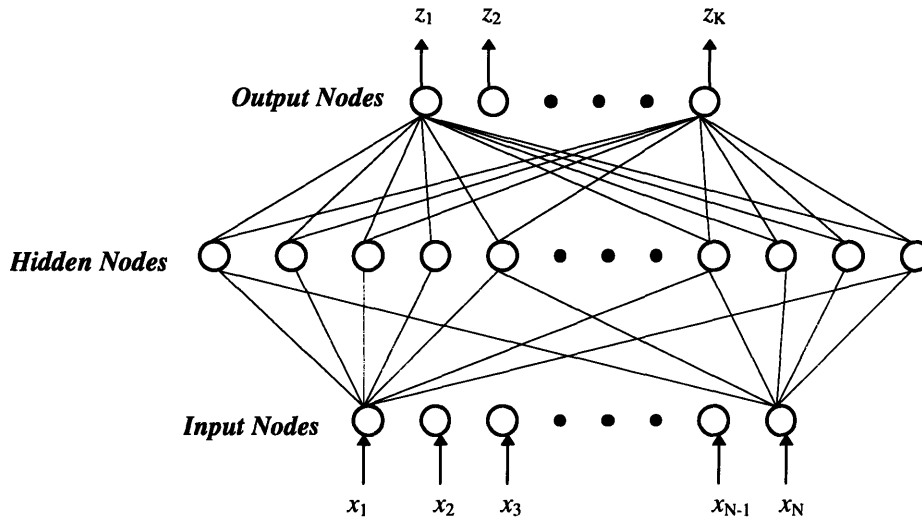


Figure 2.9 An schematic diagram for a feedforward network. This diagram shows a one hidden layer network that receives input $\mathbf{x} = [x_1 \ x_2 \dots x_N]^T$ and produces an output vector $\mathbf{z} = [z_1 \ z_2 \dots z_K]^T$.

(iii) neural networks

Neural network approaches have received much attention in the pattern recognition field recently. Many theories have been developed about neural networks. Perhaps the most famous, and often quoted, result is the Hornik, Stinchcombe, and White (1989) proof of multilayer feed-forward networks being *universal approximators*. In other words, this proof shows that such an architecture is able to approximate arbitrarily close to any *reasonable* functions. This result is quite reassuring since most functions that people encounter fall into the *reasonable* functions category. Unfortunately, even with this proof, no guarantee can be made about whether a learning problem can be learned by such a network. The main culprits lie with the imperfect training algorithms. For example, the popular delta rule [Rumelhart, *et.al.*, 1986], or backpropagation (BP), is a stochastic gradient descent method. Like other gradient based methods, BP is prone to be trapped in local minima [Hertz, *et.al.*, 1991]. Once learning gets stuck in a local minimum, another initialization of the network weight vector has to be done in order to place the weight vector away from the local minimum. In other words, even though the feed-forward network structure has the *potential* to learn all *reasonable* functions, in practice, the learning procedure required could take an infinite number of trials.

An example a feed-forward neural network is shown in Figure 2.9. The (2.9) network is a so-called a one *hidden* layer network since one of the layer of “neurons” is not connected to any of the inputs nor the outputs. This is the structure of the neural network expert used later in Chapter 6 as another *expert*.

2.5.2 Unsupervised Methods

Unsupervised methods try to discover the underlying structure of a set of unlabeled learning set χ_L using iterative methods. Like other iterative procedures, the learning process of an unsupervised method is prone to be trapped in local minima and tends to converge very slowly. Recent application of the expectation maximization (EM) algorithm to the iterative learning procedures have alleviated some of the problems that usually come along with these iterative algorithms [Dempster, *et.al.*, 1977; Kelly, *et.al.*, 1988; Liang, *et.al.*, 1994; Zhang, *et.al.*, 1994; Jordan and Jacobs, 1994]. The EM algorithm is discussed in Appendix B and is used for estimating the model-based classification experts’ parameters in Chapter 5.

(i) *unsupervised Bayes’ methods*

The unsupervised Bayes’ methods are straightforward variations of the supervised Bayes’ methods in the last sub-section. The main difference among them is that the unsupervised methods receive a learning sets χ_L that do not have labels while the χ_L for the supervised methods do. The consequence of this difference is that the prior probabilities of the different classes $p(z_k)$ are now unknown. In the case of the multi-variate normal densities discussed in the supervised Bayes’ methods section, all the parameters $\Phi = \{p(z_k), \mu_k, \Sigma_k\}$, where $k = \{1, 2, \dots, K\}$, are now coupled across different classes. In order to constrain the parameter space for the estimation problem, Lagrange multipliers can be applied to regularize the solutions. Chapter 4 shows a recent development in using the EM algorithm for estimating the multi-variate normal parameters in a maximum likelihood fashion [Bezdek, *et.al.*, 1992; Zhang, *et.al.*, 1994].

(ii) *K-means clustering*

The K-means clustering algorithm [MacQueen, 1967] has been a standard algorithm for performance comparison for many years. This algorithm is guaranteed to find the local minimum, if it exists. Numerous variations of this theme have been proposed, such as the recent *generalized* K-means algorithm of [Pappas, 1992].

The essential idea of a K-means algorithm is to iteratively find cluster centers in the input space. Mathematically, every step of the algorithm recomputes the cluster centers, μ_k , in the following fashion:

$$\mu_k^{(t+1)} = \frac{1}{N_k^t} \sum_{x_i \in S_k} x_i \quad (2.18)$$

where N_k^t is the estimated number of inputs x_i 's that belong to class k during iteration t . The set of all inputs in class k is denoted by S_k . Implementing (2.18) is fairly straightforward.

For classifying an input x_i , which could be a set of features discussed earlier in this chapter, all that is required is to find the most similar cluster to x_i . The performance of the K-means algorithm is compared against other algorithms in Chapter 5 as well as Chapter 8 of this thesis.

The above unsupervised methods assume that the number of classes, K , is known. Methodologies now exist to estimate this parameter. This research problem is called cluster validation. Several recent publications in this area are: [Zhang and Modestino, 1990; Li, *et.al.*, 1992].

After the unsupervised learning algorithms discussed above have learned their parameters and have clustered the inputs, feedback must come from the user in order to reassign the class numbers for the inputs that have been randomly chosen. This class reassignment task forms an extra post-processing step for unsupervised methods.

2.6 Summary

The classification and segmentation schemes reviewed in this chapter represent only the most commonly encountered schemes. The scope of research in these two fields is very large. Nearly all of these techniques approach image classification or image segmentation from a single model point of view.

Chapter 3

THE THEORY OF MULTI-EXPERTS APPROACH

This chapter proposes an image segmentation (or pixel classification) scheme using multi-experts approach and “soft splitting” of the data space. The scheme is based on two recently proposed ideas in statistics, the stacked generalization of Wolpert (1992), and the Hierarchical Mixture of Experts (HME) of Jordan and Jacobs. The main advantages of this classifier are two-fold. First, the scheme avoids the problems generally associated with hard decision-making classifiers such as sensitivity to noise, instability when given large and complex data sets, and poor performance given small numbers of examples. Second, the proposed classifier intelligently utilizes all available expert knowledge concerning a given data set. This later point will become clear as the chapter progresses.

We start in Section 3.1 by reviewing the background for the multi-experts approach for image segmentation (or pixel classification). The proposed segmentation scheme is presented in Section 3.2, with both the gating network approach and the stacked generalization approach. Section 3.3 discusses the parameter estimation issues involved with the Multi-Experts Classifier (MEC), which includes a learning algorithm for the scheme. Finally in Section 3.4, related works are discussed.

3.1 Introduction

This section considers the general problem of data classification. The most popular techniques for classifying a given data are maximum likelihood (ML) and maximum *a posteriori* (MAP) methods based on some estimated underlying parameters of the data. This thesis shows an alternative approach based on optimal *soft* Bayesian decisions made with all knowledge derivable from a data set.

Consider an input data set $\mathbf{y} = [y_1, y_2, \dots, y_N]^T$, for which one can propose a series of models for characterizing the data set. These models are based on the set of possible hypotheses $\mathcal{H} = \{ \mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_K \}$ about the data. Each of these models contains a set of parameters $\Phi_i \in \{ \Phi_1, \Phi_2, \dots, \Phi_K \}$ to be estimated. For hypothesis \mathcal{H}_k , assume that the model has been determined to have a joint density function $f(\mathbf{y}|\Phi_k, \mathcal{H})$. The ML classification scheme for an input value y_i , after the model parameters are estimated, can be expressed as the following maximization of its likelihood:

$$\mathcal{H}_i^{*(ML)} = \arg \max_{\Phi_k, \mathcal{H}} f(y_i|\Phi_k, \mathcal{H}) \quad (3.1)$$

Equation (3.1) says that given the estimated model parameters for all the available hypothesis, the ML hypothesis is selected as the one with the highest likelihood value. The model parameters for each hypothesis \mathcal{H}_i are obtained by maximizing the likelihood of the observed data assuming at each stage the particular hypothesis \mathcal{H}_i is correct.

Similarly, the MAP approach for classification can be expressed by a maximization process. In addition to maximizing the likelihood of the observed data for each hypothesis, MAP also takes into consideration the prior probability for each hypothesis, $f(\Phi_i|\mathcal{H})$, which can be obtained through a variety of ways that can be found in many operations research or statistics books [Drake, *et.al.*, 1978; Fukunaga, 1992]. Given the prior probability density function $f(\Phi_i|\mathcal{H})$, the MAP approach can be expressed as:

$$\mathcal{H}_i^{*(MAP)} = \arg \max_{\Phi_i, \mathcal{H}} f(y_i|\Phi_i, \mathcal{H})f(\Phi_i|\mathcal{H}) \quad (3.2)$$

The expression on the right hand side is the same expression as the *a posteriori* probability distribution. Therefore, this procedure is exactly the same as maximizing the *a posteriori* probability (MAP). Classifying an input value y_i is done by simply choosing the hypothesis with the highest *a posteriori* probability. In general, MAP estimations are superior to those obtained by ML.

In both of the methods presented above, the classification step has two parts: first, the selection of the best model, and second, the determination of the class assignment for the given data y_i based on the best model. Perhaps a more lucid way to consider this above classification procedure is: make a *hard* decision on what the *best* model is and based on that decision, choose the *best* class for an input. (A hard decision makes an input y_i belong to one model and that one only. In contrast, a soft decision allows y_i to belong to multiple classes simultaneously.) Within the information theory and neural network communities, *hard* decisions are well known to be less robust than *soft* decisions. In order to improve the classification performance of a classifier, we must avoid making hard decisions during classification.

The next section gives an overview of the multi-experts approach. Two methods are presented. In the *gating network* approach, not only does the classifier take all available models into consideration when classifying an input, the decision is made by soft partitioning of the input space. The soft decision nature of this approach in utilizing all available *expert* knowledges should become quite apparent to the readers as the discussion progresses. The stacked generalization approach, all models are also considered when a decision is made on a given input; however, the stacked generalizer (SG) that combines the different models does not deal with the input space. Instead, this SG learns from the output of different experts. Therefore, outputs are results of a higher level abstraction from the input than outputs obtained from the experts. Technically, Wolpert (1992) called this learning *first level learning*, in contrast with the *zeroth level learning* by the experts in learning to map inputs to outputs.

3.2 Multi-Experts Classification

Consider an input image S with intensity vector y , our goal is to assign every pixel in S to one of K classes in a meaningful fashion, the definition of which is problem dependent. We have several algorithms, each of which classifies certain types of image very well. However, outside their “expertise” domain of image, they perform poorly. This situation is very common in image segmentation. For example, Simultaneous Autoregressive models (SAR) have been shown to be very effective for many types of stationary textural images [Kashyap, *et.al.*, 1983; Mao, *et.al.*, 1992; Grunkin, 1993; Hu, *et.al.*, 1994]. If a noisy brickwall image with strong deterministic patterns is given to a SAR classification “expert”, the expert’s performance cannot be predicted. In a later chapter on a feature-based expert, experiments will be described in which a *good* feature set for one type of image may not be so good for another type of image. If an expert E_a has been “tuned” to do well on certain brickwall type images, E_a might not perform well on the textural domain. No doubt that any image classification task is best done when the “right” set of features and the “right” set of methods are used for particular images. Realistically, however, such an approach is not always feasible. Should a complex real-life image be considered a textural image? A fractal image? A constant plus additive noise image? Multiplicative noise image? How about non-linear chaotic image? Clearly, no single model can be perfect.

The central idea in the proposed Multi-Experts Classifier (MEC) is to intelligently combine the expertise of several experts with different image models to achieve accurate classification results. The combining process has to be handled carefully to ensure a robust and well justified scheme. The multi-experts approach presented here has some resemblance to the adaptive modular network of [Mui, Agarwal, Gupta, and Wang, 1994], except the later approach uses hard splits of input space and is not grounded in a rigorous framework. The theory of MEC can be traced to the several recently proposed ideas in statistics -- “Stacked Generalization” (SG) of David Wolpert (1992), “Bagging Predictors” (1994) and “stacked regression” (1992) of Leo Breiman, and the Hierarchical Mixtures of Experts (HME) of Jordan and Jacobs (1994). To the best knowledge of the

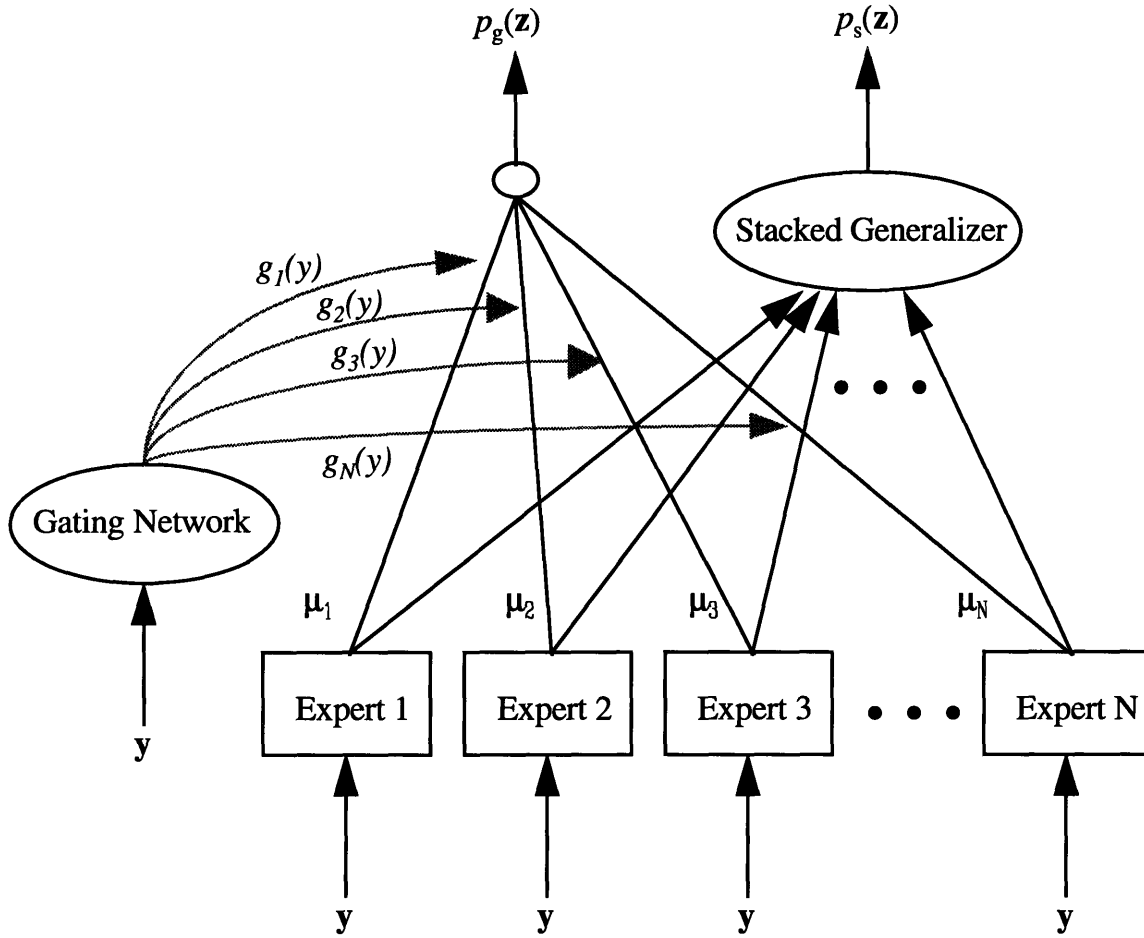


Figure 3.1 Multi-experts scheme for data (y) classification by soft decisions through priors calculated by the gating network for outputs of all experts or through 1st level learning by the stacked generalizer..

author, this thesis is the first attempt to apply the multi-experts approach to image classification and segmentation.

3.2.1 Overview of the MEC

The architecture of the MEC is shown in Figure 3.1. Every expert E_n receives an input intensity vector y . For example, y can be range measure, intensity of X-ray attenuation, amount of backscatter of an ultrasound echo, proton density, T1, or T2 acquisition time of a magnetic resonance image (MRI). Expert E_n takes in y and produces an output vector $p(y|z, \Phi_n) = \mu_n = [p_n(y|z_1, \Phi_n), p_n(y|z_2, \Phi_n), \dots, p_n(y|z_K, \Phi_n)]^T$, $n \in \{1, 2, \dots, N\}$. μ_n is in the form of a vector of probabilities of observing y given in state z_i , $i \in \{1,$

2, ... K}. In image classification, state vector \mathbf{z} represents the class labels for the input vector \mathbf{y} . These class labels can be from class 1 to class K .

In accordance with objective probability theory, the state vector μ_n produced by expert n must contain elements that satisfy the following criterion:

$$\begin{cases} 0.0 \leq p(\mathbf{y}|z_i, \Phi_n) \leq 1.0 \\ \sum_{i=1}^K p(\mathbf{y}|z_i, \Phi_n) = 1.0 \end{cases} \quad (3.3)$$

This criterion precludes the use of classification experts or methods that do not produce confidence indices for the classification results. Examples include simple region growing, intensity thresholding, and split and merge algorithms [Lim, 1990; Gonzalez and Woods, 1992; Russ, 1993]. In order to have a statistical framework for image classification and segmentation, we have to have a way to assess the classification and segmentation results. The probabilistic requirement (3.3) is a sensible way to provide this assessment.

Selection of experts will be dealt with in later chapters. For now, assume the experts E_1, E_2, \dots, E_N have been chosen. We consider how to combine their classification results here. As hinted by Figure 3.1, this thesis considers two ways for this combination process. The gating network approach and the stacked generalization approach.

(i) gating network approach

This sub-section considers two ways to view the combination of experts' outputs problem through the gating network. The first way is suggested by Jordan and Jacobs (1994). If \mathbf{y} is an observable obtained by one of the N generative processes model by the experts E_i 's, then finding the best expert for classifying \mathbf{y} is an N -way classification problem. A natural probability model then is the multinomial density. In other words, for a given input \mathbf{y} , a classifier based on the multinomial density attempts to assign \mathbf{y} to one N classes.

Consider m input vectors $\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m$, for each of these, each expert outputs a state probability vector $\boldsymbol{\mu}_n = [p_n(\mathbf{y}|z_1, \Phi_n), p_n(\mathbf{y}|z_2, \Phi_n), \dots, p_n(\mathbf{y}|z_K, \Phi_n)]^T$, where $i \in \{1, 2, \dots, K\}$, $n \in \{1, 2, \dots, N\}$. The joint probability density of the states under an N -way classification scheme can be expressed by the following multinomial density [Jordan and Jacobs, 1994]:

$$f(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_N | \mathbf{y}, \Phi) = m! \prod_{n=1}^N \frac{p(\mathbf{z}_n | \mathbf{y}, \Phi)^{z_n}}{z_n!} \quad (3.4)$$

Appendix C details the derivation for the so-called *link function* in statistics associated with this distribution [McCullagh, *et.al.*, 1983]. As shown in that appendix, the link function turns out to be the appropriate output function for the gating network. The link function is the well known *softmax* function [Bridle, 1989]:

$$p(z_n | \Phi) = \frac{e^{\mathbf{w}_n^T \mathbf{y}}}{\sum_{l=1}^N e^{\mathbf{w}_l^T \mathbf{y}}} \equiv g_n(\mathbf{y}) \quad (3.5)$$

where \mathbf{w}_n^T is the transpose of a vector of parameters associated with expert n . This function partitions the input space using *soft* splits, which allow a given point in the input space to *belong* to more than one class of experts. From another perspective, the input \mathbf{y} is projected onto different parameter vectors \mathbf{w}_n 's. The magnitude of this projection ($\xi_n = \mathbf{w}_n^T \mathbf{y}$) is passed through a nonlinear function $\tilde{g}_n(\xi_n) = \frac{e^{\xi_n}}{\sum_l e^{\xi_l}}$. The greater the

magnitude of the projection, i.e., the greater the *resemblance* of \mathbf{y} to the expert class n , the greater the g_n value for that given \mathbf{y} . The magnitude of g_n is a measure of how close an input \mathbf{y} falls near the expertise area of an expert classifier. Notice that g_n provides a probabilistic estimate given that $\sum_{n=1}^N g_n = 1.0$.

Another probabilistic model for a gating network approach is proposed in this thesis. We approach the combination problem as determining how likely \mathbf{y} is generated by each one of N processes modeled by the experts. In other words, we divide the selection

problem into N separate problems, and assess the probability of \mathbf{y} being in the “expertise” area of each expert separately. Here, we have N binary classification problems and the natural probability model is the Bernoulli process.

For most natural images, the likelihood that a given image is generated by a single process is extremely small. Most of the time, a real life image is produced by many, perhaps an infinite, number of simple processes that are easy to be modeled. Therefore, for assigning the credit for how likely a given region of the image is generated by one process, a Bernoulli trial with outcome being a success or a failure is appropriate. A collection of such trials can be attempted to model all the processes simultaneously.

Given the outcome z_i is either a success (1) or a failure (0), and the probability of success is θ , a Bernoulli process is characterized by the following density:

$$f(z_i|\mathbf{y}, \Phi) = \theta^{z_i} (1 - \theta)^{1-z_i} \quad (3.6)$$

This equation is actually a special case of equation (3.4), for when $N = 2$, which makes sense since a binary classification is just N-way classification with the “N” being 2. Therefore, through a similar derivation as Appendix C ¹, the output function for a gating network with a Bernoulli process assumption is:

$$p(z_n|\Phi) = \frac{1}{1 - e^{-\mathbf{w}_n^T \mathbf{y}}} \equiv g_n(\mathbf{y}) \quad (3.7)$$

Just like the softmax function, this is a very familiar function in the machine learning community -- the logistic function [Rumelhart, *et.al.*, 1986; Hertz, *et.al.*, 1991].

(ii) stacked generalization approach

¹ An easy way to derive this output function is to substitute 2 for every N in Appendix C.

Other than the gating network approach, combining the experts' outputs can be seen as a first level learning problem, which has already been introduced. The name of *stacked generalization* was coined by Wolpert (1992) for higher level learning than zeroth level learning, by which he means learning using inputs directly. First level learning is then learning using outputs of generalizers who generalize the inputs directly. Breiman (1992) has applied this idea in his *stacked regression* to regression problems and obtained very encouraging results. Here in this thesis, we consider classification problems.

By using first level learners such as the stacked generalizer of Figure 3.1, the aim is to reduce the generalization error of a generalizer, as opposed to the learning error. In addition to learning the input space by the different experts, the MEC uses the stacked generalizer to learn the output space of the experts. Even if all the experts miss certain input patterns, by observing their mistakes, the stacked generalizer can still output the correct output by generalizing from the *usual* mistakes made by the different experts. This is the crux of the argument for using stacked generalization for classification tasks. An interesting illustration of a stacked generalizer through the weather forecast example is given in Section 1.2.

3.2.2 A Probabilistic View of the Gating MEC

The MEC with the gating network for combination of experts can be viewed from a probabilistic perspective. The gating network serves a special role of providing the prior probability for each expert network. We can see this role through the following expression for the total output vector (where μ_n 's are the output probability vectors of the experts):

$$p(\mathbf{z}|\mathbf{y}, \Phi) = \sum_{n=1}^N g_n(\mathbf{y}) \mu_n \quad (3.8)$$

This final classification result is another vector $p(\mathbf{z}) = [p(z=1) \ p(z=2) \ \dots \ p(z=K)]^T$, where the dependence on \mathbf{y} and Φ are omitted for succinctness. Note that this final output vector is a result obtained through all available experts, combined using a probability framework. The selection of the best class can be done in several different manners. A simple and

reasonable method is to choose the class k with the highest posterior probability $p(z=k|\mathbf{y}, \Phi)$. Discussions of other methods is done later in this chapter as well as in Chapter 8, which also shows some experimental results.

We can rewrite (3.8), the output probability vector, as:

$$p(\mathbf{z}|\mathbf{y}, \Phi) = \sum_{n=1}^N g_n(\mathbf{y}) p(\mathbf{y}|\mathbf{z}_n, \Phi_n) \quad (3.9)$$

where Φ represents all MEC parameters, which include those of the gating network as well as those of the expert classifiers. The posterior probability given the input \mathbf{y} and the classifier parameters is $p(\mathbf{z}|\mathbf{y}, \Phi)$. $g_n(\mathbf{y})$ serves as the prior probability for expert n , while $p(\mathbf{y}|\mathbf{z}_n, \Phi_n)$ is the likelihood vector given by expert n with parameters Φ_n . This likelihood vector could be a result of a ML (3.1) or MAP (3.2) estimation procedure. If we consider the input \mathbf{y} being generated by one of the experts, $g_n(\mathbf{y})p(\mathbf{y}|\mathbf{z}_n, \Phi_n)$ is the posterior probability of this event. Instead of a ML or MAP procedure which chooses either the maximum of $p(\mathbf{y}|\mathbf{z}_n, \Phi_n)$ or $p(\mathbf{z}_n|\mathbf{y}, \Phi_n)$, MEC considers an *a priori* process modeled by a multinomial probability function with output $g_n(\mathbf{y})$. This approach considers all available posterior probabilities of the classification provided by all the experts. Every expert's opinion is taken into consideration in making the final classification decision.

Equation (3.9) provides a means to combine all the output vectors from all available experts through an optimal Bayes' decision rule -- optimizing the posterior probability of the output given the output results from individual experts. No *hard* decision is made here for choosing the final classification of an input pixel. In this sense, a given scene can be generated by multiple processes at the same time. The modeling of this mixed scene is done through the prior probability assignments g_n 's.

3.2.3 The Stacked Generalizer

Two different types of stacked generalizer (SG) are attempted to learn the outputs from the experts. These two types are the coincidence matrix based SG and the network based SG.

Consider the *first level* learning task ² of learning from the outputs of the experts, a generalizer attempts to learn certain structure from these data. In the case of two experts, a straightforward way is to construct a *coincidence matrix* (or tensor if the number of experts is greater than 2). This matrix has vector elements c_{ij} where i and j are the outputs from expert 1 and expert 2, respectively. c_{ij} is a vector of K components, corresponding to the K different classes for classification. Component $c_{ij,k}$ represents the number of times expert 1 and expert 2 has outputs i and j but the actual class of the input is k . These coefficients are learned during the training stage. The ones with no training data receive a class label of REJECT. The actual classification is then simply a *table lookup* operation using the coincidence matrix. An example of this approach is shown in the chapter on experiments -- Chapter 8.

Another more sophisticated way to perform the first level learning is to use a network, just like the gating network. In fact, this thesis also attempts the two probabilistic models discussed in the last section for this first level learning task. Recall these two probabilistic models are the multinomial density and the set of Bernoulli processes. The performance of these first level learners will be shown in Chapter 8.

² Refer to the last section for the definition of a first level learning task, or to Wolpert's paper (1992).

3.3 Parameter Estimation for a MEC

3.3.1 Gating Network Parameter Estimation

This subsection derives a gradient descent based learning algorithm for estimating the parameters in a MEC. Parameters are estimated in a maximum likelihood fashion. For the gating network parameter estimation, the methodologies used are comparable to the approach taken by Jordan and Jacobs (1994) for their HME architecture. Given E number of examples provided for learning, $\{(\mathbf{y}^e, \mathbf{z}^e)\}$, for $e = \{1, 2, \dots, E\}$. (The output \mathbf{z}^e is a unit vector with the component corresponding to the target class equal to 1.0.) For E examples, we maximize the log likelihood of a given MEC in estimating the parameters Φ , which is given by :

$$l(\mathbf{y}^e, \mathbf{z}^e, \Phi) = \sum_{e=1}^E \log \sum_{n=1}^N g_n(\mathbf{y}) p(\mathbf{y}^e | \mathbf{z}^e, \Phi_n) \quad (3.10)$$

Here two sets of parameters are to be estimated, $\Phi = \{ \mathbf{w}_n, \Phi_n \}$ where $n = 1, 2, \dots, N$. The parameters associated with the gating network are represented by the vector \mathbf{w}_n , and the parameters associated with each expert are represented by the vector Φ_n . The ML estimates of these parameters satisfy the equations :

$$\begin{aligned} \nabla_{\mathbf{w}_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) &= 0 \\ \nabla_{\Phi_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) &= 0 \end{aligned} \quad (3.11)$$

To avoid excessive notation, we consider the case for a single pair of examples $(\mathbf{y}^e, \mathbf{z}^e)$, we substitute the symbol $\xi_n = \mathbf{w}_n^T \mathbf{y}$. Let's first calculate the ML estimates for the gating network using ξ_n and a pair of example:

$$\frac{\partial}{\partial \xi_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) = \sum_{j=1}^N \frac{\partial l}{\partial g_j} \frac{\partial g_j}{\partial \xi_n} \quad (3.12)$$

where g_j is given by equation (3.5) and (3.7). We first consider the partial derivative of the softmax output g_j^S :

$$\begin{aligned}\frac{\partial g_j^S}{\partial \xi_n} &= \frac{\delta_{nj} e^{\xi_j}}{\sum_{l=1}^N e^{\xi_l}} - \left(\frac{e^{\xi_j}}{\sum_{l=1}^N e^{\xi_l}} \right)^2 \\ &= g_n^S (\delta_{nj} - g_j^S)\end{aligned}\quad (3.13)$$

where δ_{nj} is the Kronecker delta function. Now, let's calculate the derivative of the log likelihood function in (3.12):

$$\begin{aligned}\frac{\partial}{\partial \xi_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) &= \sum_{j=1}^N \frac{p(\mathbf{y}^e | \mathbf{z}^e, \Phi_j)}{\sum_{m=1}^N g_m^S p(\mathbf{y}^e | \mathbf{z}^e, \Phi_m)} (g_n^S (\delta_{nj} - g_j^S)) \\ &= p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) - g_n^S \frac{\sum_{j=1}^N g_j^S p(\mathbf{y}^e | \mathbf{z}^e, \Phi_j)}{\sum_{m=1}^N g_m^S p(\mathbf{y}^e | \mathbf{z}^e, \Phi_m)} \\ &= p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) - g_n^S\end{aligned}\quad (3.14)$$

where we have used the definition of posterior probability of :

$$p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) = \frac{g_n^S(\mathbf{y}^e) p(\mathbf{y}^e | \mathbf{z}^e, \Phi_n)}{\sum_{m=1}^N g_m^S(\mathbf{y}^e) p(\mathbf{y}^e | \mathbf{z}^e, \Phi_m)} \quad (3.15)$$

So now, we can write down the gradient of the log likelihood (3.10) with respect to \mathbf{w}_n , and we consider all E example pairs :

$$\begin{aligned}\nabla_{\mathbf{w}_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) &= \sum_{e=1}^E \left(\frac{\partial}{\partial \xi_n} l \right) \nabla_{\mathbf{w}_n} \xi_n \\ &= \sum_{e=1}^E (p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) - g_n^S(\mathbf{y}^e)) \mathbf{y}^e\end{aligned}\quad (3.16)$$

For the logistic output case, the partial derivative logistic function in equation (3.12) is:

$$\frac{\partial g_j^L}{\partial \xi_n} = g_n^L (1 - g_j^L) \quad (3.17)$$

By substituting this gradient quantity to (3.12) and manipulating the outcome in a similar fashion as (3.14), we can derive the corresponding log likelihood (3.10) for the logistic function output case, which is:

$$\nabla_{\mathbf{w}_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) = \sum_{e=1}^E \left(p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) - g_n^L(\mathbf{y}^e) \right) (1 - g_n^L(\mathbf{y}^e)) g_n^L(\mathbf{y}^e) \mathbf{y}^e \quad (3.18)$$

Equation (3.16) and (3.18) provide us “greedy” means of learning the parameters to maximize the likelihood in (3.10). It has been well known in the machine learning community that such methodologies are inferior to incremental stochastic techniques in many real life learning tasks. To convert (3.16) and (3.18) to stochastic estimation techniques simply involves two steps. First, we consider each example pair individually -- drop the summation sign. Second, we introduce a learning rate constant $0 < \eta < 1$ to the parameter update equations. η changes the values of \mathbf{w}_n by a fraction of what (3.16) and (3.18) dictate. So, our learning algorithms for the gating network that go up the gradient of the likelihood surface are as followed for the softmax output and the logistic output cases:

$$\begin{aligned} \Delta \mathbf{w}_n^S &= \eta \left(p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) - g_n^S(\mathbf{y}^e) \right) \mathbf{y}^e \\ \Delta \mathbf{w}_n^L &= \eta \left(p(\mathbf{z}^e | \mathbf{y}^e, \Phi_n) - g_n^L(\mathbf{y}^e) \right) (1 - g_n^L(\mathbf{y}^e)) g_n^L(\mathbf{y}^e) \mathbf{y}^e \end{aligned} \quad (3.19)$$

The usual value taken for η is around 0.1. A *hack* can also be used to accelerate the stochastic learning process in (3.19) through the use of momentum [Rumelhart, *et.al.*, 1986; Hertz, *et.al.*, 1990] This momentum term is essentially a fraction of the previous iteration’s weight change. This term helps learning if the gradient surface has long *ravines* or large constant surfaces, which are fairly common in real world learning problems. The implementation of the gating networks used in this thesis adopts this momentum idea.

3.3.2 Stacked Generalizer Parameter Estimation

Parameter estimation for the *coincidence matrix* (or tensor if the number of experts is greater than 2) is fairly trivial. This matrix has vector elements \mathbf{c}_{ij} where i and j are the outputs from expert 1 and expert 2, respectively. \mathbf{c}_{ij} is a vector of K components, corresponding to the K different classes for classification. Component $c_{ij,k}$ represents the number of times expert 1 and expert 2 has outputs i and j but the actual class of the input is k . During the learning stage, the coincidence matrix vector coefficients corresponding to the learning outputs from the experts and the actual class are incremented. By the end of learning, there are most likelihood coefficients which have not received any learning increment at all. The vector elements, \mathbf{c}_{ij} , which do not receive any learning increment automatically are associated with the REJECT class. As mentioned previously, the actual classification is performed in a table lookup fashion. The inputs who receive a REJECT status are classified by K-nearest neighbors approach [Fukunaga, 1990].

Parameter estimation for the networks used in the network approach in first level learning is done in a similar stochastic fashion as the gating networks. The key learning equation is again (3.15). The details will be discussed in Chapter 8 along with performance evaluation results.

3.3.3 Experts Parameter Estimation

For estimating the parameters of the expert classifiers in a MEC, there are two ways to approach the estimation problem. One way is to estimate the parameters by maximizing the overall MEC log likelihood. In other words, all the parameters of all the experts Φ are estimated at the same time. The second approach is to approximate the maximization of the overall log likelihood process by maximizing each Φ_n for each expert individual and combining them through the ML gating network derived above.

There are two main reasons for not using the first approach, which is again: $\nabla_{\Phi_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) = 0$. First of all, many of the experts that are used for this thesis are nonlinear systems. Optimizing some of these experts already takes approximation steps

that the error incurred in those steps is already bigger, maybe much bigger, than a non-ML procedure. Secondly, if the first procedure is used, modularity of the MEC is severely compromised. A modular classifier allows experts to be added on and taken away easily, with only adjustments of the gating parameters. The addition or deletion of an expert has no effect on the parameters of other experts. Such a modular classifier can easily be used for different applications without going through parameter estimations for every component of the classifier. The first approach for estimating all the parameters at the same time by maximizing the overall likelihood requires re-estimation every time an expert is added or deleted from the overall system in order to yield *maximum likely* results. These two disadvantages point to the need for a better approach than the first for estimating the experts' parameters.

A second approach for estimating the parameters is to estimate each expert's parameters individually and to combine the results by adjusting the gating network parameters. Consider for expert n , our goal is to maximize its posterior probability given the input \mathbf{y} :

$$\Phi_n^* = \arg \max_{\Phi_n} \log f(\mathbf{y}|\Phi_n)f(\Phi_n) \quad (3.20)$$

This expression is essentially (3.2) for maximum *a posteriori* probability estimation given hypothesis n is correct. For every $n \in \{1, 2, \dots, N\}$, the parameters Φ_n are optimized in the MAP sense. The next several chapters will shown how the MAP optimizations for different expert classifiers are accomplished. The main learning steps for the overall MEC is then to find parameters for the gating network that satisfy the following relation: $\nabla_{\mathbf{w}_n} l(\mathbf{y}^e, \mathbf{z}^e, \Phi) = 0$. The only assumption here is that the MAP estimates for the individual experts' parameters are given. The learning procedure for achieving this goal was already shown in the last sub-section for the gating network parameter estimation³.

³ Jordan and Jacobs (1994) suggested an expectation maximization (EM) based learning algorithm for their HME architecture and have obtained very encouraging results in terms of speed of convergence to the (local) ML. EM can potentially be useful for estimating the parameters of the MEC. The author is currently considering applying it to the MEC.

3.4 Related Works

The Multi-Experts Classifier (MEC) presented in this section can be related to several existing statistical schemes. All these schemes have in common their divide-and-conquer nature in processing the input data space. Wolpert (1992) first explicitly proposes a general framework for combining several *generalizers* (which we refer to as *experts*) to reduce the generalization error of an estimator. In statistical terms, the bias (or roughly the inconsistency of estimation) of the estimate is reduced. Earlier methods along similar ideas are several well-known methods such as the Classification and Regression Tree (CART) algorithm, developed by Breiman, Friedman, Olshen, and Stone (1984), the ID3 algorithm of Quinlan (1986) and the Multivariate Adaptive Regression Splines (MARS) algorithm of Friedman (1991). These algorithms use “hard splits” of input space with additional constraints on the orientations of these splits with respect to the coordinate systems. Hard splits have negative consequence on the variance of the estimations. Soft splittings, on the other hand, reduces variances. Jordan and Jacobs (1994) therefore adopted a soft-splitting scheme with a softmax function-based gating network, which they call the Hierarchical Mixtures of Experts (HME). They have demonstrated clearly the superiority of their approach to the former three methods through experiments cited in their paper.

The MEC follows the HME’s example in using soft-splittings of the input space, with arbitrary orientations of these splitting surfaces ⁴. However, unlike the HME, the Multi-Experts Classifier (MEC) is designed strictly for classification purposes, not also for regression. HME is a generalized linear model (GLIM) for supervised learning. On the other hand, the MEC presented in this section is in general composed of non-linear components for both supervised and unsupervised learning. From experimental results shown later on in the thesis, classifying highly complex and noisy signals require much more discriminating power than that offered by most GLIM models, which include many connectionist schemes such as multi-layered perceptrons (MLP). MEC is not hierarchical

⁴ For the “single” layer softmax function or logistic function discussed earlier, these surfaces turn out to be hyperplanes.

as HME since only one “layer” of experts is employed. The author of this thesis sees no obvious ways to combine the experts sensibly above this first layer except to combine all of them to form the output. In the following chapters, the uses of MEC are illustrated in the context of image classification and segmentation.

Chapter 4

MODEL-BASED CLASSIFICATION METHODS

Modeling the configuration of an image can be a very involved process. The non-linearity of the acquisition device, the quantization effect due to digitization, the bandwidth limit of the transmission medium from a transducer to a storage device, among other causes, all contribute to the error in forming an “exact” image. The detailed physics of image formation is virtually impossible to model. Given this fact, any model-based image processing techniques can at best be based on approximations. How good these approximations are depend on the validity of the assumptions and the appropriateness of the methodologies resting on these assumptions.

This chapter describes a model-based image classification and segmentation scheme, which is used for performance comparison with the MEC results in Chapter 8. This scheme can potentially be incorporated into the MEC as an expert. The chapter starts by explaining the assumptions made on the image formation process and the resultant image models used. The models are all statistical models that are based on extensively studied techniques in statistical pattern recognition. These models include Markov random field models (MRF), Gaussian distribution models, and simultaneous auto-regressive (SAR) models. We present improvements to some of these approaches, as well as extensions for potential incorporation into the Multi-Expert Classifier (MEC) presented in the previous chapter. Because the model-based classifiers in this chapter are all parametric

classifiers, parameter estimation is a very important step in using these models. This step is viewed from an incomplete data perspective. An expectation maximization (EM) procedure for estimating these models is presented. Finally, some experimental results are reported along with discussions.

4.1 Notations and Assumptions

Given an image S with N grid points, the intensity values can be represented by a vector $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$ which is a realization of the intensity random vector $\mathbf{Y} = \{Y_1, Y_2, \dots, Y_N\}$. The number of intensity levels is denoted by C . In other words, every y_i can take C different values. The space composing of all possible images is $\Omega = \{(y_1)^C, (y_2)^C, \dots, (y_n)^C\}$. Clearly, for any moderate size image, such as one with 128x128 lattice points S , this space of all possible images is enormous.

Assume every pixel i in the image S belongs to one of K classes^{*}, then the random vector $\mathbf{Z} = \{Z_1, Z_2, \dots, Z_n\}$ designates the underlying classes random process that an intensity random vector \mathbf{Y} belongs to. The goal of image classification and image segmentation is to find the true class vector $\mathbf{z}^* = \{z_1^*, z_2^*, \dots, z_n^*\}$, where $z_i = \{1, 2, \dots, K\}$ and $i = \{1, 2, \dots, N\}$. To formalize our goal, we would like to estimate the true \mathbf{z}^* by maximizing the joint probability density of \mathbf{y} and \mathbf{z} :

^{*} this can be considered as a necessary assumption within the objective probability theory where a given random variable x_i satisfies the following probability relations :

$$0 \leq P(x_i) \leq 1.0$$

$$P(\text{certainty}) = 1.0$$

$$P(x_1 \text{ or } x_2) = P(x_1) + P(x_2)$$

Recently, there have been criticisms of the objective probability theory approach to model probabilistic events. Many recent publications dealing with subjective probability theory and fuzzy logics have received much attention. Interested readers are referred to [Kosko, 1992] and [Bezdek, 1993]. Nevertheless, objective probabilistic theory is still an excellent and rigorous theory for dealing with uncertainties and beliefs [Pao, 1989], [Fukunaga, 1992], [Duda and Hart, 1972].

$$\mathbf{z}^* = \arg \max_{\mathbf{z}} P(\mathbf{y}, \mathbf{z}) \quad (4.1)$$

With this goal in mind, we make two basic assumptions on the images. The rest of this chapter will approach image classification segmentation based on these two fairly general assumptions.

Assumption 1 : the intensity random variables (Y 's) depend on local class assignments (\mathbf{z} 's) through the conditional probability density function (pdf) $p(y_i | \mathbf{z}_j, j \in \nu_i)$ where ν_i the vicinity of pixel i , which includes i . (This assumption is essentially stating a common fact that the point spread function for most images has a small spread.)

Assumption 2 : the true class assignment \mathbf{z}^* is locally dependent and can be modeled by a conditional probability density function at pixel i as : $f(z_i) = f(z_i | \mathbf{z}_{S \setminus i})$, where $S \setminus i$ denotes all pixels in an image S *except* pixel i . As Julian Besag remarked in [Besag, 1987], $S \setminus i$ is the *only* natural conditioning set for any spatial distribution where the pixel ordering provides no causal constraints.

These assumptions are frequently made by researchers in the fields of image classification and segmentation [Besag, 1987; Derin, *et.al.*, 1987, Zhang, *et.al.*, 1994]. These assumptions have been found to be fairly general and robust for many types of images. Based on them, we can rewrite our goal (4.1) as:

$$\begin{aligned} \mathbf{z}^* &= \arg \max_{\mathbf{z}} f(\mathbf{y}, \mathbf{z}) \\ &= \arg \max_{\mathbf{z}} f(\mathbf{y} | \mathbf{z}) f(\mathbf{z}) \\ &= \arg \max_{\mathbf{z}} \prod_{i=1}^n f(y_i | \mathbf{z}_j, j \in \nu_i) f(\mathbf{z}) \end{aligned} \quad (4.2)$$

Essentially, (4.2) has separated our problem into two parts, the state process represented by $f(\mathbf{z})$ and the intensity process represented by the product term. The rest of this chapter

will focus on applying different probability distributions for $f(\mathbf{y}|\mathbf{z})$ and $f(\mathbf{z})$ to find the optimal classification \mathbf{z}^* for different images.

4.2 Model Specification

An image realization $f(\mathbf{y}, \mathbf{z}) = f(\mathbf{y}|\mathbf{z})f(\mathbf{z})$ is done in two separate parts. First, the class (or state) distribution $f(\mathbf{z})$ can be a realization of various processes. The two main ones have been the simultaneous auto-regressive (SAR) models and the conditional Markov (CM), or more commonly, the Markov random field (MRF) models [Besag, 1974; Kashyap and Chellappa, 1983; Geman and Geman, 1984; Mao, *et.al.*, 1992; Grunkin, 1993; Hu, *et.al.*, 1994; Lundervold, *et.al.*, 1995]. SAR is a subset of MRF. In other words, for every realization of SAR, there exists a unique MRF with equivalent spectral density function [Besag, 1974; Mao, *et.al.*, 1992]. SAR is generally more parsimonious in its representation of an image than MRF is. If SAR is extended to include the moving average [MA] part, then the resulting SARMA is no longer a subset of MRF. However, SARMA is computationally very expensive.

Secondly, the intensity process, $f(\mathbf{y}|\mathbf{z})$, can be realized through various probabilistic processes. For a variety of other images, $f(\mathbf{y}|\mathbf{z})$ can be modeled as a Gaussian distribution very well. But in some specific applications, other models are more appropriate. For certain types of textures, Simultaneous autoregressive (SAR) models are very effective [Grunkin, 1993]. In nuclear medicine, such as images derived from single-photon emission computed tomography (SPECT), the Poisson distribution represents the data intensity from first principles. For ultrasonic backscatter images common in medical echocardiography, Rayleigh distribution seems to be an excellent model [Melton, *et.al.*, 1992].

4.2.1 State Process

For the state process, we are concerned with specifying the distribution $f(\mathbf{z})$. Since $f(\mathbf{z})$ depends on all pixels, expressing this joint distribution $f(\mathbf{z})$ in closed form is in general

very difficult. Besag has shown in his landmark paper [Besag, 1972] that the most general form for $f(\mathbf{z})$ can be written in the expansion :

$$\ln f(\mathbf{z}) - \ln f(\mathbf{0}) = \sum_{1 \leq i \leq n} z_i G_i(z_i) + \sum_{1 \leq i < j \leq n} z_i z_j G_{i,j}(z_i, z_j) + \dots \\ z_1 z_2 \dots z_n G_{1,2,\dots,n}(z_1, z_2, \dots, z_n) \quad (4.3)$$

where the function $G_{i,j,\dots,c}(z_i, z_j, \dots, z_c)$ is any arbitrary function that is non-zero if and only if i, j, \dots, c forms a clique[†]. This representation holds as long as $f(\mathbf{0}) > 0$, which is the *positivity condition* that is used to prove the Hammersley and Clifford theorem mentioned in the previous background chapter.

We can use a Markov random field to realize $f(\mathbf{z})$. As defined in the background section, a MRF satisfy the following condition : $f(z_i) = f(z_i | z_j, j \in \eta_i)$ where η_i is the neighborhood of pixel i , which obviously does *not* include i . The size of the neighborhood determines the *order* of the MRF. We now relate the conditional pdf $f(z_i | z_j, j \in \eta_i)$ to the joint distribution $f(\mathbf{z})$ through the same pseudo-likelihood as Besag [Besag, 1986] and Zhang [Zhang, *et.al.*, 1994] to approximate the joint distribution $f(\mathbf{z})$:*

$$f(\mathbf{z}) \approx \prod_i f(z_i | z_j, j \in \eta_i) \quad (4.4)$$

where again η_i is again the local neighborhood of pixel i . Now, $f(z_i | z_j, j \in \eta_i)$ has the form as that of MRF. We can now write the state marginal conditional pdf for every pixel in the image as :

[†] The concept of a clique was discussed in Chapter 2.

^{*} This pseudo-likelihood makes use two concepts, the coding scheme and Iterated Conditional Model (ICM) of Julian Besag [Besag, 1972, 1986]. The approximation is justified by the rapid convergence of the likelihood estimation (in less than half a dozen cycles.)

$$\begin{aligned}
f(z_i) &= \sum_{z_j, j \neq i} f(\mathbf{z}) \\
&\approx \sum_{z_j, j \neq i} \prod_j f(z_j | z_i, l \in \eta_j) \\
&= f(z_i | z_i, l \in \eta_i)
\end{aligned} \tag{4.5}$$

Now, we can find the state probability at any pixel i by using the above conditional pdf given the states of the pixel's neighborhood pixels.

We are now concerned with specifying the conditional pdf $f(z_i | z_j, j \in \eta_i)$, which a *locally* dependent field. Various spatial interaction models within this setting have been studied extensively [Besag, 1972, 1986; Kashyap, *et.al.*, 1983]. We will concentrate on a powerful pairwise interaction model defined as :

$$f(z_i | z_j, j \in \eta_i) = \frac{1}{Z_n} e^{\left\{ \sum_{1 \leq i \leq n} G_i(z_i) + \sum_{1 \leq i < j \leq n} G_{ij}(z_i, z_j) \right\}} \tag{4.6}$$

where the normalizing factor Z_n is called the *partition function* in the statistical mechanics community. $G(\mathbf{z})$ is still the same as defined in a previous paragraph. When we define the energy U as :

$$U(z_i | z_j, j \in \eta_i) = - \left(\sum_{1 \leq i \leq n} G_i(z_i) + \sum_{1 \leq i < j \leq n} G_{ij}(z_i, z_j) \right) T \tag{4.7}$$

where T stands for temperature, we have made a link to statistical mechanics through the Gibbs distribution [Geman and Geman, 1984]:

$$p(z_i | z_j, l \in \eta_i) = \frac{1}{Z_n} e^{-U(z_i | z_j, l \in \eta_i) / T} \tag{4.8}$$

We now see that our pairwise interaction MRF specifies a measure of the energy of a spatial system. Maximizing the pdf $f(z_i | z_l, l \in \eta_i)$ is equivalent to finding the state of pixel i that has the minimal (Gibbs free) energy.*

Now, let's consider in more concrete terms the MRF model used in this thesis for specifying the state random vector \mathbf{z} . If the current pixel is i , with state z_i , $G_i(z_i)$ is taken to be the product of a constant specific for each class k , with the number of the local pixels (n_k) that also have class label k , i.e. $G_i(z_i) = \alpha_k n_k$. $G_{ij}(z_i, z_j)$ is taken to be the product of another class specific constant (β_{kl}) with the number of neighboring pixel j ($j \in \eta_i$) that have class label l , i.e. $G_{ij}(z_i, z_j) = \beta_{kl} n_l$. We now have the following conditional pdf for realizing the MRF:

$$f(z_i = k | z_j, j \in \eta_i) = \frac{1}{Z_n} e^{\{\alpha_k n_k - \sum_{l \neq k} \beta_{kl} n_l\}} \quad (4.9)$$

where $k = \{1, 2, \dots, K\}$ is the value of the state random variable z_i at pixel i . α_k 's and β_{kl} 's are the parameters to be estimated for a given MRF. The α_k 's controls the prior probability of class k while β_{kl} controls the local smoothing around pixel i . This local smoothing effect is in the spirit of conditioning the ill-posed nature of an early vision problem such as image segmentation [Poggio, *et.al.*, 1986; Marroquin, *et.al.*, 1987].

4.2.2 Intensity Process

The intensity process is dependent on the state process through the conditional probability distribution $f(\mathbf{y}|\mathbf{z})$. In our assumption 1, we have specified that the random variables Y 's are dependent on state Z 's in their vicinities. We consider two types of vicinity for the intensity process. First, the intensity at pixel i , Y_i is dependent only on that

*More specifically, the most likely statistical mechanical state is one with the minimal Gibbs free energy, which is a measure of both the internal energy E and the entropy S of a system. Let U represents the Gibbs free energy, then $U = (E - TS)$ where T is the temperature of the system. Note that the state with minimal Gibbs free energy has maximal entropy.

particular pixel's class assignment (v_i only contains pixel i which means that the intensity random variable Y_i is independent of neighboring state assignments). This simple intensity process can be represented by the pdf: $f(\mathbf{y}|\mathbf{z}) = \prod_{i=1}^n f(y_i|z_i)$. In the second case, Y_i depends on class assignments in a larger vicinity: $p(y_i|z_j, j \in v_i)$. These two cases will be discussed separately below.

(1) single pixel vicinity case:

For the former case where Y_i depends only on a single Z_i , we model the pdf using a Gaussian density. Image S has lattice points indexed by i . The pdf equation is presented for a single pixel i with state $z_i = k$ as follows:

$$f(y_i|z_i = k) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(y_i - \mu_k)^2}{2\sigma_k^2}} \quad (4.10)$$

The Gaussian density has been used by many researchers and has been found to be a good model for many real world images, at least as a first order approximation [Besag, 1986; Zhang, *et.al.*, 1994]. If an image has K classes, we would use K different Gaussian densities to model all K classes. Essentially, we are treating this intensity modeling problem as a Gaussian mixture (or mixture of Gaussian) problem [Fukunaga, 1990].

(2) multiple pixels vicinity case

Now, let's consider the later case in which the intensity random variables Y 's depend on their local vicinities (v)' state values through the relation: $p(y_i|\mathbf{z}) = p(y_i|z_j, j \in v_i)$. A natural choice to use to model this intensity process is again MRF. We again use the pairwise interaction model presented in the state process section, with appropriate changes to account for the intensity process :

$$f(y_i|z_j, j \in v_i) = \frac{1}{Z_n} e^{-U(y_i|z_j, j \in v_i)/T} \quad (4.11)$$

Again, Z_n is a normalizing factor and T is the *temperature* of the system. The local energy measure U is defined as follows:

$$U(y_i|z_j, j \in v_i) = u_1(y_i|z_i) + \sum_{(i,j) \in C_2, i \neq j} u_2(y_i, y_j|z_i, z_j) \quad (4.12)$$

C_2 represents second order clique (see clique definition in the background chapter). u_1 and u_2 are pairwise energy functions just like those of the state process in (4.9). Note particularly the dependence of this intensity process on the vicinity of pixel i includes pixel i , in contrast to the MRF state which depends only on the neighborhood of pixel i , which does **not** include pixel i . Now, let's consider the conditional joint pdf for this intensity process $f(\mathbf{y}|\mathbf{z})$. Although the intensity y_i of a pixel i depends on the class assignment in its vicinity, y_i is independent of y_j for $j \in v_i$ (see *Assumption 1*). The dependence of y_i on its vicinity's state value can be expressed as:

$$\begin{aligned} f(\mathbf{y}|\mathbf{z}) &= \prod_i f(y_i|z_j, j \in v_i) \\ &= \frac{1}{Z_n} e^{-U(\mathbf{y}|\mathbf{z})} \end{aligned} \quad (4.13)$$

where the Gibbs energy is defined by the following U function:

$$U(\mathbf{y}|\mathbf{z}) = \sum_i \left[u_1(y_i|z_i) + \sum_{(i,j) \in C_2, i \neq j} u_2(y_i, y_j|z_i, z_j) \right]$$

With the single pixel and multiple pixels intensity models, assume the parameters are estimated, we now have all the information we need to classify an image through the following maximization. (Parameter estimation is considered in the next section.) For the single pixel vicinity intensity process, the goal stated at the beginning of this chapter can now be written as :

$$\begin{aligned}
\mathbf{z}^* &= \arg \max_{\mathbf{z}} \prod_{i=1}^n f(y_i | z_i) f(\mathbf{z}) \\
&\approx \arg \max_{\mathbf{z}} \prod_{i=1}^n f(y_i | z_i) f(z_i | z_j, j \in \eta_i)
\end{aligned} \tag{4.14}$$

And for the multiple pixels vicinity case, the goal can be written as :

$$\begin{aligned}
\mathbf{z}^* &= \arg \max_{\mathbf{z}} \prod_{i=1}^n f(y_i | z_i) f(\mathbf{z}) \\
&\approx \arg \max_{\mathbf{z}} \prod_{i=1}^n f(y_i | z_j, j \in v_i) f(z_i | z_j, j \in \eta_i)
\end{aligned} \tag{4.15}$$

Note that these are *maximum a posteriori* (MAP) classification rules, maximizing $f(\mathbf{y}, \mathbf{z})$ through its likelihood function $f(\mathbf{y} | \mathbf{z})$ as well as through its prior $f(\mathbf{z})$.

4.3 Parameter Estimation

Estimation of the state and intensity model parameters is presented through an incomplete data problem viewpoint. Let Φ stands for all the parameters of interest. Estimations of Φ are done using maximum likelihood (ML) by expectation maximization (EM). A brief introduction to the incomplete data problem and the EM algorithm is given in Appendix B. Details of the algorithms and various applications can be found in the references [Dempster, *et.al.*, 1972; Kelly, *et. al.*, 1988; Liang, *et.al.*, 1994; Jordan and Jacobs, 1994; Zhang, *et.al.*, 1994].

EM is an iterative optimization procedure. As explained in Appendix B, the EM algorithm consists of two iterative steps: first, calculate the expectation of the log likelihood (represented by Q) of the complete data $\mathbf{x} = \{ \mathbf{y}, \mathbf{z} \}$ based on current estimates of the parameters $\hat{\Phi}^{(p)}$ at iteration p . Second, maximize the next iteration's parameter estimates with respect to the true parameters Φ , $\hat{\Phi}^{(p+1)} = \arg \max_{\Phi} Q(\Phi | \hat{\Phi}^{(p)})$. The log likelihood of the complete data \mathbf{x} given the parameters of the model is:

$$\begin{aligned} f(\mathbf{x}|\Phi) &= f(\mathbf{y}, \mathbf{z}|\Phi) \\ \log f(\mathbf{x}|\Phi) &= \log f(\mathbf{y}|\mathbf{z}, \Phi) + \log f(\mathbf{z}|\Phi) \end{aligned} \quad (4.16)$$

The function Q is the conditional expectation of this log likelihood given \mathbf{y} and $\hat{\Phi}$. Let's first consider the part of Q that comes from the second term $\log f(\mathbf{z}|\Phi)$. Recall our earlier expression (4.8) for the conditional pdf of the state process:

$$f(z_i = k | z_j, j \in \eta_i) = \frac{1}{Z_n} e^{\{\alpha_k n_k - \sum_{l \neq k} \beta_{kl} n_l\}} \quad (4.17)$$

Finding the expression for the expectation of the second term on the right hand side can be obtained through the introduction of the vector $\mathbf{V}(\Phi)$ (\mathbf{V} was first proposed by Zhang [Zhang, *et.al.*, 1994] for finding Q ; the rest of this section follows his argument closely). The vector \mathbf{V} is defined as:

$$\mathbf{V}(\Phi) = [\log f(k=1|\Phi) \quad \log f(k=2|\Phi) \quad \dots \quad \log f(k=K|\Phi)]^T \quad (4.18)$$

where T stands for vector transposition. Note in particular that $\mathbf{V}(\Phi)$ is a function only of the true parameters Φ and does *not* depend on any estimated parameters and therefore can be taken out of the expectation brackets of $E[\log f(\mathbf{z}|\Phi)]$. With the \mathbf{V} vector, we can now write the conditional expectation of the state log likelihood as follows:

$$\begin{aligned} E[\log f(\mathbf{z}|\Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] &= E[\log \sum_i f(z_i | z_l, l \in \eta_i, \Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] \\ &= E[\sum_i z_i^T \mathbf{V}(\Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] \\ &= \sum_i E[z_i^T | \mathbf{y}, \hat{\Phi}^{(p)}] \mathbf{V}(\Phi) \end{aligned} \quad (4.19)$$

If a single pixel i has state $z_i = k^\circ$. When we consider all the possible k values that z_i can take, we can think of z_i as a vector, $z_i \in \{e_1 \ e_2 \ \dots \ e_K\}$ where e_k stands for the k^{th} unit vector $[0 \ 0 \ \dots \ 1 \ \dots \ 0]^T$. When $z_i = k^\circ$ is *dot-producted* to $\mathbf{V}(\Phi)$, the result is the k^{th} component of $\mathbf{V}(\Phi)$. This trick reduces the problem to finding the expectation of z_i over

all the pixels. Finding the expectation of the state at pixel i , $z_i=k$ where $k \in \{ 1, 2, \dots, K \}$ is essentially finding the likelihood of pixel i being in state k . Therefore, along with Bayes' theorem, we can express:

$$E[z_i = k | \mathbf{y}] = f(z_i = k | \mathbf{y}) = \frac{f(\mathbf{y} | z_i = k) f(z_i = k)}{\sum_j f(\mathbf{y} | z_i = j) f(z_i = j)} \equiv \hat{z}_{ik} \quad (4.20)$$

The significance of this quantity is that $E[z_i=k|\mathbf{y}]$ is equivalent to the *posterior probability* of pixel i in state k (we will label this quantity \hat{z}_{ik} , where the caret means “estimated”¹). Later on, we will use this posterior probability for soft-splitting a feature space when classifying an input pattern.

In the same way, let's consider the conditional expectation of the intensity log likelihood. Similar to the definition of $\mathbf{V}(\Phi)$, we will another vector quantity $\mathbf{U}_s(\mathbf{y}_i|\Phi)$. Let's consider the *single-pixel vicinity* process first and define $\mathbf{U}_s(\mathbf{y}_i|\Phi)$ as :

$$\mathbf{U}_s(\mathbf{y}_i|\Phi) = [\log f(y_i | z_i = 1, \Phi) \quad \log f(y_i | z_i = 2, \Phi) \quad \dots \quad \log f(y_i | z_i = K, \Phi)]^T \quad (4.21)$$

Just like $\mathbf{V}(\Phi)$, $\mathbf{U}_s(\mathbf{y}_i|\Phi)$ is a function of the true parameters Φ and does not depend on any estimated parameters and therefore can be taken out of the expectation brackets :

$$\begin{aligned} E[\log f(\mathbf{y} | \mathbf{z}, \Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] &= E[\log \sum_i f(z_i | z_i, l \in v_i, \Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] \\ &= E[\sum_i z_i^T \mathbf{U}_s(\mathbf{y}_i|\Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] \\ &= \sum_i E[z_i^T | \mathbf{y}, \hat{\Phi}^{(p)}] \mathbf{U}_s(\mathbf{y}_i|\Phi) \end{aligned} \quad (4.22)$$

¹ For some books in statistics, the caret usually refers to an “estimator”, or a rule for estimation [Dougherty and Giardina, 1987]. The estimated quantity is usually represented by a star such as: z^* . We avoid making this distinction so as not to introduce too many notations, which are already plentiful in this chapter.

In a single pixel vicinity, y_i at pixel i depends only on state x_i at pixel i . So, we can rewrite equation (4.21) :

$$E[z_i = k | \mathbf{y}] = \frac{f(y_i | z_i = k) f(z_i = k)}{\sum_j f(y_i | z_i = j) f(z_i = j)} \equiv \hat{z}_{ik} \quad (4.23)$$

This conditional expectation (4.26) yields an estimate of the state value at pixel i .

With the two defined vector quantities \mathbf{U}_s and \mathbf{V} , we can now write down the function Q for the EM algorithm. ML parameter estimations are found by taking the gradient of Q with respect to the true parameters and setting the result equal to 0:

$$\sum_i E[z_i^T | \mathbf{y}, \hat{\Phi}^{(p)}] (\nabla_{\Phi_y} \mathbf{U}_s(y_i | \Phi)) + \sum_i E[z_i^T | \mathbf{y}, \hat{\Phi}^{(p)}] (\nabla_{\Phi_z} \mathbf{V}(\Phi)) = 0 \quad (4.24)$$

For the multiple pixels vicinity intensity process, recall the MRF used to model the process has the Gibbs distribution of the form : $f(\mathbf{y} | \mathbf{z}) = \frac{1}{Z_n} e^{-U(\mathbf{y} | \mathbf{z}) / T}$ where T is the temperature term and $U(\mathbf{y} | \mathbf{z})$ is the Gibbs energy of the system and is defined as :

$$U(\mathbf{y} | \mathbf{z}) = \sum_i \left[u_1(y_i | z_i) + \sum_{(i,j) \in C_2, i \neq j} u_2(y_i, y_j | z_i, z_j) \right] \quad (4.25)$$

To find the expectation value of the log likelihood of this multiple pixels intensity process, let us first define a vector, \mathbf{U}_{m1} , and a matrix, and \mathbf{U}_{m2} , in the same manner as for the single-pixel vicinity case:

$$\mathbf{U}_{m1} = [u_1(y_i | z_i = 1) \quad u_1(y_i | z_i = 2) \quad \dots \quad u_1(y_i | z_i = K)]^T \quad (4.26)$$

$$\mathbf{U}_{m2} = \begin{bmatrix} u_2(y_i, y_j | z_i = 1, z_j = 1) & u_2(y_i, y_j | z_i = 1, z_j = 2) & \dots & u_2(y_i, y_j | z_i = 1, z_j = K) \\ u_2(y_i, y_j | z_i = 2, z_j = 1) & u_2(y_i, y_j | z_i = 2, z_j = 2) & \dots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ u_2(y_i, y_j | z_i = K, z_j = 1) & \dots & \dots & u_2(y_i, y_j | z_i = K, z_j = K) \end{bmatrix} \quad (4.27)$$

where T stands for transposition. We have not denoted the dependence on Φ for notational conciseness. Nevertheless, both \mathbf{U}_{m1} and \mathbf{U}_{m2} are functions only of the true parameters Φ and do *not* depend on any estimated parameters. We can now write the expectation of the state log likelihood (Q) as follows:

$$\begin{aligned}
 E[\log f(\mathbf{y}|\mathbf{z}, \Phi)|\mathbf{y}, \hat{\Phi}^{(p)}] &= E[\log \sum_i f(z_i | z_l, l \in v_i, \Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] \\
 &= E[\sum_i z_i^T \mathbf{U}_{m1}(y_i | \Phi) | \mathbf{y}, \hat{\Phi}^{(p)}] + E[\sum_{(i,j) \in C_2, i \neq j} z_i^T \mathbf{U}_{m2}(y_i, y_j | \Phi) z_j | \mathbf{y}, \hat{\Phi}^{(p)}] \\
 &= \sum_i E[z_i^T | \mathbf{y}, \hat{\Phi}^{(p)}] \mathbf{U}_{m1}(y_i | \Phi) + \sum_{(i,j) \in C_2, i \neq j} E[z_i^T z_j | \mathbf{y}, \hat{\Phi}^{(p)}] \mathbf{U}_{m2}(y_i, y_j | \Phi)
 \end{aligned} \tag{4.28}$$

the quadratic dependence on z in the second term (4.28) is a result of the chosen clique size. Essentially, z_i^T and z_j function together to choose the $(i,j)^{th}$ element from the \mathbf{U}_{m2} matrix since both of them are unit element vectors. To estimate the state of pixel i , we have to calculate the posterior probability of the state of pixel i . Now that we are in multiple pixels vicinity case, we can no longer use (4.23) to calculate the posterior probability of the state of pixel i , equation (4.20). An approximation is in order here because the evaluation of (4.20) is not a recursively computable process, which generally means nonlinear techniques are involved. For computational efficiency, we will approximate (4.20) by :

$$E[z_i = k | \mathbf{y}] \approx \frac{f(y_i | y_j, j \in \eta_i; z_i = k) f(z_i = k)}{\sum_m f(y_i | y_j, j \in \eta_i; z_i = m) f(z_i = m)} \equiv \hat{z}_{ik} \tag{4.29}$$

Essentially, this approximation reduces the intensity dependence to a local neighborhood of pixel i . In [Zhang, et.al., 1994], a detailed discussion of this approximation is offered.

Finally, if we can evaluate the second term of equation (4.28), we would have finished the E-step of the EM algorithm. That second term is a conditional joint expectation of z_i and z_j . In genera, z_i and z_j are dependent if i and j are in the same (second order) clique. Zhang, *et.al.*, (1994) have observed that if z_i and z_j are

approximated to be conditionally independent, good estimations can still be obtained. This is also the approach taken here. i.e.

$$E[z_i = k, z_j = m] \approx \hat{z}_{ik} \hat{z}_{jm} \quad (4.30)$$

Now, just like equation (4.24) for the single pixel vicinity case, we can do a ML estimation of the multiple pixels vicinity model parameters by solving:

$$\sum_i E[z_i^T | \mathbf{y}, \hat{\Phi}^{(p)}] (\nabla_{\Phi_v} \mathbf{U}_{m1}(y_i | \Phi)) + \sum_{(i,j) \in C, i \neq j} (\nabla_{\Phi_z} E[z_i^T \mathbf{U}_{m2}(\Phi) z_j | \mathbf{y}, \hat{\Phi}^{(p)}]) = 0 \quad (4.31)$$

With the parameters for both the intensity and state processes, pixel classification can be done through equation (4.14) and (4.15). We now have MAP classification rules. Incorporating this classifier into MEC of Chapter 3, we now have an optimal Bayesian decision maker, which can potentially be used as an expert in the MEC.

4.4 Some Experiments and Results

This section describes an experiment on the EM based mixture of Gaussian described in this chapter. The problem that this section is concerned with is segmenting the image in Figure 4.2. That image is a phantom image generated with a MRF state prior and a single vicinity Gaussian intensity process. The goal of these experiments is to evaluate the performance of the EM based parameter estimation schemes versus the traditional unsupervised clustering method of K-means.

4.4.1 Creating a Phantom Image

In order to quantitatively evaluate the segmentation results of different segmentation algorithms, the underlying class labels for the input pixels have to be known. The only way this can be done easily is to create a phantom image. Such an image is simulated and is shown on the right hand side of Figure 4.1.

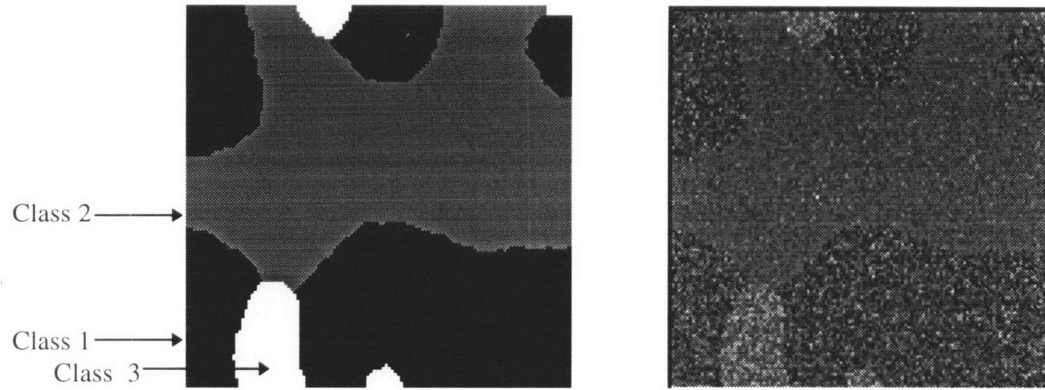


Figure 4.1 The left hand side image is the underlying class label values for the image on the right. This class label image is generated through a Gibbs sampler with $\beta=1.5$ with 2000 iterations. The right hand image is a Gaussian density images. The means and variances of the three different classes are tabulated in Table 4.1.

The right hand image in Figure 4.1 is synthetic image whose underlying class label is generated by a MRF. The underlying class labels are shown in the left hand image of 4.1. Recall that Chapter 2 has described three-level MRF simulation procedure using a Gibbs sampler (see Figure 2.3). The underlying class labels in Figure 4.1 is generated in exactly the same process. The number of iterations for the Gibbs sampler is 2000 while the β value is 1.5 (These parameters are described in [Geman and Geman, 1984]).

The intensity values in the phantom image are sampled from three different Gaussian densities (equation 2.1). The parameters of these three Gaussian densities are tabulated in Table 4.1. below:

<i>Parameters</i>	<i>Means</i>	<i>Variances</i>
<i>Class 1</i>	120	1600
<i>Class 2</i>	150	400
<i>Class 3</i>	180	900

Table 4.1 Gaussian density parameters of Figure 4.2 for the three different classes

One of the advantages of working with phantom images is that we can view the intensity data in the image class by class because we know exactly which class a particular pixel belongs to. We consider the histograms of the three different classes in Figure 4.2. These histograms shows that a difficulty involved in segmenting the phantom image in Figure 4.2 is that the intensity overlap among the three classes is quite severe. Also note

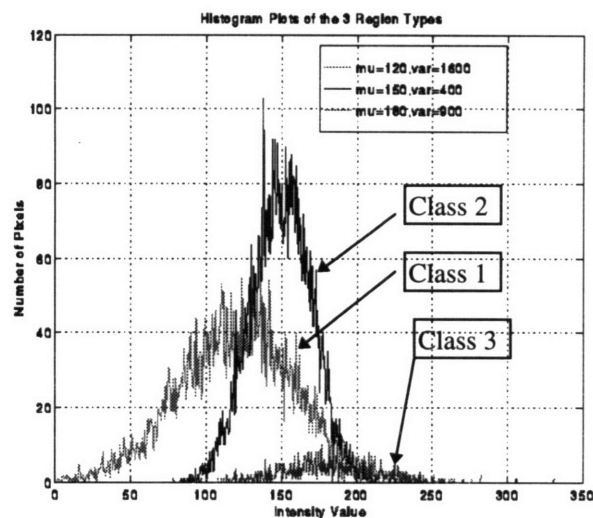


Figure 4. 2 Histogram plots of the three different regions in the phantom image on the right hand side of Figure 4.1. Note the severe overlaps among the three different classes.

the underlying class labels in 4.1 that there are two small regions, one lies at the upper right hand corner while the other one lies near the bottom center. Small regions often create new challenges to a segmenter.

4.4.2 Maximum Likelihood Results

We first consider results obtained by the K-means clustering methods. (K-means was described in Chapter 2 and in more detail in [Fukunaga, 1990].) From an input image, K-means iteratively clusters the intensity values of the image into three clumps according to equation (2.18). This is an unsupervised process. At the end of the clustering, class labels chosen by K-means could be different from what the user wants. Therefore, a

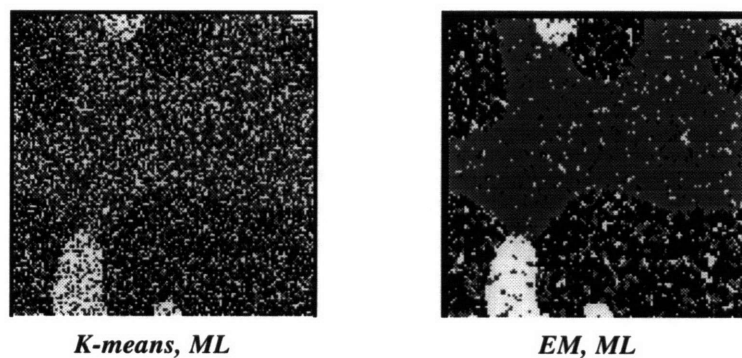


Figure 4.3 ML K-means results on segmenting the Phantom in Figure 4.1. The left hand image is the result after 10 iterations of the K-means. The right hand image is the result after 50 iterations of the K-means.

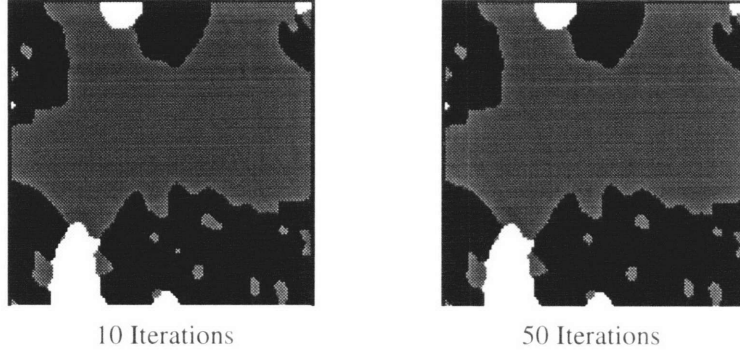


Figure 4.4 MAP K-means results on segmenting the Phantom in Figure 4.1. The left hand image is the result after 10 iterations of the K-means. The right hand image is the result after 50 iterations.

postprocessing step is required to do this class label conversion for the K-means segmentation results.

Figure 4.3 shows the maximum likelihood K-means and EM results (with no spatial priors; refer to Equation 3.1) after both algorithms have gone through 10 iterations of clustering. We can see that the results are not very good. The error of pixel classification is plotted in Figure [4.6].

4.4.3 Maximum A Posteriori K-means Results

The problem with the ML result is that no spatial constraints are imposed on the ML classification results. Let $z_i^{*(ML)}$ be the ML estimate of the class label for pixel i with local region intensities represented by the vector \mathbf{y}_i , then $z_i^{*(ML)}$ is:

$$z_i^{*(ML)} = \arg \max_k f(\mathbf{y}_i | z_i = k) \quad (4.32)$$

$z_i^{*(ML)}$ does not depend on local pixels' class values. As we have seen in Section 4.2, the natural step to take for improving the ML results is to apply a MRF on the class estimates. MRF imposes spatial constraints on the classification results through the Gibbs distribution (4.8). The new class estimate can now be described as:

$$z_i^{*(MAP)} = \arg \max_k f(\mathbf{y}_i | z_i = k) f(z_i | z_j, j \in \eta_i) \quad (4.33)$$

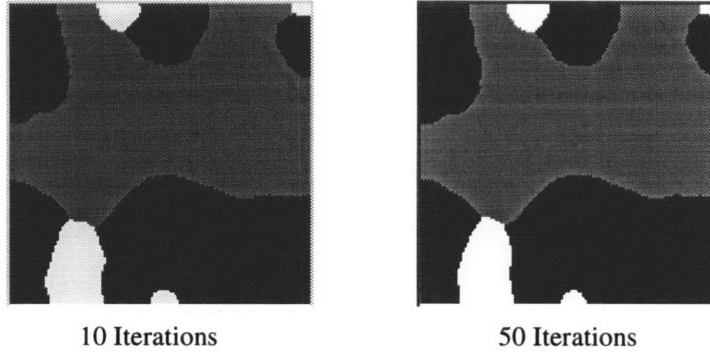


Figure 4.5 MAP results after 10 and 50 iterations of EM for estimating the parameters of the underlying Gaussian densities. Comparisons between these results to those of the K-means results shown that the EM algorithm can more robustly estimate these parameters than K-means can.

This is equivalent a maximum *a posteriori* class estimate. The results of using (4.32) to segment the phantom image in Figure 4.1 are shown in Figure 4.4. These MAP results are definitely much better than the ML results. This big improvement is due entirely to the spatial constraints imposed by the MRF on the class labels.

4.4.4 Maximum A Posteriori EM Results

Finally, we consider the results of this chapter. The EM based unsupervised estimation of the Gaussian parameters are performed using (4.31). The posterior probabilities of the class labels are found using (4.29). These results are shown in Figure 4.5. Clearly the segmentation results in Figure 4.5 are much better than those obtained by K-means algorithm.

Quantitatively, the classification error of the approaches we have seen in this section is plotted in Figure 4.6. Clearly, we can see that ML K-means methods perform worst of all. In comparing the two MAP classification results obtained by K-means and EM, we can see that K-means does not perform as well as EM. Further experiments and results regarding K-means and EM are described in Chapter 8.

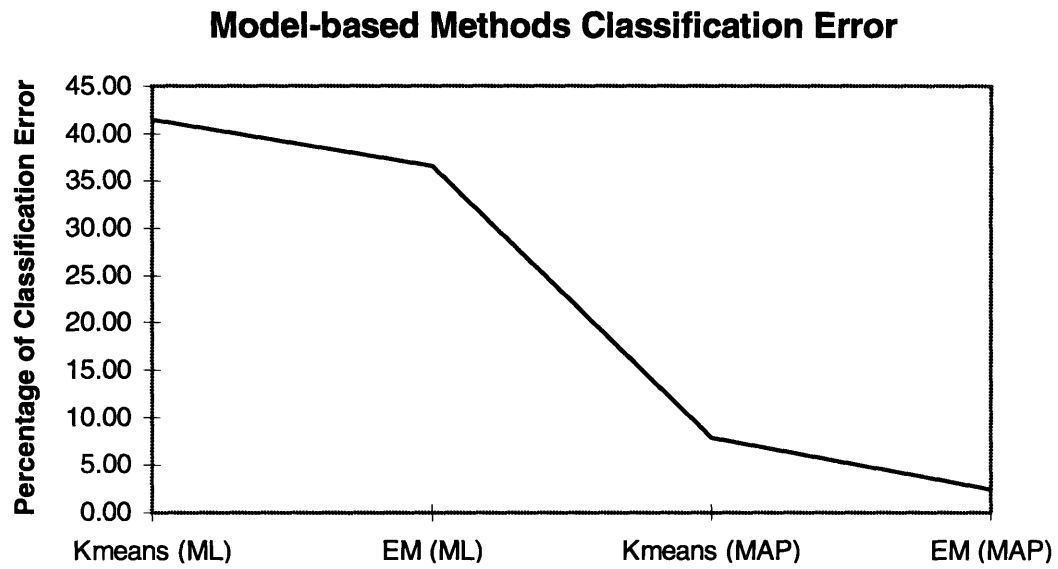


Figure 4.6 Performance evaluation of several model-based pixel classification schemes in the segmentation of image in Figure 4.1.

Chapter 5

FEATURE-BASED CLASSIFICATION METHODS

Chapter 2 of this thesis has briefly discussed the vast amount of feature-based methods for image classification and segmentation. It is probably fair to say that most published techniques on image classification and segmentation are in some way *featured-based* since any discriminating rules have to utilize some kind of “characteristics” of a region to differentiate it from another region. Another name for the “characteristics” is the feature set that represents the region. For example, even the statistical model-based techniques of Chapter 4 can be considered a subset of the feature-based techniques since the actual classification and segmentation are performed using the estimated model parameters, which are the “features” that characterize the region model of interest.

The feature based classifiers developed in this chapter will be used for performance comparison with the Multi-Experts Classifier (MEC) in Chapter 8. In addition, the techniques developed in this chapter for selecting an “optimal” feature set of a given image will also be used in Chapter 8 for providing input features to the gating network version of the MEC. In that respect, the gating network is partitioning the whole feature space provide by the “optimal” features.

Section 5.1 briefly reviews the use of features for image classification and segmentation. An approach to construct a feature based classifier is described in Section

5.2 with discussions on the different types of features used and their estimations. Feature selections, along with classification and segmentation rules, are then discussed Section 5.3. Also presented in Section 5.3 is an extension of the feature-based classifier to include a spatial constraint modeled by a MRF. The chapter is concluded by discussions of some experiments and the results.

5.1 Introduction

For feature-based classification and segmentation techniques, choosing the “right” set of features is clearly the most important step. Numerous researchers have proposed many different types of features, which include spectral features, m^{th} order statistical features, syntactic image features, and sets of “complete” features¹ [Weszka, et.al., 1976; Fu, 1981; Kashyap and Chellappa, 1983; Wilson and Spann, 1988]. Each type of proposed features have their advantages and disadvantages for different types of images. Features that are very good for characterizing textures might not perform so well when applied to cartoon-type images. Features that have been *optimally* chosen for a certain level of granularity might fail miserably when the input image is composed of different sizes of primitive structures².

For a given image, the goal of feature selection is to choose a small set of features that can capture the *essence* of an image for discriminating different pattern classes. A small set of features is desirable for two main reasons. First, redundant features can unnecessarily increase the computations required. Second, as will be shown later on in this

¹ The set of “complete” features is referring to the Finite Prolate Spheroidal Sequences (f.p.s.s.) used by Wilson and Spann (1988) for characterizing stationary images. What is meant by “complete” is that by an appropriate choice of the f.p.s.s. for representing an image, the set of f.p.s.s. can form a basis for the entire space of images. In other words, any image can be reconstructed from its representation in terms of f.p.s.s., in principle. But as Wilson and Spann have pointed out, general this approach is, it is perhaps too general since the appropriate *tessellations* of the spatial and frequency domains is difficult to define. For details of this approach, the reader is referred to [Wilson and Spann, 1988].

² A primitive is the basic pattern which repeats over a texture image.

chapter, a larger feature set does not automatically results in better performance. In fact, a large set of features can deteriorate the performance of a classifier [Fukunaga, 1990; Hertz, et.al., 1991; Ohanian, et.al., 1992]. As Section 5.4 will show through experiments, larger feature sets usually lead to higher classification error (when the number of features is greater than an *optimal* number).

This chapter describes the selection of features for a feature-based classifier, as well as estimations of the feature parameters. This feature based classifier can be used as an expert in the MEC architecture, and is used in Chapter 8. Clearly, more than one feature-based expert can be used for MEC, which might be the desirable step to take. Section 5.2 discusses the choice of several commonly used features. A feature selection scheme is described for selecting a set of *optimal* features. As all researchers in this field would concur, there does not exist a *best* set of features for all images. However, for a particular image, a best subset of all available features can potentially be obtained. The methodologies described in this chapter aims to choose the *optimal* set of features for a given image.

5.2 Features for Classification

The *optimal* features for a given image is selected from the features described in this section according to a prescribed procedure. Not only are these features used for a feature based classifier, they will also be used in Chapter 8 for the inputs to a gating network version of the MEC. In that chapter, two feature-based experts will be used to illustrate the principles behind the multi-experts approach to image segmentation.

The feature-based expert described in this section extracts four main types of features for classification, which include first and second order statistics such as mean and co-occurrence matrix features, SAR and MRF model-based parameters. These features are discussed in the following section. We consider feature selection in the next section.

5.2.1 First Order Statistics

Four first order statistics are considered to be used as features. These statistics are: pixel intensity values, local mean intensity value, local intensity standard deviation, and local median intensity. The definition of these statistics are as follows given an image with N number of pixels:

$$\textbf{Pixel Intensity:} \quad y_i = \{y_1, y_2, \dots, y_N\} \quad (5.1)$$

$$\textbf{Local Mean Intensity:} \quad \mu_i = \frac{1}{N_{\eta i}} \sum_{j=1}^{N_{\eta}} y_j \quad (5.2)$$

$$\textbf{Local Variance:} \quad \sigma_i^2 = \frac{1}{N_{\eta i}} \sum_{j=1}^{N_{\eta}} (y_j - \mu_i)^2 \quad (5.3)$$

$$\textbf{Local Median Intensity:} \quad m_i = \{m_1, m_2, \dots, m_N\} \quad (5.4)$$

where $N_{\eta i}$ is the number of local neighbors of pixel i , and m_i is calculated by taking the median value of the local neighbors of pixel i . ($N_{\eta i}$ is sometimes denoted by the cardinality $CARD(S)$ where S is the image lattice.) Equivalent definitions of these first-order statistics can be given in terms of the normalized histogram [Carstensen, 1992; Pitas, 1993].

For ease of explaining the feature selection process in the next section, these first order statistics are given feature numbers as shown in Table 5.1.

Feature	Pixel Intensity	Local Mean	Local Std Dev	Local Median
Number	0	1	2	3

Table 5.1 First Order Statistics and their number assignments for classification

5.2.2 Co-occurrence Matrix Features

Co-occurrence matrix features have been found to be among the best features for textural image classification [Weszka, et.al., 1976; Haralick, et.al., 1979; Ohanian, et.al, 1992]. These features have been found to outperform a variety of other features such as

run-length statistics, gray-level difference statistics, power spectrum features, fractal features, Garbor filter features and Markov random field features.

The co-occurrence matrix has already been presented in Chapter 2. The set of co-occurrence matrix features that are used for feature selection are shown in Table 5.2. A total of four types of co-occurrence matrices are used, and they are $(d, \theta) = \{ (1, 0^\circ), (1, 45^\circ), (1, 90^\circ), (1, 135^\circ) \}$, where d stands for pixel displacement³ and θ stands for the direction of the displacement. From these four matrices, we extract five features from each one. These five features have been found by several researchers to be effective co-occurrence features [Weszka, et.al., 1976; Haralick, et.al., 1979; Ohanian, et.al., 1992]. Assume the co-occurrence matrix coefficients are represented by $c_{ij\theta}$'s, which are normalized frequencies that pixels with intensity value i in the sub-image of interest are neighbors to pixels with intensity value j . The intensity range has been divided into G different bins, of which we have chosen $G = 8$ ⁴.

$$\text{Entropy (ENT):} \quad -\sum_{i=1}^G \sum_{j=1}^G c_{ij\theta} \log(c_{ij\theta}) \quad (5.5)$$

$$\text{First Order Contrast (1CON):} \quad \sum_{i=1}^G \sum_{j=1}^G |i - j| c_{ij\theta} \quad (5.6)$$

$$\text{Second Order Contrast (2CON):} \quad \sum_{i=1}^G \sum_{j=1}^G (i - j)^2 c_{ij\theta}^2 \quad (5.7)$$

$$\text{Angular Second-Moment (ASM):} \quad \sum_{i=1}^G \sum_{j=1}^G c_{ij\theta}^2 \quad (5.8)$$

³ The displacement for diagonal directions (45° and 135°) is actually $\sqrt{2}$ but for the sake of simplicity, as most researchers do, we take the displacement to be 1.

⁴ G is an important parameter of the co-occurrence features. We will show in a later section that $G = 8$ is a good tradeoff between computational speed and accuracy.

$$\text{Correlation (COR):} \quad \sum_{i=1}^G \sum_{j=1}^G \frac{(ijc_{ij\theta} - \mu_r \mu_c)}{\sigma_r \sigma_c} \quad (5.9)$$

where in the definition of the correlation feature, μ_r and μ_c are the means of the row and column normalized co-occurrence coefficients, σ_r and σ_c are the corresponding standard deviations of the means. For a given pixel i , the co-occurrence matrices calculations are done over a region of size $(2L+1) \times (2L+1)$, where integer L is dependent the particular image of concern. This size of the region is determined through tradeoff between computational speed and accuracy. In the case of border values, the smaller the region size, the more pixels there are with similar features because the smaller region size make fewer pixels with regions *touch* the boundaries. Therefore, although larger regions are better for image recognition and classification tasks than smaller regions, arbitrarily large region is not necessarily more conducive to better image segmentation results than smaller region size is. This additional tradeoff results in our choice of 17×17 . Experimental results for this choice will be shown in Section 5.4.

Before co-occurrence matrices are calculated, we found that if histogram equalization is performed on the image, pixel classification results are superior to results obtained without histogram equalization. We believe this observation is caused by the *spreading* effect of histogram equalization on the coefficients of the co-occurrence matrices. Histogram equalization spreads the intensity values across the entire range available to an input image [Lim, 1990; Gonzalez and Woods, 1992; Pitas, 1993], which enhances the dynamic range of an image, thereby making subtle details clearer than without equalization.

Again, just as the first-order statistics, these co-occurrence features are given feature numbers for ease of explaining the feature selection process in the next section. These feature number assignments are shown in Table 5.1.

<i>Co-occur. Matrix</i>	<i>(1, 0°)</i>	<i>(1, 45°)</i>	<i>(1, 90°)</i>	<i>(1, 135°)</i>
<i>Feature Number</i>	4, 5, 6, 7, 8	9, 10, 11, 12, 13	14, 15, 16, 17, 18	19, 20, 21, 22, 23

Table 5.2 Co-occurrence matrix features and their respective feature numbering⁵.

5.2.3 Gaussian Markov Random Field Features

Markov random field (MRF) features have been used by various researchers for image classification (especially texture) and simulation, examples are [Hassner and Sklansky, 1980; Cross and Jain, 1983; Chellappa, et.al., 1985; Zhang, et.al, 1994]. There are two commonly used MRF models that people use for grayscale images, the binomial and the Gaussian MRF (GMRF). The latter model is used in this thesis because of its ability to simulate images that are very realistic, especially in the medical domain. Some example of simulated GMRF images are shown in Figure 5.1. The parameters in the captions are for a second order symmetric toroidal GMRF with eight neighbors. A GMRF can be characterized by its conditional probability density [Kashyap and Chellappa, 1983]:

$$f(y_i | y_j, j \in \eta_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2} (y_i - \sum_{j \in \eta_i} b_j y_j)^2} \quad (5.10)$$

where η_i is the local neighbors of pixel i with intensity y_i . σ_i^2 is the local intensity variance. Figure 5.1 is generated through the following conditional Markov model equation, assuming that the observed intensity values $\{ y_i, i \in S \}$ of image lattice S is zero mean (which can always be satisfied by shifting all pixels by a constant value):

$$y_i = \sum_{j \in \eta_i} b_j y_j + e(i) \quad (5.11)$$

⁵ The ordering of the features within each co-occurrence matrix is the same as the feature definitions in equation (5.5) - (5.9).

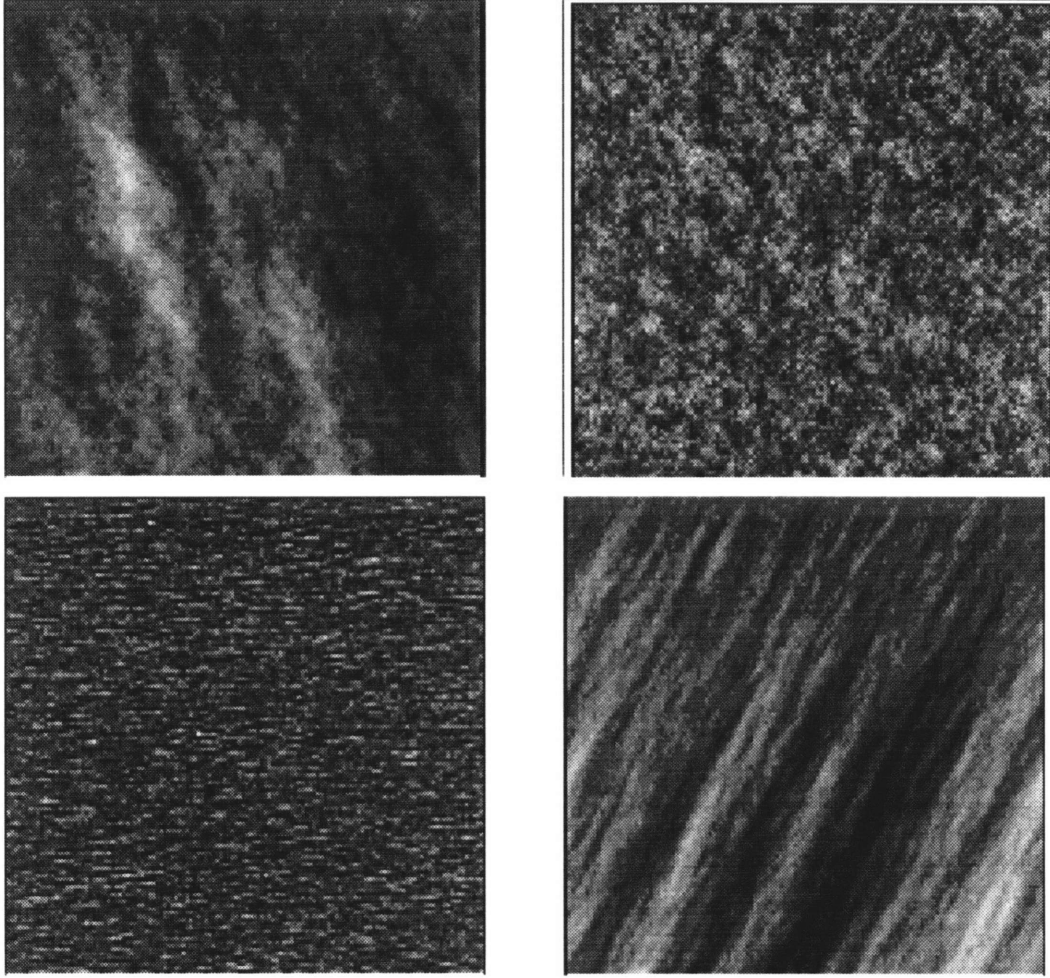


Figure 5.1 Simulated textural images using Gaussian Markov random fields. The upper left image has parameters: $\{0.25, 0.25, 0.25, 0.25\}$; the upper right image has parameters: $\{0.2, 0.2, 0.2, 0.2\}$; the lower left image has parameters: $\{0.6, -0.1, 0.0, 0.0\}$; the lower right image has parameters: $\{0.0, 0.5, 0.5, 0.0\}$. These are images generated by equation (5.11).

$b_{-1,-1}$	$b_{0,-1}$	$b_{1,-1}$
$b_{-1,0}$	y_i	$b_{1,0}$
$b_{-1,1}$	$b_{0,1}$	$b_{1,1}$

Figure 5.2 Second order neighborhood of a Gaussian Markov random field (GMRF). b_{ij} represents the parameters of such a model. We consider a symmetric toroidal GMRF, which means that $b_{ij} = b_{-i,-j}$.

where $e(i)$ represents an innovation process that generates Gaussian noise with the following properties:

$$\begin{aligned} E[e(i) | y(j), j \neq \eta_i] &= 0 \\ E[e^2(i)] &= \sigma^2 \end{aligned} \quad (5.12)$$

Woods (1972) shows that the infinite lattice version of (5.11) generates a series of y_i that satisfies the Markov condition that $f(y_i | y_j, j \neq i) = f(y_i | y_j, j \in \eta_i)$.

The parameters of the GMRF (b_j 's) satisfy the symmetry condition that $b_j = b_{-j}$, where b_j and b_{-j} are diametrically across from each other. An example of second order GMRF neighborhood is shown in Figure 5.2. To use these parameters as features for pixel classifications, these parameters have to be estimated. However, because of the intractable normalization factor of the MRF [Besag, 1972], calculating the exact parameter estimates are virtually impossible in practice. Besag (1972) has proposed a coding method for estimating these parameters in an approximate maximum likelihood fashion. This coding method has been discussed in the background chapter of this thesis. Hassner and Sklansky (1980) have used the coding method to estimate the parameters of binary Markov random field. Cross and Jain (1983) have also used this coding method to estimate the parameters of a MRF with intensity values described by a binomial distribution. The coding method seems to give good results for many applications, as attested by the number of researchers who have successfully used it. Unfortunately, Kashyap and Chellappa (1983) have pointed out that the coding method is not an efficient and not a consistent estimation scheme for a non-causal system such as the second order GMRF in Figure 5.2. This thesis follows their proposal for a more consistent least squares estimation scheme than the coding method, which is the following least-square estimate⁶:

⁶ Although this is an improvement, this estimate is still not consistent as pointed out by Grunkin (1993). A consistent estimate requires the use of non-linear techniques, which means the computation could be very burdensome [Grunkin, 1995].

$$\mathbf{b}^* = \left[\sum_{i \in \Omega_I} \mathbf{q}(i) \mathbf{q}^T(i) \right]^{-1} \left(\sum_{i \in \Omega_I} \mathbf{q}(i) y_i \right) \quad (5.13)$$

where \mathbf{b}^* stands for a vector with all GMRF parameters, in our second order case, there are four parameters, as shown in Table 5.3 below. $\mathbf{q}(i)$ is a column vector = col [$y_{i+r} + y_{i-r}$; $r = (0, 1), (1, 0)$]⁷. The indexing of the pixels $i \in \Omega_I$ is defined through another Ω_B , viz.:

$$\begin{aligned} \Omega_B &= \{i : i \in S \text{ and } (i+r) \notin S, \text{ for at least one } r \in \eta_i\} \\ \Omega_I &= S - \Omega_B \end{aligned} \quad (5.14)$$

where S is the whole set of lattice points of the image. Ω_B represents all the boundary points while Ω_I represents all the interior points which do not contain any neighbors outside S .

We assign feature numbers in the same fashion as we do for the first order statistics and the co-occurrence cases. These assignments are shown in Table 5.3. b_0 is equivalent $b_{1,0} = b_{0,1}$ in Figure 5.2. The same equivalency applies to $b_1 \Leftrightarrow b_{0,-1} = b_{0,1}$, $b_2 \Leftrightarrow b_{-1,1} = b_{1,-1}$, and $b_3 \Leftrightarrow b_{-1,-1} = b_{1,1}$.

<i>Feature</i>	<i>Horizontal (b_0)</i>	<i>Vertical (b_1)</i>	<i>Diagonal 1 (b_2)</i>	<i>Diagonal 2 (b_3)</i>
<i>Number</i>	24	25	26	27

Table 5.3 GMRF feature number assignments for classification purpose.

5.2.4 Simultaneous Auto-Regressive Features

Just like Markov random field (MRF) features, Simultaneous Auto-Regressive (SAR) model features are commonly used by various researchers for textural image classification and simulation, examples are [Kashyap, et.al., 1982, 1986; Chellappa, et.al., 1985; Grunkin, 1993; Zhang, et.al, 1994; Hu, et.al, 1994]. SAR models can be divided

⁷ In reality, the GMRF parameters are estimated by solving a linear equation, which is an equivalent procedure as equation (5.13). Expliciting, the linear equation is: $\left[\sum_{i \in \Omega_I} \mathbf{q}(i) \mathbf{q}^T(i) \right] \mathbf{b}^* = \sum_{i \in \Omega_I} \mathbf{q}(i) y_i$

into causal and non-causal models. Causal SAR models contain neighborhood definitions that are recursively computable [Lim, 1990], while non-causal SAR models contain non-recursively computable neighborhood systems. Causal models are preferred over non-causal models because of two main reasons. First, the parameter estimations can be performed using a least-squares method, which provides a consistent estimate of the parameters. For non-causal models, although some researchers also applied the least-squares method [Kashyap and Khotanzad, 1986; Mao, *et.al.*, 1992] for parameter estimation, this method does not provide a consistent estimate of the parameters [Kashyap and Chellappa, 1983; Grunkin, 1993]. The second reason for favoring the use of causal model is that a consistent parameters of a non-causal model require non-linear techniques. (Such a scheme has been proposed by [Grunkin, 1995]. Such non-linear techniques take much more computation than least-squares methods.

There are two main types of causal SAR models, the ones with quadrant support, or quarter-plane models (QP), and the non-symmetrical half-plane (NSHP) models [Lim, 1990; Grunkin, 1993]. Most researchers would agree that NSHP models outperform QP models in almost all cases. This thesis therefore uses the causal NSHP to derive features for pixel classifications. Some examples of simulated textural images using such models are shown in Figure 5.3. (Note that the simulated textural images use a NSHP model that has only a 4 neighbors system. The actual feature extraction is done using a NSHP SAR model that has 12 neighbors.)

Let's first consider the support systems for the SAR models. Figure 5.4 illustrates a four neighbors NSHP system that is used to simulate the textural images in Figure 5.3, as well as the twelfth neighbors system that is used to extract features for pixel classification later on in this chapter. In a finite image lattice S , the intensity values of a SAR textural image can be described by the following generative equation for all intensity values $\{ y_i, i \in S \}$ [Kashyap and Chellappa, 1983]:

$$y_i = \sum_{j \in \eta_i} d_j y_j + \sqrt{\rho} w(i) \quad (5.15)$$

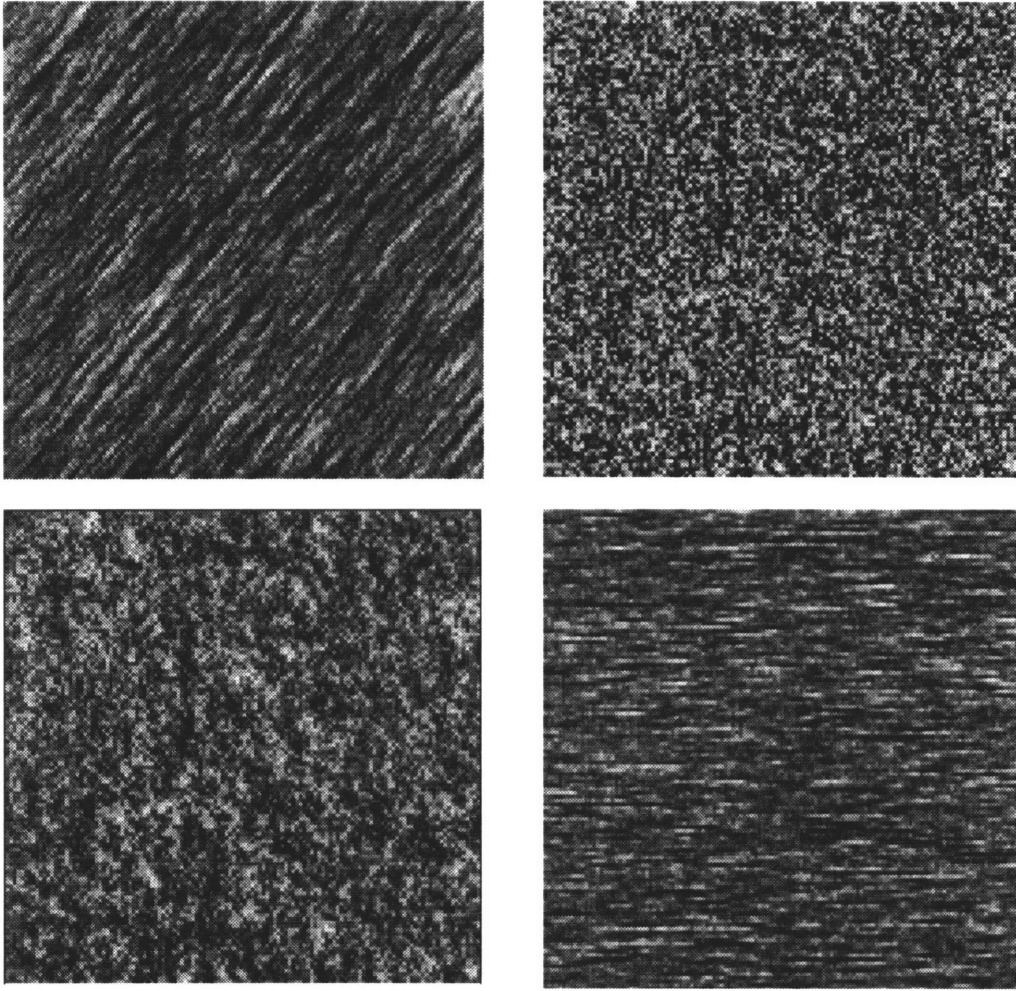
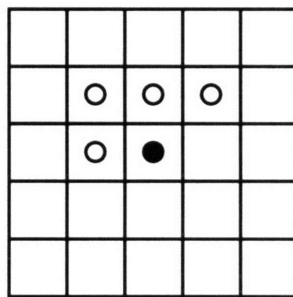
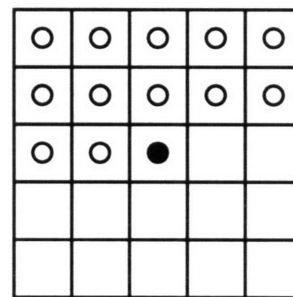


Figure 5.3 Simulated Simultaneous Auto-Regressive (SAR) textural images. The SAR model is one of NSHP with four neighbors (this model an instance of the eight neighbor NSHP with 12 neighbors). The upper left image has parameters: $\{0.05, -0.05, 0.2, 0.7\}$; the upper right image has parameters: $\{0.1, 0.1, 0.1, 0.1\}$; the lower left image has parameters: $\{0.2, 0.2, 0.2, 0.2\}$; the lower right image has parameters: $\{0.8, 0.0, 0.0, 0.0\}$. These are simulated by applying equation (5.15).



4-neighbors NSHP



12-neighbors NSHP

Figure 5.4 Two types of NSHP support systems used in SAR models. Dark circles represent the current pixel of concern; white circles represent the neighbors of the center pixel. The 4-neighbors NSHP has been used to simulate the textural images in Figure 5.3 while the 8-neighbors NSHP is used later to extract features for classification.

where ρ and d_j 's are unique parameters for each SAR textural images. η_i is the local neighbors of pixel i with intensity y_i . Figure 5.3 is generated through this equation with a four neighbors NSHP model in Figure 5.4. Note that equation (5.15) is remarkably similar to equation (5.11) for generating GMRF textural images. However, as cautioned by Kashyap and Chellappa (1983), the intensity values y_i 's generated through (5.15) do not satisfy the Markov property. i.e. $f(y_i|y_j, j \neq i) \neq f(y_i|y_j, j \in \eta_i)$. We use least-squares (LS) to estimate the NSHP SAR parameters. These SAR parameters would be part of the feature set being considered for the *optimal* features.⁸

$$\mathbf{d}^* = \left[\sum_{i \in S} \mathbf{z}(i) \mathbf{z}^T(i) \right]^{-1} \left(\sum_{i \in S} \mathbf{z}(i) y_i \right) \quad (5.16)$$

$$\rho = \frac{1}{N} \sum_{i \in S} (y_i - \mathbf{d}^{*T} \mathbf{z}(i))^2$$

where the vector $\mathbf{z}(i)$ is a column vector: $\mathbf{z}(i) = \text{col} [y_{i+r}; r \in \eta_i]$. Recall that local variance is our Feature #2, so, we do not use the ρ estimate in (5.16). We assign feature numbers to the SAR features in the same fashion as we do for the first order statistics, the co-occurrence and the GMRF features. These assignments are shown in Table 5.4.

<i>Feature</i>	<i>SAR Model Coefficients</i>
<i>Number</i>	28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38

Table 5.4 SAR feature number assignments for classification purposes.

5.3 Features Selection and Classification

With the features described in the last section, this section illustrates a feature selection procedure to get the “best” set of features for classifying both synthetic and real-

⁸ As several researchers have pointed out, these SAR parameters are not really robust since they are sensitive to rotation of the image, additive noise, and small perturbations of the textural image [Kashyap, et.al., 1986; Mao, et.al., 1992; Grunkin, 1993]. However, for illustration of feature selection of the feature-based expert in the MEC, this feature set is adequate. Current effort is under way to extract robust parameters such as the spectral density features as discussed in [Grunkin, 1993].

world images. The feature selection procedure is performed by a Whitney feature extractor [Whitney, 1971; Ohanian, 1992]. Then, two classification schemes are presented, along with extensions of these schemes to fit the MEC approach.

5.3.1 Selecting the Optimal Feature Set

To evaluate the quality of a set of features chosen for a given image, we need some example sets for estimating the parameters as well as for comparing them. Let's call the example learning set with E_L number of learning sets as $\chi_L = \{\mathbf{y}_i, \mathbf{z}_i\}$, where $i \in \{1, 2, \dots, E_L\}$, and the example testing set with E_T number of testing sets as $\chi_T = \{\mathbf{y}_i, \mathbf{z}_i\}$, where $i \in \{1, 2, \dots, E_T\}$.

The goal of the feature selection stage is to choose the optimal set of features \mathcal{F}^* that minimizes certain cost function. So, our first goal is to determine the appropriate cost function for selecting the features. For illustrating the feature selection process, consider the example images of interest are the 256 x 256 pixels images shown in Figure 5.5. The left image is a mosaic of constant regions with additive Gaussian noise of different variances. The image on the right is a mosaic image composed of four Brodatz textures (D29, D57, D93, D100). The optimal pixel classifier would generate four equal square regions for each image, which would give zero classification error. Since we know *a priori* the true class of each pixel, we can formulate the our goal as followed:

$$\arg \min_{\mathcal{F}} \left\{ \mathcal{E} = \sum_{i=1}^{E_T} \delta(\mathbf{z}_i^*, \mathbf{z}_i) \right\} \quad (5.17)$$

where $\delta(\cdot)$ is the Kronecker delta. Equation (5.17) essentially states our goal as minimizing the total number of misclassified test pixels. The example learning sets (χ_L) are chosen randomly (such as by a user with a mouse) and are used for estimating the parameters. For each class, the user randomly chooses 20 points in the region which belongs to that class. The test set χ_T is selected similarly. However, for some experiments such as the Brodatz mosaic image in Figure 5.5, test sets of less than 100 are inadequate for comparing the performance of different feature sets. Therefore, in comparing feature sets in these cases, a

large testing set is required for meaningful results. Large testing sets take a lot of manual labor to collect. This thesis has used an automatic random process for selecting 1024 points in the entire image for testing (or 256 test points per class). Since the underlying class memberships are known *a priori*, such automatic selections can easily be done.

Feature selection is done in a serial manner. Consider a total of N features. The first step in feature selection is to individually estimate the feature values for different classes and for testing the estimated features using the testing set χ_T . The one feature with the lowest testing error \mathcal{E} is selected as the first optimal feature F_1 in \mathcal{F}^* . The next feature F_2 is chosen from among the remaining $(N - 1)$ features which, in combination with the first feature F_1 , produces the next smallest testing error \mathcal{E} , which is usually smaller than any single feature can attain. The next feature is chosen similarly. This process continues until the minimal testing error is reached. An important point to be observed here is that the second best (individual) feature in the first feature selection process might not be the second optimal feature F_2 selected in combination with the first optimal feature. This result might at first seems strange until one realizes that the first optimal feature might in some way contain much of the information in the second best (individual) feature. This point will become clear when the results are presented later on in this chapter.

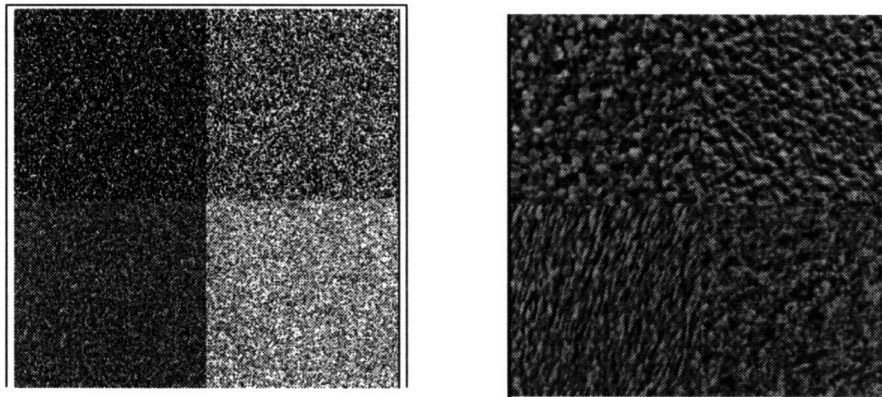


Figure 5.5 These two sample 256 x 256 images are used to evaluate the performance of the feature-based classifying expert. The left image is a mosaic of four constant patches with additive Gaussian noise of various magnitudes of variance. (The patches have respectively, from top-left patch clockwise, mean 120 and variance 900, mean 140 and variance 400, mean 160 and variance 2500, mean 180, variance 900.) The right hand side sample textural. This image is mosaic of four Brodatz textures, which are real life textures (D29, beach sand, D57, handmade paper, D93, fur, D100, ice crystal).

5.3.2 Pixel Classification

Once the set of optimal features has been found for a given image, we can perform pixel classifications. We consider a maximum likelihood approach here. The next section presents a maximum *a posteriori* approach. Before we begin the classification discussion, let's rethink what we want of an optimal feature set. Such a feature set should be one that captures the *essense* of an image region very well. In other words, just from the feature set, one can tell which image region is represented. The quantification of "well" can be viewed from the standpoint of a classification problem. Given just the feature set of an image, how likely could we identify the original image amid many other images? If the answer is extremely likely, then the feature set is very good for representing that image. So, for classifying a pixel, given its local region, we need to find the best *prototype* feature set \mathcal{F}_k that represents the local region well. (\mathcal{F}_k can be characterized by a mean vector μ_k and a covariance matrix Σ_k). This last statement is then the goal of the feature-based classification methods presented here.

Now, we have the problem of comparing the feature set \mathcal{F}_i of a subimage local to pixel i to several prototype feature sets $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_K$, and find the prototype feature set that is closest to \mathcal{F}_i . For this problem, we need a distance measure for comparing different feature sets. A commonly used distance measure is the Euclidean distance, $\|\mathcal{F}_i - \mathcal{F}_k\| = \sqrt{|\mathcal{F}_i - \mathcal{F}_k|^2}$. As illustrated by Figure 5.6, the Euclidean distance measure does not take into account the correlation between elements of the feature vectors. A more sensible measure than the Euclidean distance for pattern recognition is the Mahalanobis distance, which is defined as [Fukunaga, 1990]:

$$\|\mathcal{F}_i - \mathcal{F}_k\| = (\mathcal{F}_i - \mu_k)^T \Sigma_k^{-1} (\mathcal{F}_i - \mu_k) \quad (5.18)$$

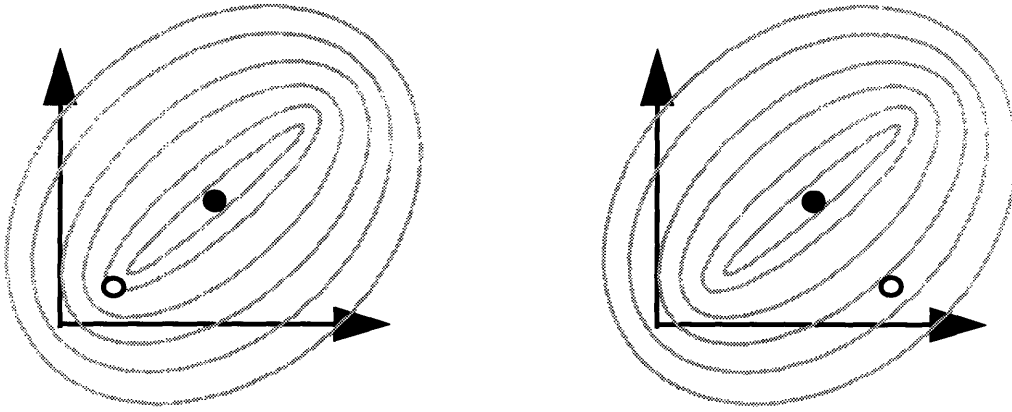


Figure 5.6 These two sketches are for illustrating why Mahalanobis distance measure is more sensible than Euclidean distance measure for pattern recognition tasks. Assume that we are interested in measuring the similarity between the two points shown in each graph (black and white dots). The ellipses represent iso-contours for the equiprobable lines of the two identical density functions shown. Clearly, if Euclidean distance measure is used, the black and white points would have two very similar distances. However, if Mahalanobis distance measure is used, the two points in the left hand graph would have a much smaller distance than those on the right hand side because of the use of the density covariance matrix.

where μ_k and Σ_k^{-1} are the mean vector and the inverse of the covariance matrix associated with class k , respectively. The mean vector and the covariance matrix are estimated from the example learning set χ_L . The Mahalanobis distance is a better distance measure for multi-variate problems than Euclidean distance because Mahalanobis distance takes into consideration the correlation between different elements of the feature vector through the covariance matrix. As illustrated in Figure 5.6, the Euclidean distance between two feature sets (represented by dots in the figure) might not represent the actual similarity between the two feature sets very well while a Mahalanobis distance measure differentiates the two cases clearly.

With an appropriate distance metric, the function that specifies the likelihood a given feature set \mathcal{F}_i (for pixel i) belongs to a particular class can now be made explicit. Notice that the Mahalanobis distance measure is the same as the exponential factor in a multi-variate Gaussian density function. Essentially, by using the Mahalanobis distance, the parameters of interest are being modelled as Gaussian distributed. Earlier we mentioned that model-based methods can be considered feature-based methods. Here, we see that actually, feature-based methods can equally be considered as model-based methods. The

model we use is the Gaussian density. The function that specifies the likelihood of a given pixel to class k is then:

$$p(y_i|z_i = k, \Phi) = \frac{1/\sqrt{|\Sigma_k|} e^{\{-(\mathcal{Z}_i - \mu_k)^T \Sigma_k^{-1} (\mathcal{Z}_i - \mu_k)\}}}{\sum_{j=1}^K 1/\sqrt{|\Sigma_j|} e^{\{-(\mathcal{Z}_i - \mu_j)^T \Sigma_j^{-1} (\mathcal{Z}_i - \mu_j)\}}} \quad (5.19)$$

Equation (5.19) is the likelihood we use to classify a given pixel i with its local subimage as input. The classification rule is a maximum likelihood approach:

$$z_i^* = \arg \max_{z_i} p(y_i|z_i, \Phi) \quad (5.20)$$

This classification rule does not take into account the spatial correlation between adjacent pixels. In the next section, a better classification rule will be presented in terms of a MAP approach that explicitly takes into consideration the spatial correlation between adjacent pixels.

5.3.3 Enhancing the Feature-based Classifier for MEC

As noted in Chapter 3, MEC takes the *a posteriori* probability vectors from the available experts. The previous section on pixel classification describes a maximum likelihood (ML) classification approach, yielding the probability vector $p(y|z, \Phi)$, which is not the maximum *a posteriori* (MAP) probability vector $p(z|y, \Phi)$ that MEC needs. (Φ represents all the parameters of interest.) So, we need to somehow transform the ML results we have so far to a better one -- the MAP solution.

Image segmentation (or pixel classification) is an ill-posed problem [Poggio, 1986; Marroquin, 1987]. Given an input image, finding the homogeneous regions can be done in many ways. To *regularize* this problem, one way is to apply spatial constraints. To restate the problem in the previous paragraph, the ML pixel classification scheme presented in the last section does not take into account the spatial correlation between neighboring pixels. In order to improve the results, we have to impose some spatial constraints on the

classification results. This sounds like the problem we were trying to solve in the previous chapter on model-based methods. As discussed in that chapter, an effective way to impose spatial constraints on the classification results is to construct a prior based on the maximum likelihood results. As was also discussed in that chapter, this prior can take a variety of forms. A conditional Markov (CM) process has been shown to be an effective and efficient prior. Therefore, a CM process is used here for imposing the spatial constraints on the classification results.

Consider a given pixel i with an intensity conditional probability function $p(y_i|z, \Phi)$, we would like to find the posterior probability of pixel i in class z_i^* . Our classification procedure is to find the class z assignments that maximize the familiar MAP equation:

$$z^* = \arg \max p(y|z, \Phi)p(z|\Phi) \quad (5.21)$$

As pointed out by Julian Besag (1986), such maximization is extremely unwieldy due to the awkward normalization factor generally associated with the state (or class) priors. So, we take the same approach as he did for his Iterated Conditional Mode (ICM) algorithm -- by the use of an approximation technique [Besag, 1986]. In explicit terms, we find the set of state assignments that individually maximizes the following:

$$\begin{aligned} z_i^* &= \arg \max \{p(z_i|y, z_j, j \neq i)\} \\ &= \arg \max \{p(y_i|z_i)p(z_i|z_j, j \in \eta_i)\} \end{aligned} \quad (5.22)$$

where just as in Chapter 4, η_i is the neighborhood of pixel i . $p(y_i|z_i)$ can be found using the Mahalanobis distance measure given in the last section. A pairwise interaction Markov process is used to model $p(z_i|z_j, j \in \eta_i)$ in the following way:

$$p(z_i = k|z_j, j \in \eta_i) = \frac{1}{Z_n} e^{\left\{ \beta \sum_{j \in \eta_i} \delta_{i,j}(k) \right\}} \quad (5.23)$$

where Z_n is the so called partition function for normalization. $\delta_{i,j}(k)$ is the Kronecker delta which is equal to 1.0 when the neighboring pixels i and j have the same class value of k ,

and 0.0 otherwise. β is usually chosen to be around 1.5, which has been found to work well with many types of images. As pointed out by Besag (1986), this value is not crucial to the final outcome. In other words it can take a wide range of values without affecting the final outcomes.

The classification scheme expressed by equation (5.22) is a maximum *a posteriori* (MAP) approach since the right hand side is exactly the *a posteriori* probability of the pixels in class \mathbf{z}^* . The resulting posterior probability vector $p(\mathbf{z}|\mathbf{y},\Phi)$ is the contribution of this feature-based classification expert to the Multi-Experts Classifier of Chapter 3.

5.4 Some Experiments and Results

This section describes two kinds of experiments. One type is conducted to choose the parameters of the feature-based expert classifier. The other type is for demonstrating the image classification and segmentation using the estimated classifier parameters and the *optimal* set of image features.

5.4.1 Estimating Classifier Parameters

In addition to the optimal features set selected for classification, the parameters of the co-occurrence matrix features are also estimated using the Whitney procedure described in the last section. There are two main parameters to be estimated for co-occurrence matrix features. First, the dimension G of the co-occurrence matrix, a square matrix, is to be chosen. G represents the number of graylevel bins for dividing the intensity values into. Second, the local region size $S_L = (2L + 1)^2$ for calculating the co-occurrence matrix has also to be determined. L is the *one-sided* neighborhood length of a given pixel (which is located in the middle of the region). In the following illustration of how G and L are found for a given image, the Brodatz mosaic image in Figure 5.5 is used. The following procedure can be applied to any type of image.

Figure 5.7 plots the *minimum* classification error on the Brodatz mosaic image as a function of the dimension of the co-occurrence matrix. What is meant by “minimum” is that the classification error is obtained with the optimal (co-occurrence) feature set with the estimated parameters (G and L). The optimal feature set is selected through the Whitney feature extractor, exactly as described in the last section. (An example of the feature selection process using only the co-occurrence matrix features is plotted in Figure 5.9, which will be discussed shortly.) The region size used in Figure 5.7 is 41 x 41 pixels, the selection of which will be discussed in the next experiment. As shown in Figure 5.7, the lowest classification error belongs to a G value of 8, although the margin of error for this G estimate is definitely great enough to prevent laying any claim to this G value as being the best of all values. In fact, there seems to exist a plateau in error curve of Figure 5.7 between values of $G = 8$ and $G = 24$, which suggests that G can take a pretty wide range of values without significantly degrade the classification performance of the co-occurrence feature-based classifier. Since $G = 8$ is on the lower part of this range, from a computational point of view, choosing $G = 8$ for the test image (Figure 5.5) has the additional advantage of reducing the computation requirement.

The next figure, Figure 5.8, shows the minimum classification error as a function

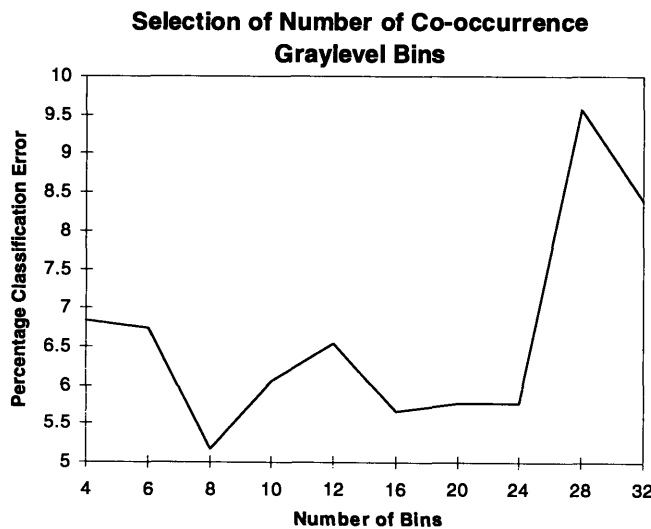


Figure 5.7 Plot of classification error on the test set χ_T belonging to Figure 5.5 as a function of the G value (number of Graylevel bins) of the co-occurrence matrices.

of the region size ($S_L = (2L + 1)^2$). (The plot again shows the result for the Brodatz mosaic image in Figure 5.5.) For small regions, the classification performance of the co-occurrence features degrades because the co-occurrence matrices cannot be estimated consistently from one region to another belonging to the same class. Recall that a co-occurrence matrix records the probability of *co-existence* of two pixels with two given intensity values. The larger the region for estimation, the more accurate the estimate for these probabilities.

At the same time, the minimum classification error should also increase as the region size becomes very large due to boundary effects. Larger region size, no doubt, is good for recognizing homogeneous and stationary regions. However, given the goal is segmentation of an input image with multiple region classes, larger region size is a disadvantage. This point should be clear when one considers pixels near boundaries between different region types. The extreme case is when the region size is on the order of the image size. In that case, the co-occurrence matrices would be used to describe the whole image composed of different image types. Clearly, the classification performance suffers greatly. Therefore, larger region size does not necessarily mean better performance for image segmentation, contrary to image recognition tasks.

From Figure 5.8, we see that after L gets larger than around 16, the classification error stays at around 5-6 percent. For region size with $L > 24$, the computation required is extremely high. Balancing accuracy and computational speed, we choose $L = 20$ as the region size parameter for this image.

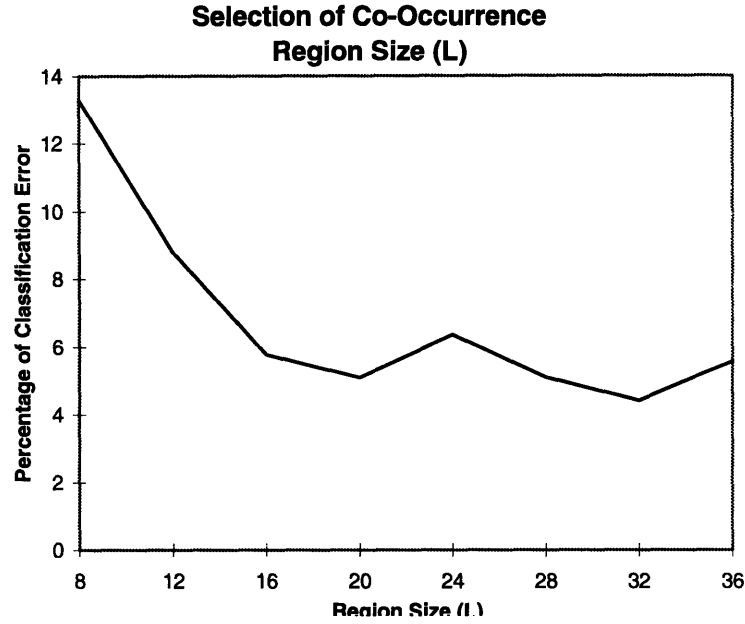


Figure 5.8 Plot of classification error on the test set χ_T as a function of the region size used for calculating the co-occurrence matrices.

We have been talking about the *optimal feature sets* for choosing the co-occurrence matrix features throughout this section. An example of how this optimal feature set is chosen is shown in Figure 5.9, which is plotted for parameter values of $G = 8$ and $L = 20$. There are several interesting points about the generation of this plot that worth paying attention to. First of all, we can see clearly that more features does not mean better classification performance. Secondly, the best individual features are usually not the ones chosen. To illustrate explicitly what this second point is referring to, let's look at the best individual feature performance in classifying the image. Using each individual feature for classifying the pixels in the Brodatz mosaic image, the classification error is tabulated in Table 5.5.

Percentage of Classification Error for Individual Feature (4-13):

4	5	6	7	8	9	10	11	12	13
49.5%	42.6%	36.2%	52.9%	56.5%	54.6%	40.8%	38.0%	54.6%	54.1%

Percentage of Classification Error for Individual Feature (14-23):

14	15	16	17	18	19	20	21	22	23
36.4%	31.9%	32.6%	40.4%	69.8%	38.9%	40.7%	39.7%	38.2%	56.9%

Table 5.5 Individual co-occurrence matrix feature classification performance. Note that the best feature is # 15 and the second best is # 16.

As expected, the first feature selected using the Whitney feature selection method is feature # 15 -- the first order contrast of the (1, 90°) co-occurrence matrix. Naturally, we would think the second best individual feature # 16 -- the second order contrast of the (1, 90°) co-occurrence matrix might end up being chosen next. The classification performance of co-occurrence based classifier with feature # 16 and one of the the other co-occurrence features is shown in Table 5.6.

Percentage of Classification Error for Feature # 15 and Feature (4-13):

4	5	6	7	8	9	10	11	12	13
12.6%	20.1%	15.9%	17.6%	33.1%	12.7%	17.3%	12.7%	19.1%	24.6%

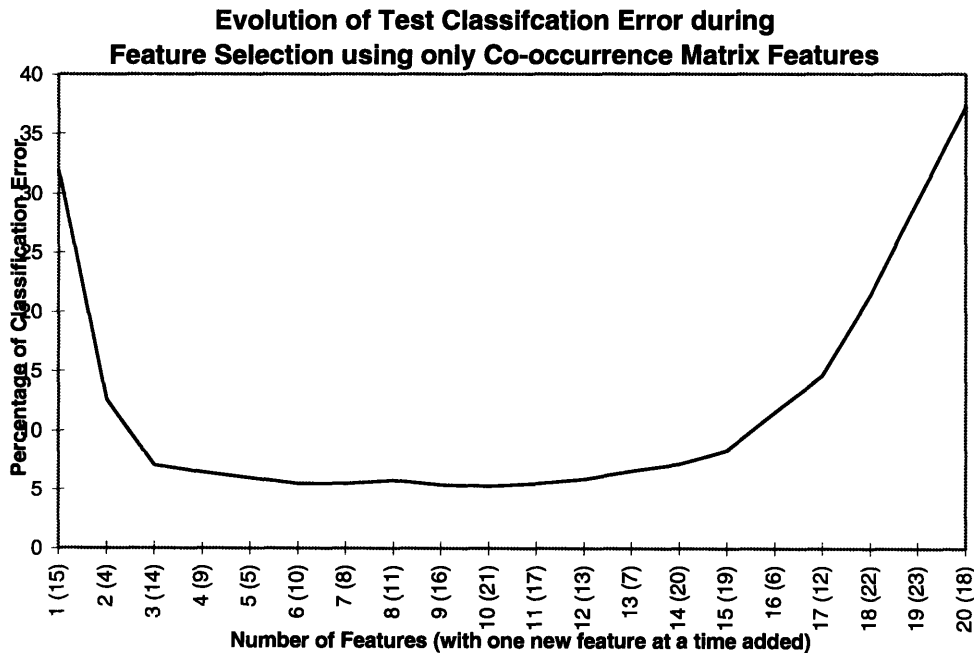


Figure 5.9 This graph shows an example of the classification error as more features are added to be used for classification. Note that these features are only those that belong to the co-occurrence matrix. The numbers in parenthesis are the corresponding feature numbers added. (The test image here is the Brodatz mosaic image in Figure 5.5.)

Percentage of Classification Error for Feature # 15 and Feature (14-23):

14	15	16	17	18	19	20	21	22	23
15.4%	n/a	27.1%	21.8%	37.1%	15.7%	22.7%	20.8%	22.1%	29.5%

Table 5.6 Classification performance of Features # 15 together with one of the other co-occurrence feature. Note the the best performance is by Feature # 15 together with # 4, not with the second best individual feature, # 16.

Surprisingly, the next feature chosen, in combination with # 15, is feature # 4 -- the entropy of the $(1, 0^\circ)$ co-occurrence matrix, which is not even close to being the top individual feature in Table 5.5. On the other hand, the second best individual feature, # 16, when combined with # 15, does not reduce the classification error as much as many other other combination. Therefore, as we have pointed out earlier, the best combination of features could consist of features from an unpredictable set of features which individually might not perform that well. One reason could be that some *essense* of the feature # 16 is already contained in feature # 15. So, the combination does not *span*⁹ a greater space than does the combination of feature # 4 with feature # 15.

5.4.2 Image Classification and Segmentation Results

With all the features presented in this chapter, the first step in classifying an input image is to determine the optimal set of features to use. For comparing performances of the pure maximum likelihood feature-based classifier (denoted by ML classifier) and the enhanced maximum *a posteriori* feature-based classifier (denoted by MAP Classifier), we also apply two other standard classifiers to the images in the following experiments. The two standard classifiers used are the multi-layer perceptron (MLP) classifier trained using the backpropagation algorithm [Rumelhart, et.al., 1986], and the K-nearest neighbors (KNN) classifier [Fukunaga, 1990].

⁹ We borrowed the linear algebra term of “span”, which refers to the combination of vectors [Strang, 1988]. This is purely for illustration purpose. For the two feature vectors case, when two features sets are more “orthogonal”, a larger space can be represented by the features than otherwise

We first consider a phantom image composed of four constant patches with additive Gaussian noise on the left hand side of Figure 5.5, which we will refer to as Gaussian mosaic image from now on. The histogram of the image is plotted in Figure 5.10. The four different regions have plenty of overlapping intensity values. The mean and variances are tabulated in Table 5.7.

<i>Region Location</i>	<i>Top-left</i>	<i>Top-right</i>	<i>Bottom-left</i>	<i>Bottom-right</i>
<i>Mean</i>	120.09	139.80	159.97	179.87
<i>Std. Dev.</i>	40.63	20.42	50.67	30.19

Table 5. 7 The mean and standard deviation of the intensity values in the Gaussian mosaic image 1 of Figure 5.5.
Note the large overlap among the different regions.

These sample mean and standard deviation values are calculated with the *a priori* knowledge of the underlying class assignments. For quantitative evaluations of different segmentation techniques, we must know the true underlying class labels of an input image. Most of the time, any real world images are not fitted for quantitative evaluations since their underlying class labels are not clear. Some pixels in these images could belong to more than one class (consider a satellite image whose pixel size could correspond to several kilometers of real distance on earth). Phantom images such as this Gaussian mosaic image are necessary for these quantitative evaluations.

With the parameters of the feature based classifier discussed in the previous section estimated, we first find the optimal set of features for the Gaussian mosaic image. The *evolution* of the classification error as more and more features are added is shown in Figure 5.11. This *evolution* process proceeds as follows: the first feature is chosen as the best individual feature; the second feature is chosen in combination with first chosen feature to yield the best classification result; the third is chosen in combination with the first two chosen features, and so on.

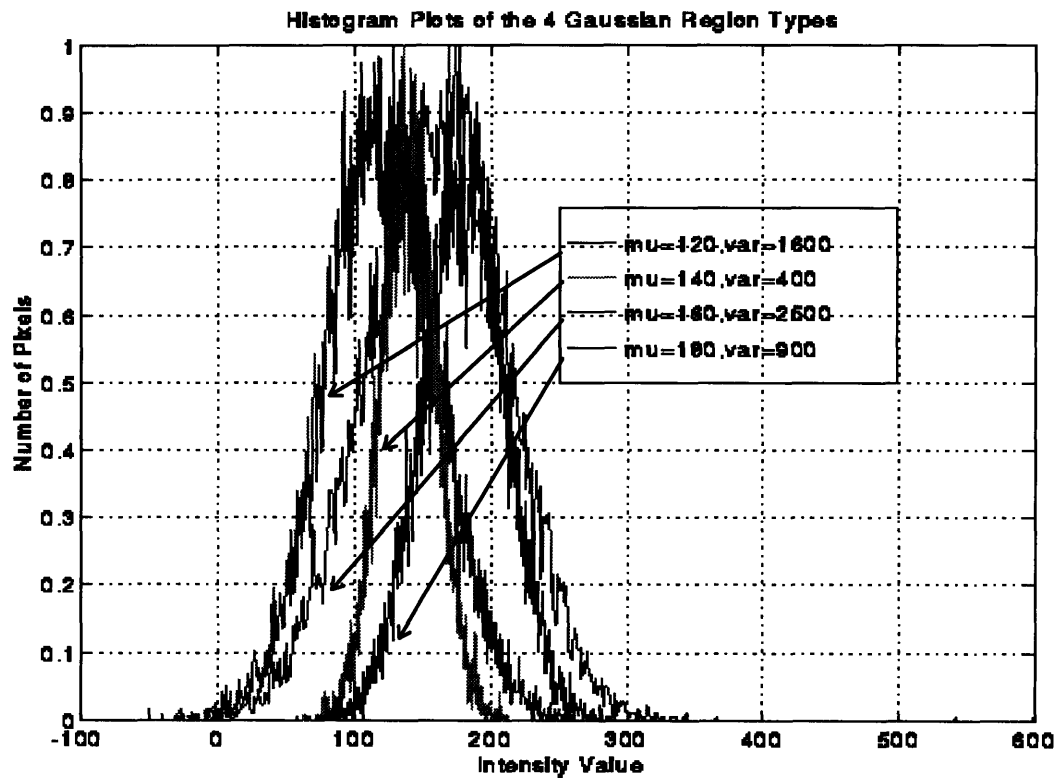


Figure 5.10 Histogram plots of the Gaussian mosaic image of Figure 5.5. Note the severe intensity overlaps among the four different regions.

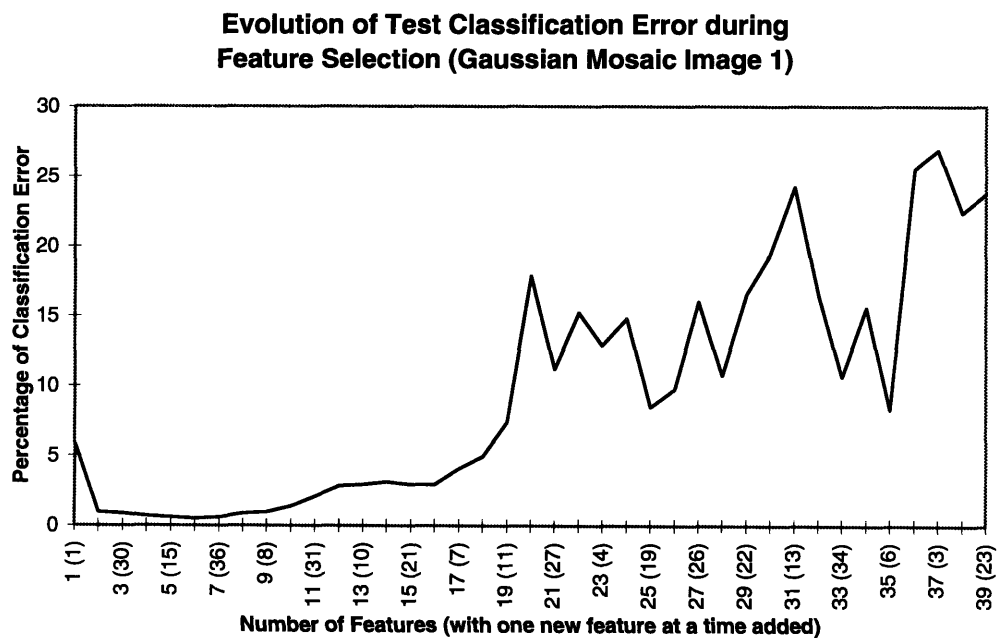


Figure 5.11 This graph shows how the classification error changes as more features are added to be used for classifying the Gaussian mosaic image in Figure 5.5. The numbers in parenthesis are the corresponding feature numbers added. The minimum error occurs with 6 features (1, 20, 30, 37, 15, and 29).

The minimum testing error -- 0.48% of all pixels -- occurs with 6 features chosen, which are : 1, 20, 30, 37, 15, and 29. Clearly from Figure 5.11, we can see that the synergy among different features greatly increases the discriminatory power of the classifier. Another way to view Figure 5.11 is that the information of the original image is “projected” differently into each types of features space since the best set of features consists of features from different feature extraction techniques.

Let’s compare the results of the feature-based segmentation with two commonly used methods we discussed earlier -- multi-layer perceptron network (MLP from now on, with two hidden layers, 60 nodes and 20 nodes each), and the K-Nearest Neighbors (KNN from now on where $K = 24$). Like the feature-based classifier, both of the later classifier receive local intensity regions as inputs for segmenting an image. For the images here, we found that 17×17 for MLP and 5×5 for KNN regions of intensities serve quite well. In training the MLP network, the criterion for stopping is for the mean square error (MSE) of the output nodes to be less than 0.05 [Rumelhart, *et.al.*, 1986; Mui, *et.al.*, 1994]. The actual MSE after complete training is 0.035. Before we look at the results, we have to

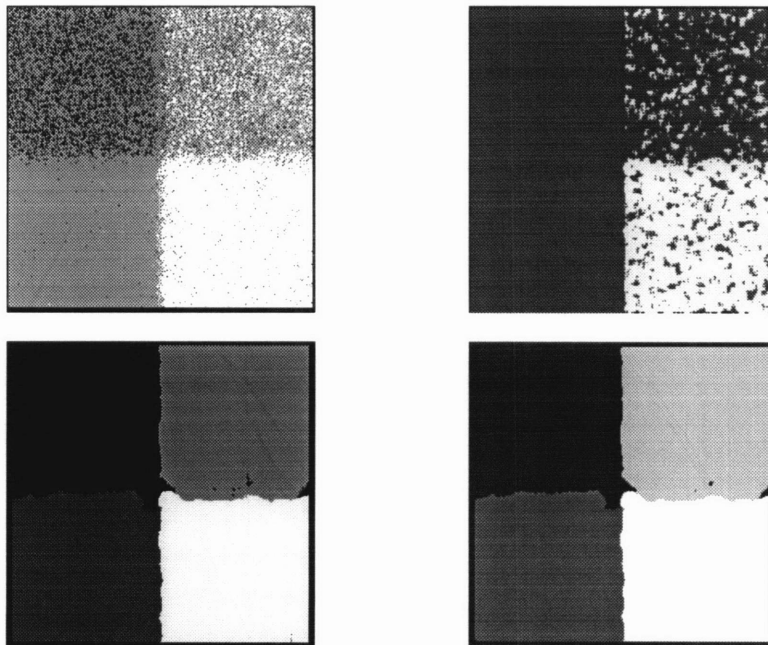


Figure 5.12 Classification of input image Gaussian Mosaic Image (Figure 5.5). Top-left is classified by MLP; top-right is classified by KNN; bottom-left by ML feature-based classifier; bottom-right by MAP feature-based classifier.

**Evolution of Test Classification Error for Selecting
Features for the Brodatz Mosaic Image**

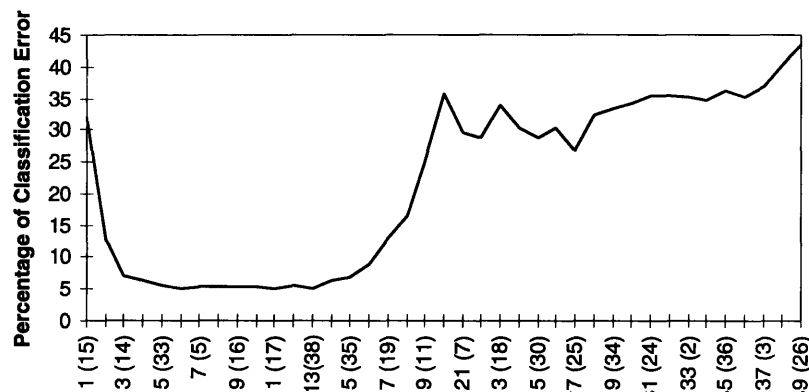


Figure 5.13 This graph shows how the classification error changes as more features are added to be used for classifying the Brodatz mosaic image in Figure 5.5. The numbers in parenthesis are the corresponding feature numbers added. The minimum error occurs with 6 features (15, 4, 14, 9, 33, and 29)

admit that the MLP and KNN are applied blindly without any substantial optimization of their parameters. Nevertheless, we do not expect that the most optimized MLP or KNN using only an intensity map as input would perform much better than what are presented here. For some images to be presented later on in this section, MLP and KNN do perform very well. In the case of a whale temporal bone CT image, MLP performs better than both of the feature-based techniques (ML, MAP). That phenomenon again points out that a single expert, no matter how well “optimized” for an image, cannot expect to perform better than all other methods for other images. Again, a multi-experts approach is needed for robust segmentation on many types of images.

For the Gaussian mosaic image of Figure 5.5, the segmented results are shown in Figure 5.12. Clearly, both MLP and KNN using only the intensity values have difficulties in differentiating the four regions with different intensity means and variances. However, both versions of the feature based classifier seem to do quite well. Quantitatively, the classification errors of all these classifiers are tabulated in Table 5.8.

<i>Classifier</i>	<i>MLP</i>	<i>KNN</i>	<i>ML Feature</i>	<i>MAP Feature</i>
Error	43.90%	72.27%	3.07%	3.02%

Table 5. 8 Classification error of different classifiers on the Gaussian Mosaic Image of Figure 5.5. It is quite clear that the terrible results of MLP and KNN are because of their blind usage; however, we have to admit that if some careful parameter adjustments of MLP (such as number of hidden layers, number of nodes, etc) and of the KNN (the size of K), their performance could improve, but probably not by very much.

We now consider some naturally formed textural images -- the Brodatz textural image in Figure 5.5. Again, we apply the feature selection process described earlier in this chapter to find the optimal set of features to use for segmenting the image. The resulting evolution of the classification error is shown in Figure 5.13. The best feature set consists of 6 features, which are: {15, 4, 14, 9, 33, and 29}. This set of features achieves a minimum testing error of 5.08%. Again the feature set consists of features from different feature extraction techniques -- no single set of features is the best. Nevertheless, for comparing different feature types, we have to concur with the observations made by many researchers that for textural images, the co-occurrence matrix features seem to perform better than most of the other known features [Haralick, et.al., 1979; Ohanian, et.al., 1993]. The segmented images are shown in Figure 5.20. Only the two feature-based results are shown because both the MLP and the KNN fail to do any sensible classification. During training, MLP's MSE fails to fall below 0.50 (which translates to roughly 50% error of the testing set χ_T). KNN with intensity map as input yields random outputs. The "classified" images by these two method are therefore omitted. The quantitative comparison between the ML and MAP results are tabulated in the following

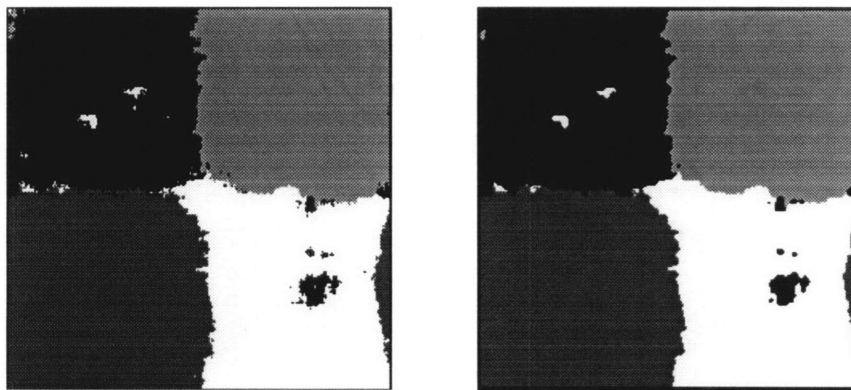


Figure 5.14 Classification of the Brodatz mosaic image of Figure 5.5 by ML and MAP feature-based classifiers using the optimal set of features derived through Figure 5.13

table:

Classifier	MLP	KNN	ML Feature	MAP Feature
Error	(100%)	(100%)	7.21%	6.98%

Table 5.9 Classification error of the feature-based classifiers on the Brodatz mosaic image of Figure 5.5. 100% for MLP and KNN indicates that both of these methods give meaningless results. For the feature-based classifiers, the MAP approach is clearly better than ML approach.

Last but not least, we consider classification of a real world image -- a CT slice image of a whale temporal bone immersed in a beaker of solution, as shown in Figure 5.15. The dark region is air and glass region; the gray area is the solution; the bright area is the temporal bone. Some part of the image is not very clear, such as the left portion of the image -- the bone/solution boundary is not clear. The resulting *soft* edges are always challenges to any classifier. Even human experts might not agree on the exact location of these edges [Bartlett, 1994]. The question naturally arises about the validity of any classification around such edges. We will address this question directly in the discussion section of Chapter 8.

From the feature extraction procedure, we have chosen the optimal set of eight features, which are : 3, 4, 10, 15, 16, 31, 22, and 32, in the order of the selection process.

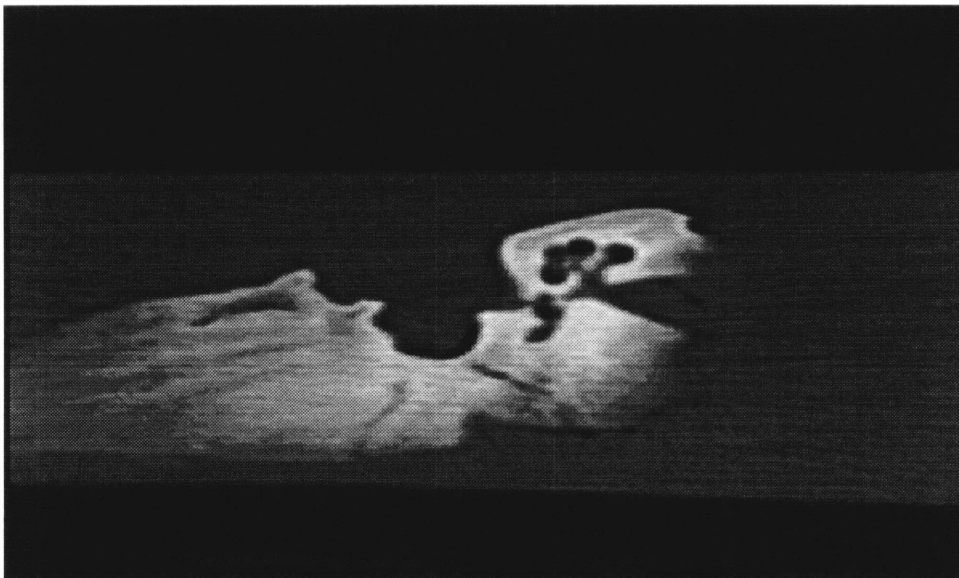


Figure 5.15 A CT temporal bone slice. The dark region is essentially air; the gray region is solution; the bright region belongs to the actual temporal bone slice.

The example learning set is composed of 60 points picked by a human expert to fall within the respective class regions. For each of the different regions, 20 test points are chosen for learning the optimal feature parameters for that region. The same applies to the testing set. The minimal testing classification error is 11.82 %. Since even human experts do not know exactly where the boundaries are, evaluation of the different results can only be done qualitatively.

The classified results are shown in Figure 5.23. The KNN with $K = 24$ seems not able to differentiate the bone and solution at all. Initially, we also encountered difficulties in classifying the temporal bone image with MAP feature-based classifier. We found that the class labels assigned for all pixels belong to the air class. By looking at the likelihood equation (5.19), we realize that for classes that do not have much variation in their parameter space, the determinants of their covariance matrices are small. For the temporal bone image in Figure 5.21, the determinants of the respective classes are shown in the following table:

<i>Class</i>	<i>Air</i>	<i>Solution</i>	<i>Bone</i>
<i> Determinant of Covariance</i>	$3.25e-34$	$3.04e-19$	$1.2e-8$

Table 5.10 Magnitudes of covariance matrix determinants for the three different classes in the whale temporal bone image.

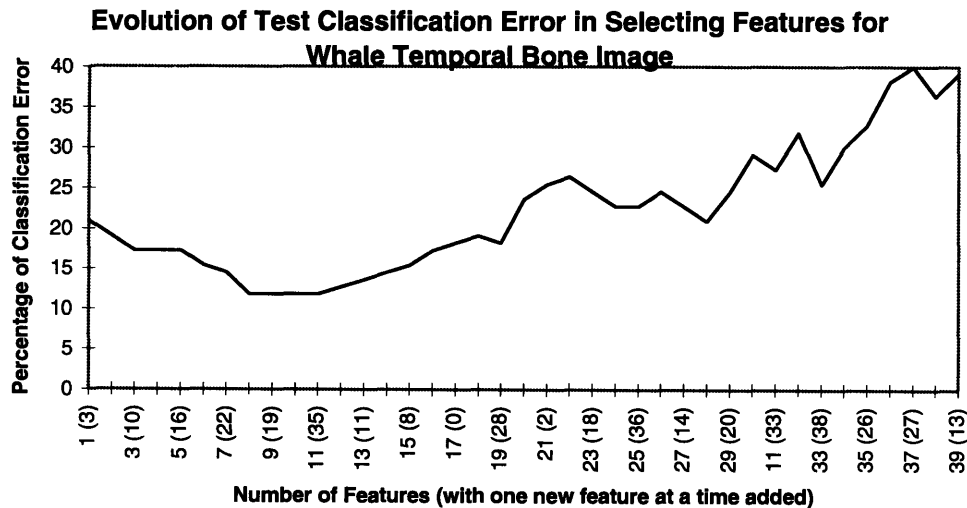


Figure 5.16 Evolution of test classification error during selection of feature for the image in Figure 5.21. The

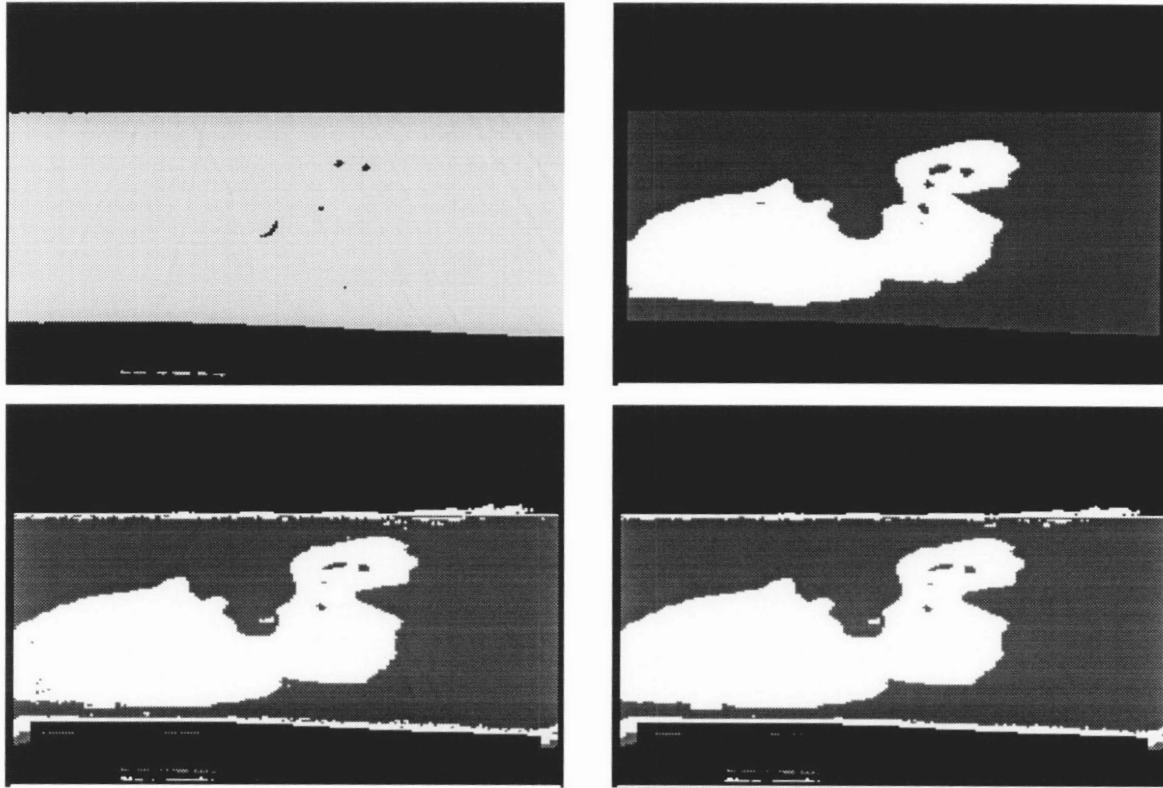


Figure 5.17 Temporate bone classification results. Top-left is the KNN result; top-right is the MLP result; bottom left is the feature-based ML results; bottom-right is the feature-based MAP result. Note that in this case the feature based methods do not perform as well as the MLP technique. In situation like this, the multi-expert approach advocated by this thesis combining both the MLP and the feature-based techniques could take advantage of both in this image as well as the previous images of this chapter.

Since the air region has a determinant at least 25 orders of magnitude smaller than the closest determinant, no wonder that the likelihood for air dominates over those of other two classes.

Subsequently, we found that if we assume all determinants to be the same for all classes, we get much better results than otherwise, as shown in figure 5.17. Even so, the MLP result is better than both the feature-based ML and MAP results. In situation like this, the multi-expert approach advocated by this thesis can take advantage of both the MLP results for this image and the feature-based results for the images presented earlier in this chapter. Results from the MEC approach will be presented in a later chapter.

In this section, we have seen classification and segmentation results for using feature-based classifiers on synthetic as well as real world images. Powerful though it is, feature-based classifiers cannot expect to perform well for all types of images, even when

the “optimal” feature set is derived for a given image. An advocate for feature-based classifier might argue that the reason that a feature-based classifier fails in certain cases is because the set of features for choosing the “optimal” feature set is not optimal. From his viewpoint, a feature-based classifier fails on a given image because we do not truly understand the image under study. This argument can always be put forward to justify spending time and resources to find a better feature set, which might never be found.

Perhaps a better alternative than the never ending cycle of finding a better feature set is the multi-experts approach advocated by this thesis. By incorporating all the existing knowledges we know about a given image, we can be sure that we could perform at least as well as the best of the feature set on any given image (provided the right MEC construction has been made). In fact, ignoring the computational burden, we can incorporate all types of “optimal” feature sets for all kinds of images into a MEC and create a “universal” feature set that can handle all of these kinds of images. Some preliminary results on using MEC to create such a universal can be found in Chapter 8. We will see that segmentation by the MEC approach achieves more accurate segmentation than the feature based methods presented in this chapter.

Chapter 6

OTHER CLASSIFICATION METHODS

This chapter describes four other segmentation methods (or pixel classification) that have been implemented for comparing with the performance of the MEC using various images. Obviously, the algorithms presented here are in no way the only possible set, nor the optimal set. These algorithms only represent a few of the commonly used techniques. All these methods can be incorporated into the MEC structure as experts. Extensions to some these traditional algorithms have been made and are described in the following sections.

6.1 K-Nearest Neighbors

KNN has been used in the pattern recognition field for over thirty years. Various theoretical results exist for this method. The most famous of these is probably the *upper bound* on the classification error of KNN. This upper bound has been proved to be twice the optimal Bayes' error [Fukunaga, 1990]. KNN is also related to several other well known pattern recognition methods such as the Parzen windows and kernel regression [Duda and Hart, 1972]. Due to KNN's intuitive approach and ease of implementation, KNN is often used as a first cut classification method. In this section, a simple extension to

KNN is proposed using MRF as a spatial constraint. This constraint is in the same spirit of regularizing low level vision discussed in [Poggio, *et.al.*, 1986; Marroquin, *et.al.*, 1987].

The KNN classification expert is implemented in two parts. The first part is the density estimation part, while the second is a smoothing operation, which is referred to as spatial estimation part hereafter. The spatial estimation is similar to applying a MRF prior as a spatial constraint on the solution.

6.2.1 KNN Density Estimation

Given the learning set χ_L , the KNN classifier estimates two features for each class - the mean (μ) and the standard deviation (σ), which can be done directly without any iterations since the class labels are known. These two parameters serve as the parameters for the underlying class density.

When a new pattern y is presented for classification, the pattern is compared to the prototypes class features -- μ and σ . A z-norm distance is calculated as followed:

$$d_k = \frac{|y - \mu_k|}{\sigma_k} \quad (6.1)$$

Classification is done simply by choosing the class from which the input pattern is closest to. This classification scheme corresponds to a ML approach.

6.2.2 KNN Spatial Estimation and MRF Prior

The classification performed through the density estimation of the last sub-section is usually very noisy, dotted with pepper noise class assignments. Some type of noise remover that considers the spatial distribution of class assignments is necessary to eliminate this extraneous noise.

This noise removal consideration is a different viewpoint of the same problem that has been addressed through MRF priors throughout this thesis. For example, the MRF prior used for the feature-based classifier is essentially doing noise remover on the

classified image (refer to Chapter 5). Noise in that case refers to the mislabeled pixels. Therefore, the same strategy is used here to impose some spatial constraints on the classification results. The MRF used is second order MRF with the following local characteristics:

$$f(z_i = k | \eta_i) = \frac{e^{\beta \delta_{i,j}(k)}}{\sum_n e^{\beta \delta_{i,j}(n)}} \quad (6.2)$$

where η_i is the local neighborhood of pixel i (refer to Chapter 2). Class assignment, or perhaps more appropriately class reassignment, is performed for pixel i by finding the k with the highest $f(z_i = k | \eta_i)$.

6.2 Multi-layer Perceptron Network

A feedforward neural network with intensity map input can potentially be used as one of the *experts* of the MEC. In Chapter 8, we will see its segmentation performance in comparison with those achieved by other methods as well as by the MEC.

The network is a three layer network, with an input layer of $(2L+1)^2$ nodes, one hidden layer of H nodes, and a final output layer with K nodes¹. Each of the output node represents a class. The values of L , H , and K are problem dependent. The learning algorithm for training the weight vectors of the MLP network is the generalized delta rule [Rumelhart, *et.al.*, 1986]. Usually the right approach for estimating the optimal number of training iterations is through cross validation [Breiman, *et.al.*, 1984], however, such a

¹ Strictly speaking the universal approximation proof mentioned above applied to one-hidden layer network with hidden nodes $> (2 * \# \text{ input nodes})$ [Hornik, Stinchcombe, and White, 1989], the training and testing of which usually take a long time. As mentioned in the background chapter (Chapter 2) even if such a network is constructed, good results are not guaranteed since most learning algorithms cannot train such a network without getting stuck in some local minima or failing to converge. For computational reasons, a network with less than this number is usually used.

technique would create a big computational burden on the *overall response* of the MEC. From experiments done on different types of images, we found that the following stopping criterion seems to work fairly well:

stopping criterion for learning:

Stop learning when either

1) number of iterations = MAX_ITERATIONS

2) output mean square error \leq MIN_MEAN_SQ_ERR

where MAX_ITERATIONS and MIN_MEAN_SQ_ERR are problem dependent. The often used MIN_MEAN_SQ_ERR for one hidden layer logistic output networks with about 80 input nodes (covering a region of size 9x9 pixels) and less than 5 output nodes are usually around 0.2 and the MAX_ITERATIONS is usually never greater than 50. When there are no hidden layer in a network, learning generally would not improve too much after two or three iterations of generalized delta rule training. The implemented softmax output function MLP has much more difficulty than the logistic output function MLP in the learning tasks considered in this thesis². Therefore, only logistic output MLP is used for performance evaluation. We will see in the next section examples concerning this observation.

The inputs to the MLP network can also be the features described in Chapter 5. However, this version of the MLP network is not implemented at the time of writing this thesis. The author expects this MLP network based on the *optimal* feature sets described in Chapter 5 to outperform the intensity map input MLP network of this section.

The MLP structure has already been described in the neural network subsection in the background chapter. Interested readers are also referred to the following references for more detailed discussions [Rumelhart, et.al, 1986; Hertz, et.al., 1990; Hornik, Stinchcombe, and White, 1989].

² Even though the structure of a softmax output MLP is almost identical to that of a logistic output MLP, the failure of the softmax output to learn could be results of the particular implementation. Since MLP is not the main focus of this thesis, no further investigation is conducted in this direction.

6.3 K-Means Based Classifier

An unsupervised method known as the K-means clustering method can be implemented as an expert for the MEC. A K-means based segmenter is evaluated in Chapter 8, along with other methods, for segmenting synthetic and real world images.

Given the number of classes K , this clustering method iteratively estimates the K most prominent centers in the input space. Recall equation (2.18) for finding the clustering centers:

$$\mu_k^{(t+1)} = \frac{1}{N_k^t} \sum_{x_i \in S_k} y_i \quad (6.3)$$

Equation (6.3) expresses the updating rule for locating the K centers of inputs y_i 's. At iteration t , the number of inputs in class k is represented by N_k^t .

Similarly, the variances of the inputs y_i can be iteratively estimated in the following fashion (the term $(N_k^t - 1)$ in the denominator is to ensure that the variance estimator below is a consistent one [Mendenhall, 1990]):

$$\sigma_k^{2(t+1)} = \frac{1}{(N_k^t - 1)} \sum_{x_i \in S_k} (y_i - \mu_k)^2 \quad (6.4)$$

After the means and variances of inputs y_i are estimated for the K classes, class label reassignments have to be done since any unsupervised methods have no *a priori* knowledge on what the true class labels should be.

The K-means classifier classifies input pixels by applying a z-norm distance metric to eliminate the difference in variances between different input features. The z-norm distance metric is defined in (6.1). The class k with the lowest z-norm distance to the input y is the class label for y .

6.3.1 Regularizing the K-means Outputs

Just as the smoothing operation for the outputs of the KNN, a MRF prior is applied to the outputs of the K-means classifier. This MRF prior takes the same form as equation (6.2). In this sense, the K-means algorithm used for this classifier is a modified version the one proposed by MacQueen (1967).

6.4 Probabilistic Co-Occurrence

The detailed algorithm used in the probabilistic co-occurrence classifier is described in Appendix A. The algorithm is based on a novel concept proposed by Lohmann (1995). The co-occurrence matrices associated with an input pattern are classified into one of K classes. The essential idea of Lohmann's idea is to consider classification of the co-occurrence matrix coefficients as an N-way classification problem. Naturally, the probabilistic model is the multinomial density.

For input region with the pixel to be classified in the center, four co-occurrence matrices are computed. These four co-occurrence matrices have displacement vectors $(1, 0^\circ)$, $(1, 45^\circ)$, $(1, 90^\circ)$, and $(1, 135^\circ)$ (refer to Chapter 5 for details on these displacement vectors). For learning the prototype co-occurrence matrices for a class k , the labeled learning set χ_L with L pairs of inputs and outputs is used. The prototype co-occurrence matrices is computed as the average of the co-occurrence matrices of all the learning sets. In other words, for class k , the prototype co-occurrence matrix coefficients for class k ($c_{ij,k}$) are:

$$c_{ij,k} = \frac{1}{L} \sum_{l=1}^L c_{i,j}^l \quad (6.5)$$

When a new input pattern is given, the classification is done by computing the mutual information $I(x,z)$ below for each class:

$$I(x, z = k) = \sum_{i,j} x_{ij} (\log c_{ij,k} - \log c_{ij}) \quad (6.6)$$

This is the same equation as Equation (A.4) in Appendix A. The class with the highest mutual information $I(x, z = k^*)$ is assigned the class label k^* for that new input pattern.

Perhaps the most impressive aspect of this novel approach is that no “optimal” features set is required be chosen first before a good classification results can be obtained. In this respect, this approach is like the non-parametric approaches, but it is definitely a parametric method. But the parametrization operations are done automatically without any intervention from the algorithm designer. Performance evaluation of this probabilistic co-occurrence classifier is done in Chapter 8, along with other classifiers.

Chapter 7

DETAILS OF THE MULTI-EXPERTS CLASSIFIER

The discussion on the Multi-Experts Classifier (MEC) presented in Chapter 3 has left out some of the details of an actual implementation. This chapter documents the portion of implementation that has been left unexplained. No theory is presented in this chapter. Only practical considerations during implementation are described.

This chapter is divided into two sections. Section 7.1 discusses the implementation issues of the gating network. Section 7.2 is concerned with the implementation issue of the stacked generalizer for learning from the outputs of different segmentation experts.

7.1 Gating Network

Chapter 3 has explained the construction and functions of the gating network in the context of the MEC. Several issues have been left untouched. These issues include the type of inputs to the gating network and the variations on the gating network architecture for learning.

7.1.1 Input Feature Sets

The function of the gating network in the MEC architecture is to provide priors for the outputs of each of the experts. As we have shown in Chapter 3, the gating network outputs a number between 0.0 and 1.0 and assign that as the prior for a particular experts. However, what we did not explicitly specify in that chapter is the type of inputs to the gating network.

For the current implementation of the MEC, the gating network receives two sets of feature inputs. There are two ways that these two sets of features are selected:

(i) *exclusive optimal features*

One set is the *optimal* feature set from the feature based classifier. The other set of inputs is also a feature set, but is completely different from the *optimal* feature set. This second set is obtained in exactly the same way as the optimal feature set, namely, through the feature extraction process described in Chapter 5. However, the second feature set is obtained by excluding the features in the optimal feature set from the feature selection process.

(ii) *inclusive optimal features*

Inclusive optimal features are used in Chapter 8 for selecting features of a multiple regions image. For example, there are two types of images in Figure 8.1 -- two Gaussian intensity images, and two Brodatz texture images. One optimal set of features are chosen for distinguishing the two Gaussian intensity images while another optimal set of features are chosen for distinguishing the two Brodatz texture images. Because there is no constraint on the selection during these the selection processes, there could be common features in the two feature sets.

The gating network receives an input vector that is the union of either of these two optimal feature sets.

7.1.2 Network Architecture

The gating network presented in Chapter 3 can be considered as a (no hidden layer) network with output function being either the softmax function or the logistic function. Such network construction has been shown to be not very general [Minsky and Papert, 1969]. Interestingly however, we have observed that the performance of a gating network MEC does not vary significantly with either implementation. Results shown in Chapter 8 will show this point more explicitly.

7.2 Stacked Generalization

A stacked generalizer (SG) learns from the outputs of different segmentation experts. Naturally, the inputs to the stacked generalizer have to come from the outputs of the segmentation experts. But, we still have a choice in terms of what kinds of outputs. The first sub-section below discusses this selection of inputs. For a given input, a SG does not have to give a label under all circumstances. As discussed in Chapter 3, SG can also assign an input to the REJECT class. The last two sub-sections considers under what circumstances should inputs be assigned to the REJECT class.

7.2.1 Inputs to a Stacked Generalizer

Each segmenter outputs a vector with K elements, each of which indicates the confidence (or probability) that the segmenter has toward classifying the input to the corresponding class. Since the SG takes inputs from the outputs of the segmenters, there are two immediately obvious ways to provide inputs for the SG. One possible input format could be the entire probability vector. The other possible input format could be the class label with the highest confidence. There are a variety of variations between these two extremes.

At first, we thought that if the inputs were the entire probability vectors, the performance of the SG would be better than otherwise. However, after several performance evaluation of this first probability vector approach and of the second highest confidence class label approach, we found that the second approach almost always outperforms the first probability vector approach. After some speculation, we believe the reason is likely to be that the SG we have used do not have the generalizability to discriminate the different probability vectors. For this approach, the dimension of the input space is K times higher than that of the class label inputs case.

Because of its better performance, we adopt this second approach of using the class labels as inputs to the SG. This is an area of research for the multi-experts approach that deserves further investigation.

7.2.2 Selection of the *REJECT* Threshold

In Section 3.2.3, we have discussed the assignment of the *REJECT* class to those inputs whose entry into the coincidence matrix is zero. The reasoning is that if there are no learning examples to indicate what class such inputs should belong to, the SG would have no such knowledge to assign labels to inputs belonging to this category.

What is the vector entry in the coincidence matrix has conflicts? By conflict we mean when two elements of the vector entry have the same number of learning examples, or very nearly the same. In this latter case, the SG would not have the built-in knowledge to choose among the two possible outputs. Such inputs suit perfectly into the *REJECT* class. Essentially, we are suggesting the use of a reject threshold. This threshold assigns those classes that the SG is not sure of to the *REJECT* class.

The **REJECT threshold criterion** has been implemented as follows:

if $\frac{c_{\max}}{c_i} \leq \text{THRESHOLD}$, assign input to be *REJECT*, where $i = \{1, 2, \dots, K\}$
 or if $c_{\max} = 0$, assign input to be *REJECT*

where c_i represents a scalar element in the coincidence matrix vector element and c_{max} represents the largest such scalar element (refer to Sub-section 3.2.3).

7.2.3 Post-Processing

After an image has been segmented by a SG MEC, those pixels whose classifications by different experts result in REJECT have to be post-processed before the image can be called completely segmented. This post-processing stage is performed by an iterative process described in this sub-section.

The iterative process to deal with REJECT pixels continues until all REJECT pixels have been converted to have class labels. During each iteration, only those REJECT pixels near at least one labeled pixel are processed. The processed pixels are classified as the most frequently occurring label in its second order neighborhood (see Chapter 2 for the definition of a second order neighborhood).

Chapter 8

EXPERIMENTS AND RESULTS

This chapter describes experiments for testing the Multi-Experts Classifier (MEC) using synthetic as well as real world images. Performance comparisons are made in two respects. First, the MEC's performance is compared to those achieved by commonly used image segmentation methods. Second, different designs of the MEC's are compared against each other. Of special concern is the performance of the gating network approach versus the stacked generalization approach to the MEC design.

The first section in this chapter describes the experimental setup and procedures common to all experiments performed here. The most significant section of this chapter is probably Section 8.2. In this section, *quantitative* evaluations of different segmentation methods are performed. Here, the results of a MEC are quantitatively compared with those achieved by other methods using phantom images with known underlying class labels. These phantom images are synthesized from various types of image formation processes, such as Gaussian Markov random field, simultaneous autoregressive model, Gaussian densities, and natural textures -- from the Brodatz Album. The following Section 8.3 applies the MEC and other methods to real world images described in Chapter 2. Finally, a brief discussion of the results concludes this chapter.

8.1 Experimental Setups and Overview

Consider a generic image S , the experiments performed in this chapter follows a straightforward procedure for segmenting S . This standardized procedure is adopted to ensure that the condition under which every segmentation method is tested is as similar to each other as as possible.

The first step is to choose the number of classes, K . Recently, progress in cluster validations have made it possible to estimate this number reasonably well [Zhang and Modestino, 1990; Li, *et.al.*, 1992]. Nevertheless, for two reasons, such automatic methods are not adopted in this thesis. First, these methods are still far from being reliable. Second, for the type of problems we consider here, selecting the number of classes is not a major problem and can easily be estimated by an expert user.

After the number of classes is selected, a user has to choose the learning set χ_L and the testing set χ_T . The learning set χ_L is used for parameter estimations and the testing set is for performance comparisons. Instead of selecting two data sets, we could have chosen to use bootstrap methods to perform parameter estimation as well as testing [Efron and Tibshirani, 1993]. However, two independently chosen sets are always better than any equivalent sized data sets obtained by bootstrap methods. Since an image commonly has at least tens of thousands of pixels, choosing two such sets should not be a problem.

The functional distinction of χ_L and χ_T can best be illustrated by an example. In the case of the feature based classification expert (Chapter 5), the optimal feature set is selected through an iterative process during which both the χ_L and the χ_T are used. During the selection of the first *optimal* feature, the parameters of the classifier are learned using χ_L . (The parameters here refer to the mean vectors and the co-variance matrices of the different classes.) Each feature is then *benchmarked* against each other using the χ_T . The feature that performs best on χ_T is then selected as the first *optimal* feature. The second optimal feature is chosen similarly in combination with the first optimal feature.

The process of selecting the learning set and the testing set is performed only once by a user (for example, by the use of a mouse pointer). The chosen pairs are recorded to a file which is read automatically every time that image is loaded. A simple parser is written to parse the file for every segmentation test thereafter. An example learning set file is shown in Appendix D. In choosing χ_L and χ_T , the user is assumed to know the approximate location of different classes. (Usually, it is the boundaries that most expert users fail to identify accurately. However, identifying pixels in other areas of a reasonably sized region should not be very difficult [Bartlett, *et.al.*, 1994].)

With the parameters estimated from χ_L , different segmentation algorithms are then applied to the images. The output of each algorithm is a probability vector of how likely each input belongs to the different classes. These output vectors are the inputs to the stacked generalization (SG) module and the gating network (GN) module of the MEC. Using Wolpert's language, the individual segmenters are zeroth level learner while the SG and GN are first level learner [Wolpert, 1992].

For comparing the performance of different algorithms, a scoring system is devised. For every test pair in χ_T that an algorithm correctly identifies, that algorithm gets one point. If there are T testing pairs of input vectors and output labels, the maximum score an algorithm can get is T . Obviously, the higher the score, the better is an algorithm and the lower the score, the worse is an algorithm for a given image. This scoring system does not penalize or award scores on the difficulty of classification different types of pixels. For example, classifying a boundary pixel is usually much more difficult than classifying a pixel in the center of a region. This difficulty is caused by the spatial dependency assumption made by most segmenters (and human as well). In the case of a MRF model, the local field could be greatly changed near a boundary as the local characteristics over a finite size region changes from one region to another. Nevertheless, the user has the control to choose whether he or she would like to put greater weight on the boundary classification ability of a segmenter by simply choosing more training and testing pairs near region boundaries.

After χ_L is first used to estimate the individual expert's parameters. the testing set is used for *learning* the parameters of the SG module and the GN module. From the output vectors of the experts, parameter estimations of the SG and GN are performed using χ_T . The parameters of interest for the SG are the coefficients in the coincidence matrices or the weights in a SG network. The parameters for the gating network are the network weights. The algorithms for estimating these parameters are the gradient descent algorithms derived in Chapter 3.

8.2 Experimental Proof of Concepts: Quantitative Results

This section tests the basic ideas behind the Multi-Experts Classifier (MEC). The underpinning concept of the MEC is intelligent combination of expert knowledges during image segmentation. To quantitatively show how MEC achieves this goal, we have to consider an image that is generated by region processes with known class labels, and check whether MEC could correctly estimate these class labels at the right locations. In the next section, we will show segmentation results on real world images. This section is important because *quantitative* evaluations are performed on the various classifiers and the MEC. For most real world images, evaluation of these results can only be done qualitatively due to lack of known class labels for those images.

8.2.1 Problem Definition

One example image that is used in this section to quantitatively test the basic concepts behind the MEC is shown in Figure 8.1. This image is formed by two distinct types of image formation processes (texture regions and Gaussian density regions). Since we know *a priori* the class assignments of all the pixels in the image, we can quantitatively evaluate different techniques' performance. Also since the image is a mosaic of two

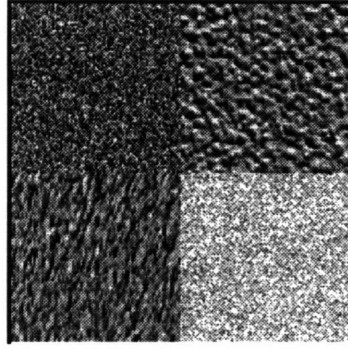


Figure 8.1 An example phantom image for proof of the MEC underlying concepts. There are two types of images there. Upper left hand quadrant and lower right quadrant are two patches with Gaussian distributed intensities (mean 120, variance 1600, and mean 180, variance 900). Upper right hand quadrant and lower left quadrant are two fairly stationary natural textures from the Brodatz Album (D9 and D68).

different types of images, we can carefully examine the claims we have made on the ability of a MEC to segment such an image.

Consider Figure 8.1, diagonally across from each other, the upper left hand quadrant and the lower right hand quadrant are generated by a Gaussian process with means and variances indicated in the figure caption. The upper right hand quadrant and the lower left hand quadrant are two natural textures from the Brodatz Album [Brodatz, 1967]. These two textures have been chosen because their textural properties seem to be fairly stationary throughout the quadrants.

Clearly there are four classes of images in Figure 8.1. For each class, 20 random points are chosen for that class' learning set and another 20 random points are chosen for the testing set. As we have found out after some initial experiments, 80 testing pairs are not enough to differentiate the different methods' performance. So, the nearest neighbors and the nearest diagonal neighbors of the testing sets are also included in the test set, assuming that they also belong to the center pixel's class. This is not a bad assumption since users generally cannot distinguish any two pixels at 1 pixel length anyway. Even if users can identify 1 pixel length, it is not clear whether the imaging modality used has that clear of a resolution to warrant these special user' unusual ability. When the learning and testing sets are chosen for experiments in this chapter, pixels very close to the boundaries are therefore avoided. As a result of this expansion of the testing set, the maximum score

for any particular segmentation is 720. Classification results are given in percentages of number of pixels correctly classified over 720 maximum pixels correct.

8.2.2 Scope of the Problem

To limit the scope of the problem to an easily interpretable one, we first consider only two experts, both of which are feature-based experts with their own *optimal* feature sets. One of these experts have an *optimal* feature set learned using only the Gaussian patches and the other one have an optimal feature set learned using only the Brodatz texture patches. These optimal feature are chosen according to the procedure described in Chapter 5. These two feature sets are:

Gaussian patches set: {1, 20, 30, 37, 15, 16}

Brodatz patches set: {15, 4, 14, 9, 33, 29}

where the numbers are the ones corresponding to the features described in Chapter 5. These features include first order statistics, co-occurrence matrix features, Simultaneous Auto-regressive model and Gaussian Markov random field features.

To provide standards for comparison, we take the two *optimal* feature sets separately and perform segmentations according to the method prescribed in Chapter 5 for the feature-based classification experts. The results are shown in Figure 8.2. Different regions after segmentation are given in different greylevels. The left hand image corresponds to that of using the Gaussian patches feature sets while the right hand image corresponds to that of using the Brodatz patches feature sets only. The classification error

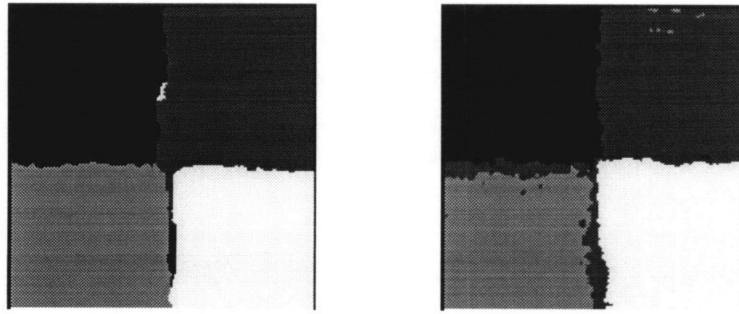


Figure 8.2 Segmentation results of Figure 8.1 by two feature based classifiers. The left hand side has a feature set that is optimal for the Brodatz patches of Figure 8.1 and the right hand side has a feature set that is optimal for the Gaussian patches of Figure 8.1. These segmentations are performed using the MAP method (Chapter 5)..

of the segmentations is tabulated in Table 8.1 below:

<i>Feature Set</i>	<i>Gaussian</i>	<i>Brodatz</i>	<i>Combined Set</i>
<i>Classification Error</i>	5.89%	5.19%	5.19%

Table 8.1 Segmentation error of the two feature-based classifiers, whose segmentation results are shown in Figure 8.2.

What happens if we use all the features in the two optimal feature sets? Table 8.1 also shows the combined feature set results. No improvement is made.

As shown in Figure 8.2, most of the error occurs near the boundaries, which makes sense since most feature extraction procedures requires the use of a local region for feature extraction. This local region, when comes near an edge, becomes highly non-uniform, and violates the stationarity assumption that most feature extraction routines make. Nevertheless, the feature based classifiers perform pretty well overall.

8.2.3 Segmentation by Single Model Methods

We consider the performance of single model methods on this phantom image in this sub-section. Figure 8.1 is an image type that most single model methods would fail to segment because the multiple image formation processes for the four different patches cannot be easily modeled by any single image formation processes. Figure 8.3 shows the results of K-nearest neighbors (KNN), multilayer perceptron (MLP), K-means, Gaussian mixtures with EM estimated parameters and a MRF spatial prior (MRF), and the probabilistic co-occurrence approach.

For generating the segmentation results in Figure 8.3, the K-nearest neighbors' K parameter is determined to be 24¹. The MLP has an input layer of 81 nodes, taking in an image area of 9x9 pixels, one hidden layer of 40 nodes, and an output layer of 4 nodes which correspond to the four different classes. Training for the MLP took 50 epoches,

¹ The K values for KNN, as well as the parameters for other single model methods are not estimated carefully. Their values are adjusted to yield the a good "eyeballing" segmentation results. To estimate these parameters in a more prudent fashion would most likely require a lot more computational resources.

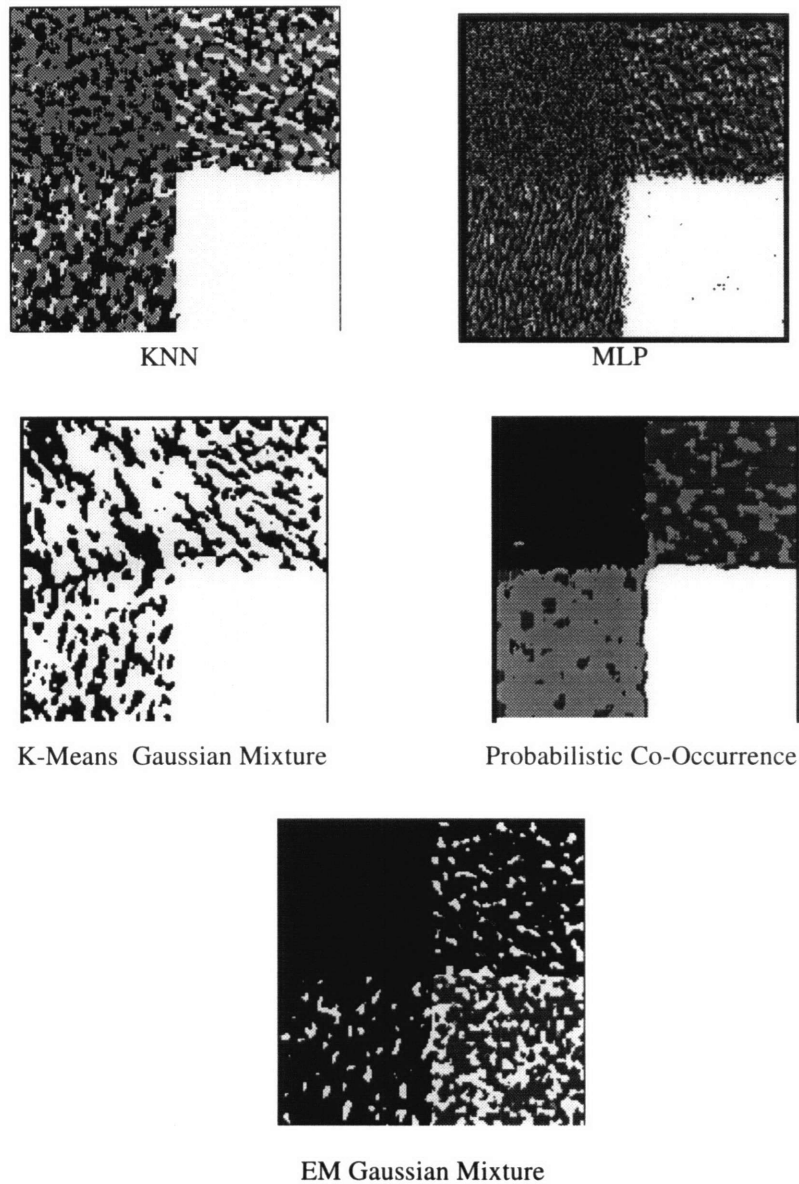


Figure 8.3 Segmentation results by single model methods on Figure 8.1. Clearly 8.7 is composed of several different types of images and most single model methods fail to perform well on these images (refer to Table 8.2).

with a final mean square error (MSE) of 0.28. No further training is performed since no improvement in the output MSE is observed. The output function used for the MLP is the logistic function. Softmax output function MLP does not converge during learning for this image and so no result is shown for this case.

The probabilistic co-occurrence classifier uses co-occurrence matrices of size 8×8 , corresponding to 8 intensity level bins. The best region size used for calculating the co-occurrence matrices is estimated to be 13×13 , with the pixel of interest in the center. Both the EM trained Gaussian Mixture (EMGM) and the K-means trained Gaussian Mixture learned their respective parameters in 10 epoches. The MRF model used for the EMGM is a second order Markov model, likewise for the output smoothing MRF of K-means. Because these two methods assume Gaussian distributed images for their inputs, their results are expected not to be very good on the Brodatz texture patches.

The segmentation error of these algorithms are tabulated in Table 8.2.

<i>Algorithm Name</i>	<i>Classification Error</i>
<i>KNN</i>	55.7%
<i>MLP</i>	48.0%
<i>K-means</i>	65.2%
<i>Probabilistic Co-occurrence</i>	10.1%
<i>Gaussian Mixture (EM)</i>	65.5%

Table 8.2 Classification error of different experts on the image in Figure 8.1

The poor performance of some of these methods shows that blind application of an image classifier, no matter how sophisticated, is a mistake. When an image is composed of multiple types of regions, such as the one shown in Figure 8.1, simultaneously modeling all these types is very difficult with a single expert. For example, we certainly do not expect the EM trained Gaussian Mixture (or K-means classifier) that uses the pixel intensity and intensity variance to perform well on the texture type images. Once again, we see the need for a multi-experts approach when confronted with an image such as Figure 8.1 which contains several different types of images.

8.2.4 Gating Network Combination Methods and MEC Results

Finally, this sub-section considers the results of segmenting Figure 8.1 using the multi-experts approach. The gating network approach to combining the expert results is attempted first. Coincidence methods are discussed in the next sub-section.

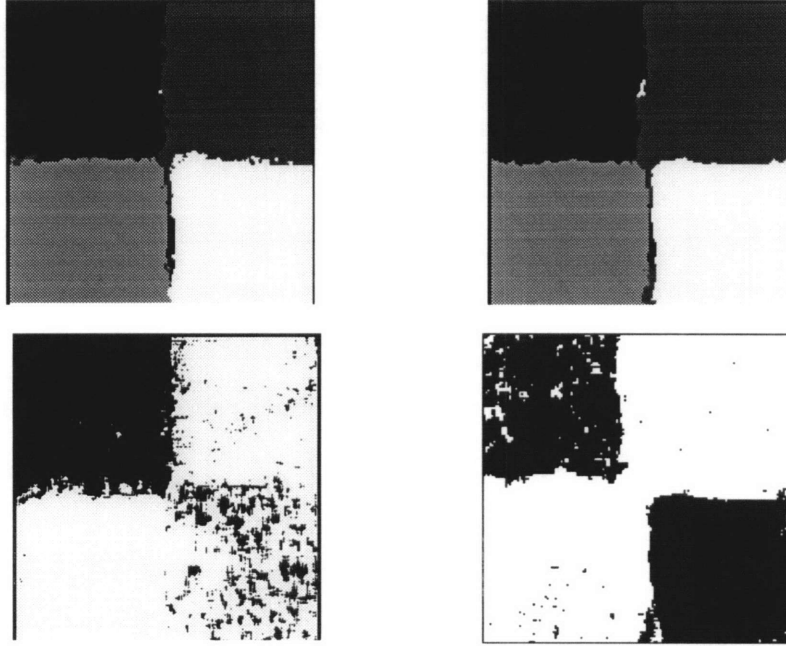


Figure 8.4 The upper two are classification by MP and MAP using the logistic output gating network, respectively. The lower 2 figures are the “expertise map” where black is feature expert 1 and white is feature expert 2. Notice the nice partition of the expertise map according to the type of image there are in the test image, especially for the MAP case on the right. i.e. the MEC successfully partition the input space into expertise

The results are shown in Figure 8.4 for the logistic output function. The upper left hand figure is selected by maximizing the prior beliefs of the gating network. Therefore, we call this network the *maximizing prior* (MP) gating network. Recall the definition of $g_n(\mathbf{y}_i)$ in equation (3.5) and (3.7), the MP class estimates are performed by applying:

$$z_i^{*(MP)} = \arg \max_{n,j} g_n(\mathbf{y}_i, z_j) \quad (8.1)$$

Essentially (8.1) describes a maximization process of jointly selecting the largest gating network output, g_n and the class label z_j . Another viewpoint of this procedure is: select the largest class of the expert corresponding to the largest gating network output z_i^* .

The upper right hand side of Figure 8.4 shows the MAP segmentation result of a MEC with a logistic output function gating network. (This network has been trained for two epoches on the testing set χ_T to achieve a output mean square error (MSE) of 0.096. Further training did not improve the MSE significantly -- less than 0.001 changes.) This classification result corresponds to a maximum *a posteriori* selection of the class label for

an input. To see this MAP connection, consider the maximization process below for deriving the segmentation result:

$$\begin{aligned} z_i^{*(MAP)} &= \arg \max_j g_n(\mathbf{y}_i) f(\mathbf{y}_i | z_j, \Phi) \\ &= \arg \max_j p(z_j | \Phi) f(\mathbf{y}_i | z_j, \Phi) \end{aligned} \quad (8.2)$$

where the substitution of $g_n(\mathbf{y}_i)$ by $p(z_j | \Phi)$ is justified by equation (3.5) and (3.7). The last equation in (8.2) is equivalent to a MAP estimate of the output states with the gating network output as the prior.

The lower part of Figure 8.4 are the “expertise map” chosen by the gating network according to the type of expertise the MEC has. For our case here, the two experts are the two feature based classification experts whose feature sets are described in Sub-section 8.2.2. Especially for the MAP case, notice that the gating network has divided the input space into two divisions almost along the true boundaries. In the *expertise map*, black represents feature based classifier 1 and white represents feature based classifier 2.

The results in Figure 8.4 are better than results obtained by any of the methods discussed so far. Quantitatively, the results of applying (8.1) and (8.2) are tabulated below:

<i>Logistic Gating Network</i>	<i>MP Classification</i>	<i>MAP Classification</i>
<i>Classification Error</i>	4.71%	4.08%

Table 8.3 Logistic Output Function Gating Network classification results.

These results quantitatively shows that not only are the gating network MEC results look better than those achieved by any of the single expert, they are actually better in terms of pixel by pixel count for the segmented image.

Recall in Chapter 3 that we have made a conjecture that a series of Bernoulli processes are better probabilistic models for the gating network than multinomial densities. Recall that the function g for the former case is the logistic function, while g is the softmax function for the latter multinomial case. The results in Figure 8.5 and Table 8.4 using the

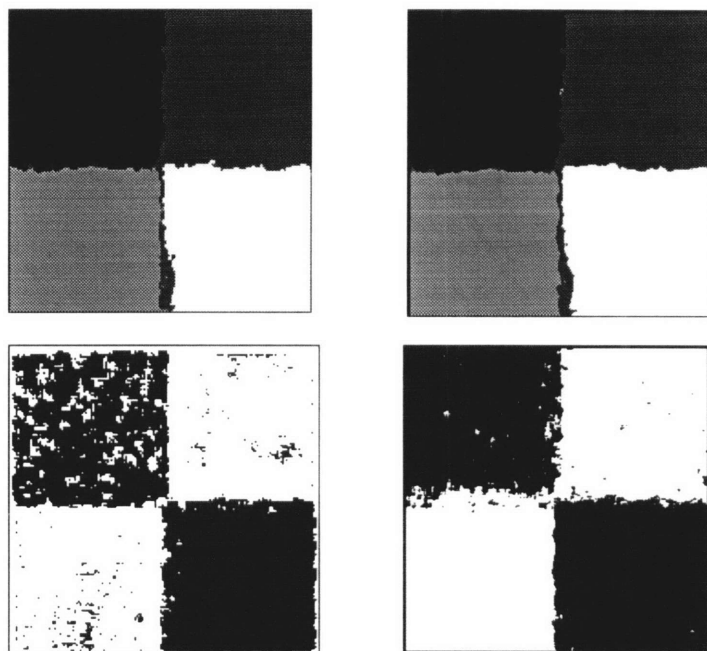


Figure 8.5 The upper two are classification by MP and MAP using the softmax output gating network, respectively. The lower 2 figures are the “expertise map” where black is feature expert 1 and white is feature expert 2. Notice the nice partition of the expertise map according to the type of image there are in the test image, especially for the MAP case on the right. i.e. the MEC successfully partition the input space into expertise regions!

softmax output gating networks prove that our conjecture is inaccurate for this phantom image.

<i>Softmax Gating Network</i>	<i>MP Classification</i>	<i>MAP Classification</i>
<i>Classification Error</i>	3.65%	3.46%

Table 8.4 Softmax Output Function Gating Network classification results.

For the softmax output case, the same set of experiments as the logistic output case is done. (The softmax gating network has been trained for two epoches on the test set χ_T and achieves an output mean square error of 0.707, with less than 0.001 decrease after this second epoch.) The results of which are shown in Figure 8.5 and numerically tabulated in Table 8.4. As mentioned earlier, the results from the softmax output gating network are better than those achieved by the logistic function based gating network. This result not only specific to this image *per se* and will be shown to be true for other images later on in this chapter. Note that the improvement in accuracy of segmentation over the feature-based classification experts is impressive, dropping from 5.19% error of the best

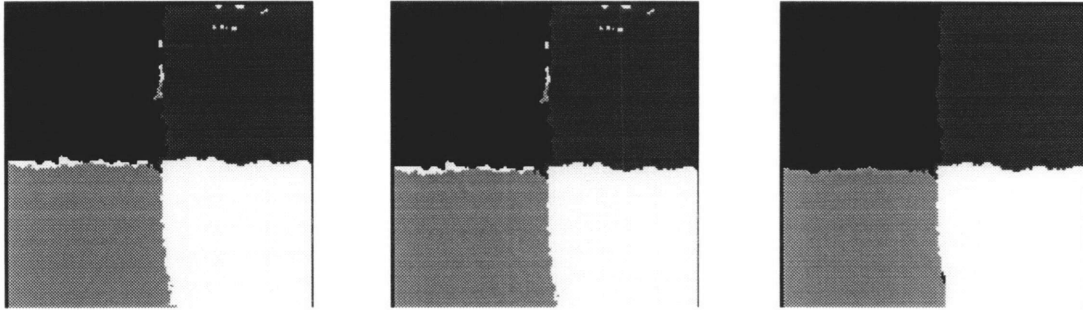


Figure 8.6 Segmentation result of a MEC using the stacked generalizer. The left hand image is segmentation result of a logistic output stacked generalizer; the middle image is the segmentation result of a softmax output stacked generalizer; the right hadn image is the segmentation result of a coincidence matrix based stacked generalizer.

single expert to 3.46% error in the case of MAP classsication using softmax output gating network. This drop (33%) is significant.

8.2.5 MEC Results Using Stacked Generalizers

Three types of stacked generalizers have been tested -- the logistic output network stacked generalizer, the softmax output network generalizer, and the coincidence matrix based stacked generalizer. For both the logistic output network and the softmax output network, the network architecture shown in Figure 2.9 is used. We have attempted to implement a network approach to SG without any hidden; however, the resultant network does not converge for over 100 epoches training using the training set. Therefore, we adopt the Figure 2.9 architecture with one hidden layer, which does converge. The inputs are the output vectors of the two feature-based experts, with a total of 11 input nodes because there are a total of 11 distinct features (refer to Section 8.2.2). The hidden layer for each of the two networks is composed of 40 hidden nodes. Finally, the output layer consists of 4 nodes, corresponding to the four classes. Training the softmax output network took 10 epoches to with a final output mean square error (MSE) of 0.77. Further training does not improve the output MSE significantly (less than 0.01). For the logistic output network, only two epoches were necessary to achieve a final output mean square of 0.12. Further training does not improve the output MSE significantly but causes osccillations in the training output MSE (at least up to the 100 epoches).

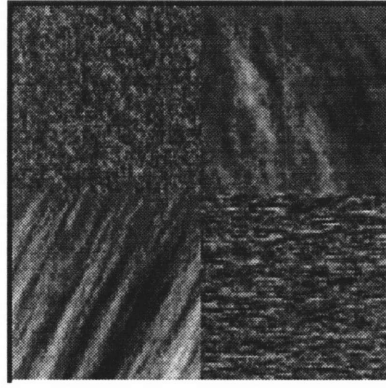


Figure 8.7 Phantom image composed of four different regions. The upper left hand image is simulated by a SAR process with coefficients $\{0.2, 0.2, 0.2, 0.2\}$. Likewise, the lower right hand image is also simulated by a SAR process with coefficients $\{0.8, 0.0, 0.0, 0.0\}$. These coefficients belong to equation 5.15 (refer to Figure 5.3). The upper right hand image is simulated by a Gaussian MRF process with coefficients $\{0.25, 0.25, 0.25, 0.25\}$. Likewise, the lower left hand image is also simulated by a Gaussian MRF process with coefficients $\{0.0, 0.5, 0.5, 0.0\}$. These coefficients belong to equation 5.11 (refer to Figure 5.1).

For the coincidence matrix version of the SG, we mentioned in the last chapter that the segmentation results are nearly always robust against variations of the REJECT threshold parameters. For the phantom image segmentation here, as this threshold is varied from the minimum of 1.0 to the maximum of infinity, the segmentation results stay the same -- classification error remains at 2.94%.

Table 8.5 below shows the segmentation performances of the three possible SG approaches to MEC introduced in this thesis:

<i>Stacked Generalizer</i>	<i>Logistic SG</i>	<i>Softmax SG</i>	<i>Coincidence Matrix SG</i>
Classification Error	3.35%	3.35%	2.94%

Table 8.5 Classification results of a MEC with three different stacked generalizer (SG) -- logistic output single layer network, softmax output single layer network, and coincidence matrix stacked generalizer.

Shown in the segmented image of Figure 8.6 and in the numbers in Table 8.5 are evidence that the SG approach to MEC is feasible and performs better than other methods attempted in this section. The classification error has been dropped to 2.94%, which is a 44% drop from the result of the best of the experts -- one of the feature-based classifiers.

8.2.6 Segmenting More Phantom Images

The results in the last sub-sections show that the MEC approach is able to segment the phantom image in Figure 8.1 better than any other method. Is this result generalizable to other images? To answer this question, we have to apply the MEC to other phantom images and observe the results. The segmentations performed on phantom images in this section follow the same procedure as the segmentation of the phantom image in Figure 8.1 described in Sub-sections 8.2.1-8.2.5.

(i) *phantom image with synthetic GMRF and SAR patches*

Figure 8.7 shows a phantom image with four patches of different classes. The upper left hand patch and the lower right hand patch are simulated by applying equation 5.15 for the simultaneous auto-regressive (SAR) model. Their corresponding coefficients are $\{0.2, 0.2, 0.2, 0.2\}$, and $\{0.8, 0.0, 0.0, 0.0\}$. The upper right hand patch and the lower left hand patch are simulated by applying equation 5.11 for the Gaussian Markov random field (GMRF) model. Their corresponding coefficients are $\{0.25, 0.25, 0.25, 0.25\}$ and $\{0.0, 0.5, 0.5, 0.0\}$ respectively.

The two optimal feature sets for the two SAR and Gaussian MRF are:

optimal feature sets for:

$$\begin{aligned} \text{SAR:} & \quad \{28, 11, 5, 1, 10\} \\ \text{Gaussian MRF:} & \quad \{10, 20, 14, 17, 16, 29\} \end{aligned} \quad (8.3)$$

These two feature sets are the inputs to the two feature-based experts as well as to the MEC gating network. These features numbers correspond to those given in Chapter 5.

Segmentation of Figure 8.7 by various single model methods are shown in Figure 8.8. Note that the phantom image in Figure 8.7 is one of the cases we claim in Chapter 1 that most single model methods would fail on because of the multiple image formation processes for the four different patches. We certainly do not expect methods such as the K-means classifier that rely on the mean and variance of image to perform well at all.

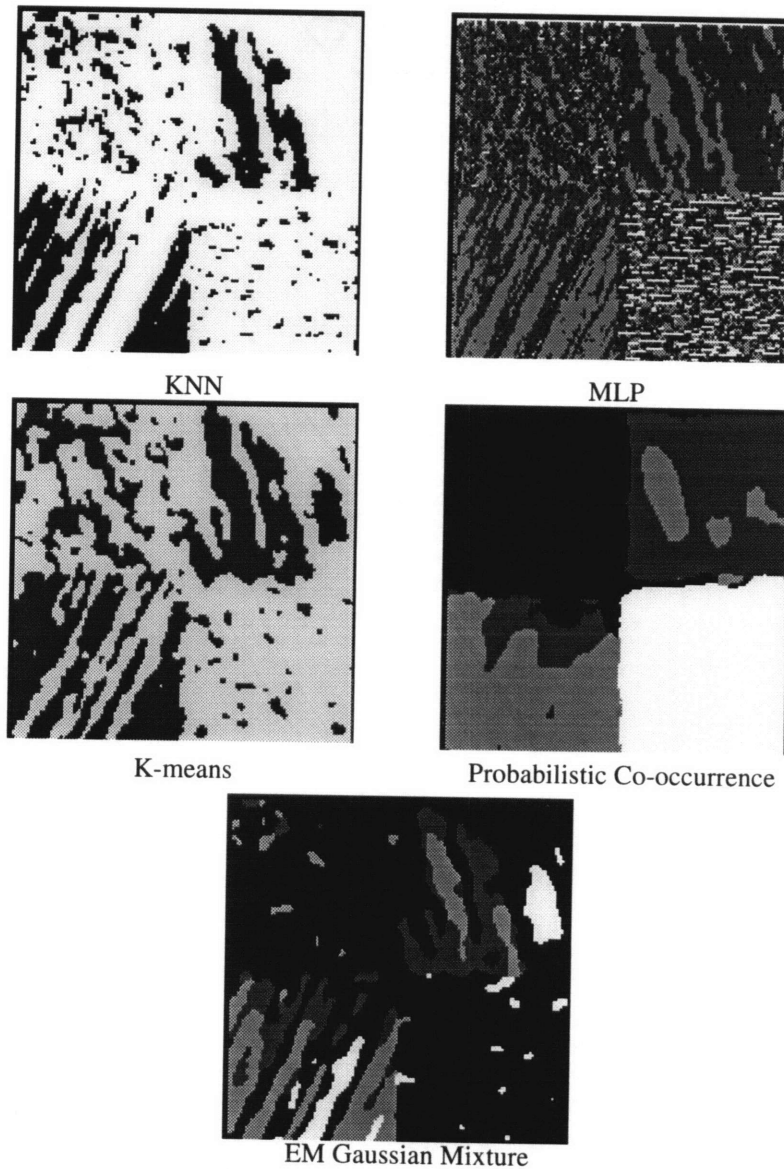


Figure 8. 8 Segmentation results by single model methods on Figure 8.7. Clearly 8.7 is composed of several different types of images and most single model methods fail on these images.

Of the single model methods, as shown in Table 8.6, the probabilistic co-occurrence method has obtained the least classification error. The parameters for these methods are: KNN has a K of 24, MLP has an input map of 9x9, a hidden layer of 40 nodes, and has been trained for 50 epoches to obtain an output mean square error of 0.29. The probabilistic co-occurrence has an input size of 25x25 to yield the 4 co-occurrence matrices described in Appendix A. Both K-means and EM Mixture of Gaussian have been iterated for 10 epoches each.

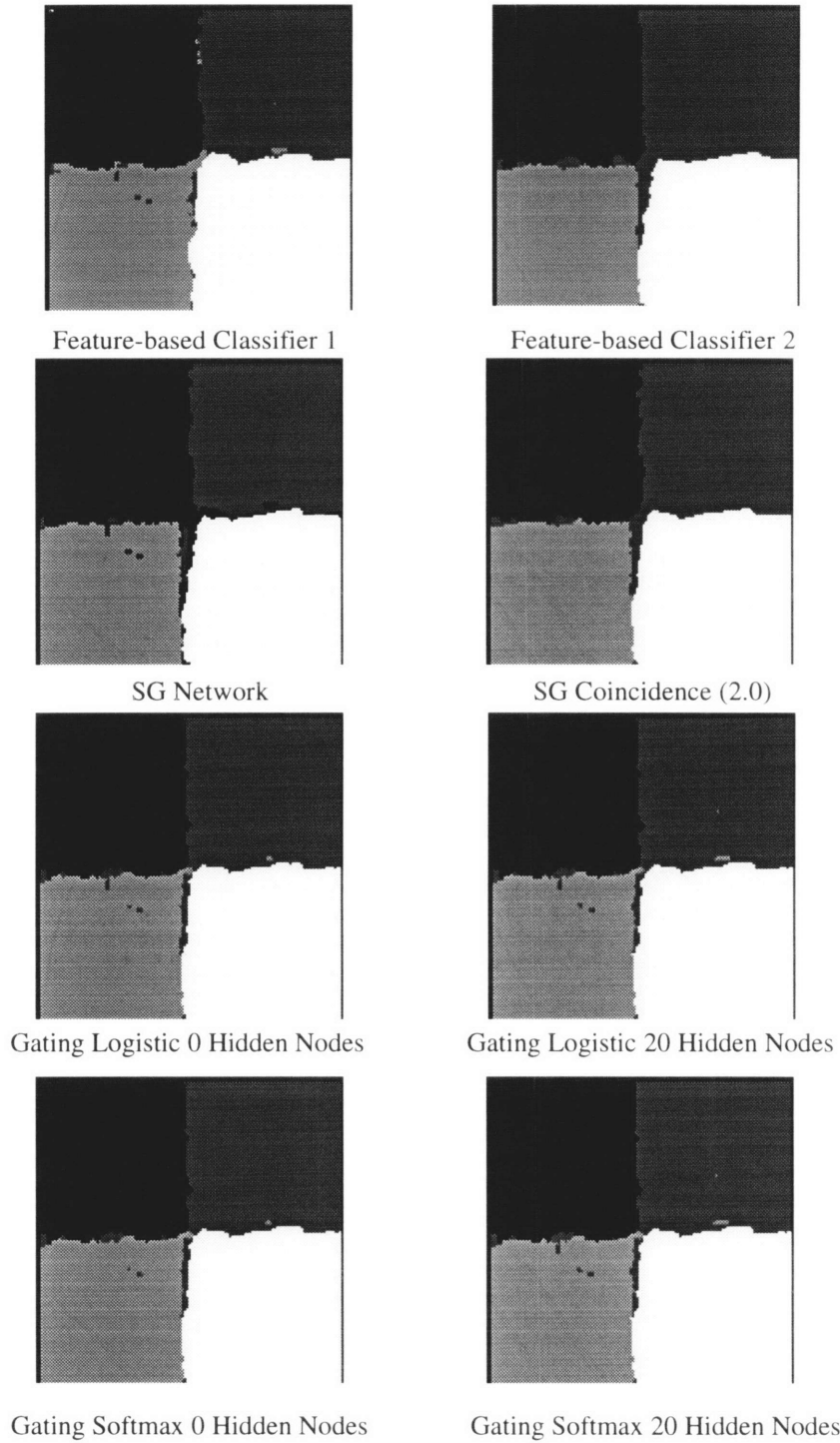


Figure 8.9 Segmentation results of applying different MEC related algorithms to Figure 8.7. The top two are results of applying the two feature-based experts with their respective optimal feature sets in (8.3). SG stands for stacked generalization approach, and Gating stands for the gating network approach. SG Network refers to the use of a single hidden layer with 40 hidden nodes) MLP trained for 5 epoches. The parenthesized numbers next to the SG Coincidence represents the REJECT threshold of 2.0. Finally, the Gating network either has no hidden layer (0 hidden nodes) or has 1 hidden layer (with 20 hidden nodes).

<i>Algorithm Name</i>	<i>Classification Error</i>
<i>KNN</i>	<i>77.01%</i>
<i>MLP</i>	<i>60.56%</i>
<i>K-means</i>	<i>67.12%</i>
<i>Probabilistic Co-occurrence</i>	<i>15.26%</i>
<i>Gaussian Mixture (EM)</i>	<i>61.05%</i>

Table 8.6 Segmentation results on the phantom image in Figure 8.7 by various single model methods.

In Chapter 1, we have mentioned that the difficulty for these single model methods in segmenting Figure 8.7 lies in the single-model nature of their method while the image is generated by multiple models. The Multi-Experts Classifier (MEC) is designed to deal with these types of difficulties. Figure 8.9 shows segmentation results of several variations of the gating network and stacked generalization MEC. We have tried different variations of the MEC approach here to see how the segmentation results would be affected by changes in the parameters of the MEC. The classification error for these various methods are tabulated in Table 8.7. For segmenting the image in Figure 8.7, the segmentation results seem to be robust against variations in the parameters of the MEC.

For the MEC approach, just as what we have described in Sub-section 8.2.1-8.2.5, we use a two feature-based classifiers as the experts. Table 8.7 shows the segmentation results for the various MEC schemes. SG stands for the stacked generalization approach while Gating stands for the gating approach. The various versions of the MEC are annotated in the caption for Table 8.7.

As shown in Table 8.7 once again, with the exception of the SG Network version of the MEC, the multi-experts approach segment the image in Figure 8.7 better than any single-model methods. No doubt that there are single model methods we have not implemented that could perform very well in segmenting Figure 8.7. Nevertheless, the MEC used here has inputs that are exactly the same as the feature-based experts. So, if another method could perform well on Figure 8.7, its inputs are likely not to be the same as those of the feature-based classifiers considered here. That new method's inputs could also be fed into the MEC to help the MEC to obtain even better results. Of course, this is

Algorithm Name	Classification Error
Feature 1 (SAR)	3.92%
Feature 2 (GMRF)	4.03%
SG Network	4.22%
SG Coincidence (1.0)	3.56%
SG Coincidence (2.0)	3.30%
SG Coincidence (5.0)	3.30%
SG Coincidence (Infinity)	3.30%
Gating Logistic 0 hidden nodes	3.54%
Gating Softmax 0 hidden nodes	3.40%
Gating Logistic 20 hidden nodes	3.53%
Gating Softmax 20 hidden nodes	3.53%

Table 8.7 Segmentation results on the phantom image in Figure 8.7 by various MEC classifiers with two feature-based experts. The top two rows show the feature-based expert segmentation performance. SG stands for stacked generalization approach, and Gating stands for the gating network approach. SG Network refers to the use of a single hidden layer with 40 hidden nodes) MLP trained for 5 epoches. The parenthesized numbers next to the SG Coincidence represents the REJECT threshold. Finally, the Gating network either has no hidden layer (0 hidden nodes) or has 1 hidden layer (with 20 hidden nodes).

all speculation because we have not encountered such a method for testing yet. To test out the speculation we just mentioned should be top priority for any further work in this multi-experts area.

(ii) phantom image with MRF states

We now try to segment a phantom image with a different topology from what we

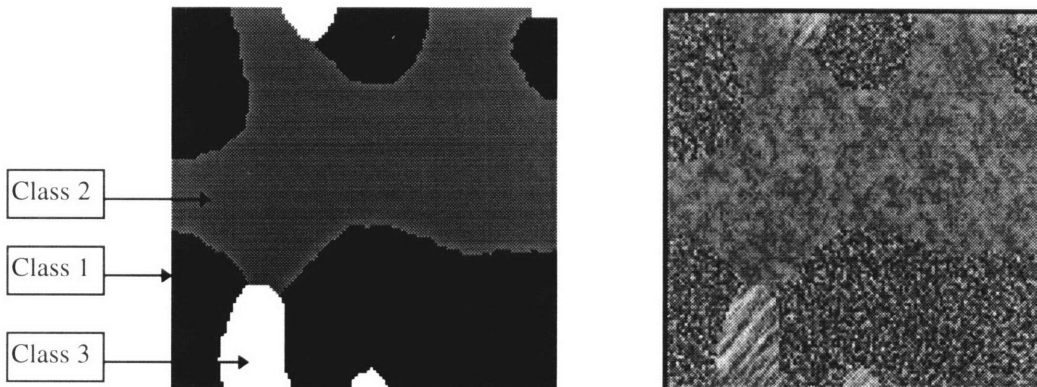


Figure 8.10 A phantom image with the underlying labels as a MRF. The left hand image is the true underlying labels for the right hand image. Class 1 is a simulated SAR region with parameters $\{0.2, 0.2, 0.2, 0.2\}$. Likewise, Class 3 is also a simulated SAR region with parameters $\{0.0, 0.5, 0.5, 0.0\}$. These parameters correspond to Equation 5.15. Class 2 has intensities from a Gaussian density of mean 150 and variance of 900.

have been using in this chapter to ensure that the topology of the different regions does not affect the relative performance of different segmentation methods.

Shown on the right hand side of Figure 8.10 is a phantom image whose underlying class regions are generated using a Gibbs sampler with a $\beta=1.5$ for 2000 iterations. (The

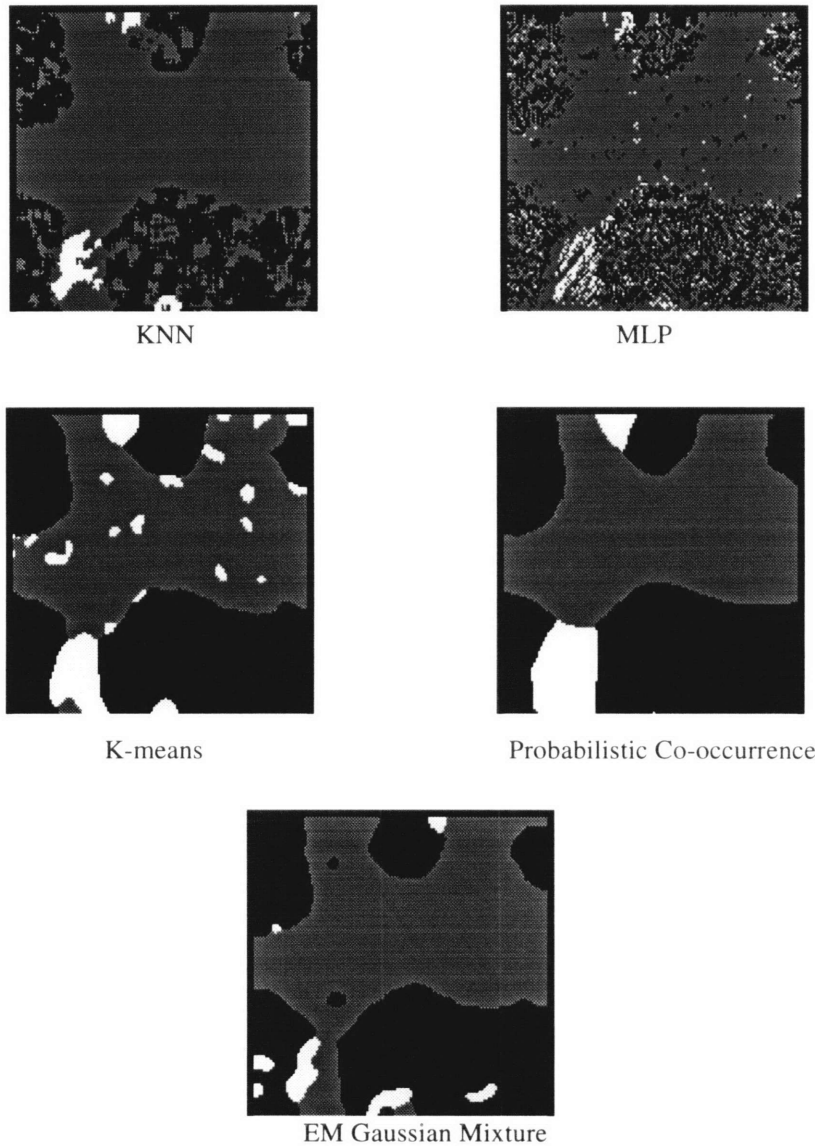


Figure 8.11 Segmentation results on Figure 8.10 by various single model methods. The true underlying class labels are given in Figure 8.10

Gibbs sampler was originally introduced to restore images by imposing a MRF priors on the local spatial field [Geman and Geman, 1984]. So the class labels actually form a MRF.) A brief introduction of Gibbs sampler has been given in Chapter 2.

Segmentation of Figure 8.10 by various single model segmenters are shown in Figure 8.11. The K in KNN is 24. MLP has an input intensity map of 9x9, one hidden layer of 40 units and has been trained for 100 epoches to achieve an output mean square error of 0.15. Both K-means and EM Gaussian mixtures have 10 iterations to estimate their model parameters. Finally, the probabilistic co-occurrence approach used 4 co-occurrence matrix with 8 uniform bins and an input region size of 25x25 [Lohmann, 1995].

<i>Algorithm Name</i>	<i>Classification Error</i>
<i>KNN</i>	<i>17.92%</i>
<i>MLP</i>	<i>29.48%</i>
<i>K-means</i>	<i>5.25%</i>
<i>Probabilistic Co-occurrence</i>	<i>5.07%</i>
<i>Gaussian Mixture (EM)</i>	<i>8.05%</i>

Table 8.8 Segmentation results on the phantom image in Figure 8.10 by various single model methods.

Table 8.8 shows the numerical results in segmenting Figure 8.10 by various single model methods.

In selecting the two optimal feature sets for the phantom image in Figure 8.10, another technique is attempted since we only have an odd number of classes. The first *optimal* set of the features is chosen in the usual way. The second optimal set is chosen also in the same as the first feature set with the condition that no feature in the second optimal feature set can also be in the first optimal feature set.

By following the above two steps, the two *optimal* feature sets for the phantom image shown in Figure 8.10 have been chosen to be:

$$\begin{aligned}
 \text{optimal feature sets: } & \{1, 4, 13, 10\} \\
 & \{3, 11, 21, 7, 16\}
 \end{aligned} \tag{8.4}$$

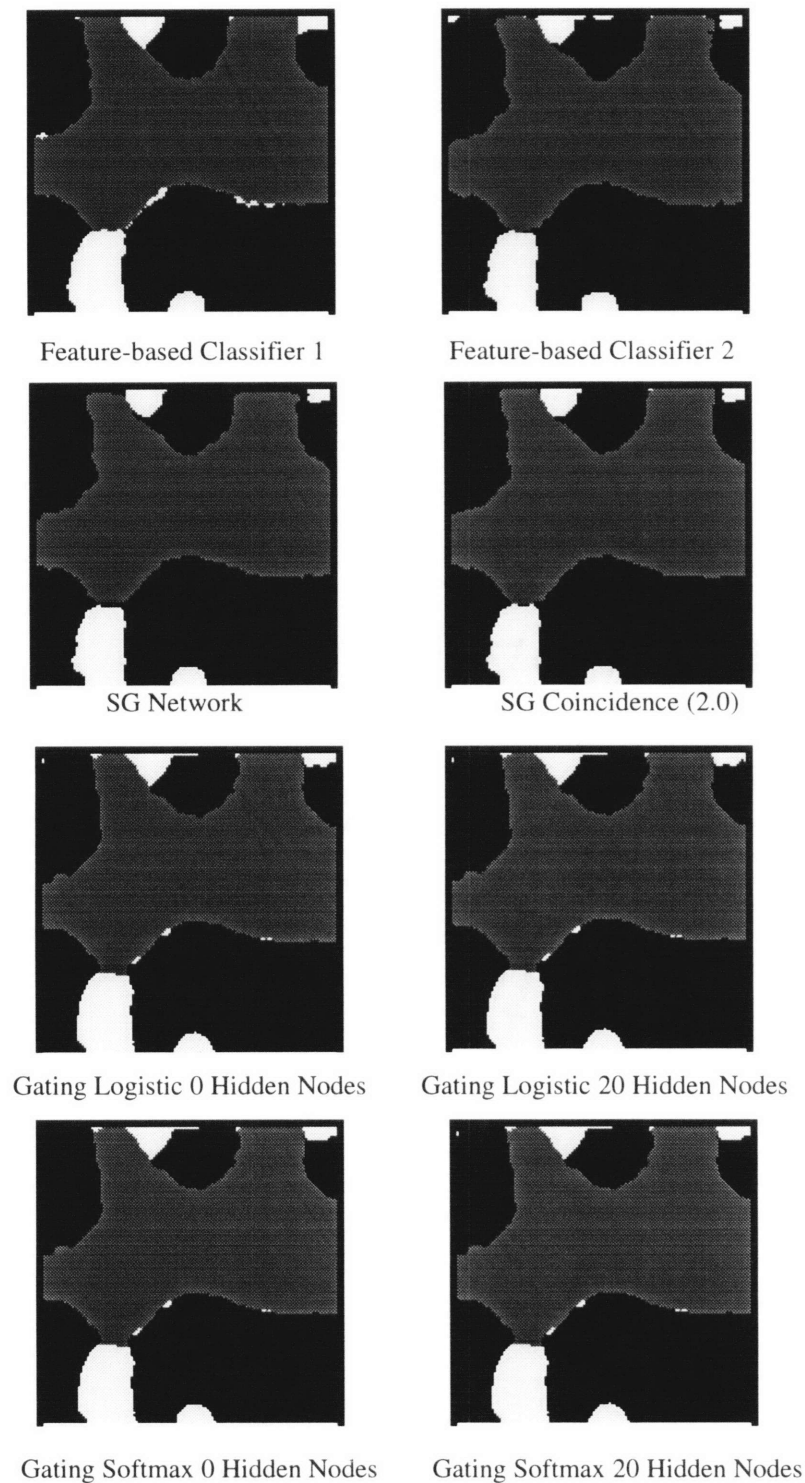


Figure 8.12 Segmentation results on phantom image in Figure 8.10. For the meaning of the notation below each figure, refer to Figure 8.9.

<i>Algorithm Name</i>	<i>Classification Error</i>
<i>Feature classifier 1</i>	<i>2.91%</i>
<i>Feature classifier 2</i>	<i>3.30%</i>
<i>SG Network</i>	<i>3.64%</i>
<i>SG Coincidence (1.0)</i>	<i>2.64%</i>
<i>SG Coincidence (2.0)</i>	<i>2.64%</i>
<i>SG Coincidence (5.0)</i>	<i>2.64%</i>
<i>SG Coincidence (Infinity)</i>	<i>2.64%</i>
<i>Gating Logistic 0 hidden nodes</i>	<i>3.28%</i>
<i>Gating Softmax 0 hidden nodes</i>	<i>3.28%</i>
<i>Gating Logistic 20 hidden nodes</i>	<i>3.30%</i>
<i>Gating Softmax 20 hidden nodes</i>	<i>3.30%</i>

Table 8.9 Segmentation results on the phantom image in Figure 8.10 by various MEC classifiers with two feature-based experts. For the meaning of the algorithm name, refer to the caption in Table 8.7.

Table 8.9 shows the MEC segmentation results. Comparing with the results in Table 8.8 for the single model methods, we once again see that all the MEC approaches achieve low classification error while only a few of the single model methods -- the two feature-based classifiers -- achieve such low error.

Referring again to Table 8.9, we see that for the coincidence approach to SG and the gating network approach to MEC, the classification performance is not affected very much by changes in the MEC's parameters.

8.2.7 Comments on the Segmentation Results

This section has evaluated several single model segmenation methods and the MEC approach. The general observation is that on average, MEC outperforms every single model approach.

In the following section, the MEC will be applied to different types of real world images. Unfortunately, unlike the phantom images we have considered in this section, the true class labels of real world images are unknown. Therefore, performance comparison among different techniques can only be done qualitatively -- through visual inspection.

8.3 Real World Image Segmentation

With the quantitative segmentation results on various types of images shown in Section 8.2, we expect the MEC to perform better than most single model methods. In this section, we apply the MEC, along with those single model methods to segment two real world images. Both of these images are related to the auditory system. The first one is a slice of an X-ray CT of the Visible Man's outer and middle ear region, with a clear view of the external auditory canal. The second image is also a middle ear CT slice of a gray whale.

(i) Visible Man Ear

From the Visible Man data set, we have chosen one image (top of Figure 8.13) to demonstrate the segmentation performed by different algorithms. The results are shown in the lower portion of Figure 8.13. An expert user randomly chooses 20 learning sets and 20 testing sets for each of the four classes before training the MEC. Two sets of optimal features are chosen according to the method described for the phantom image with MRF state labels in the last section. These two feature sets are:

optimal feature sets for the Visible Man Ear image

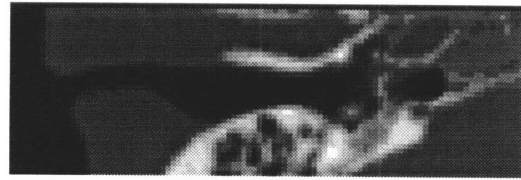
Optimal Feature Set 1 : {1, 17, 13}

Optimal Feature Set 2 : {3, 22, 9}

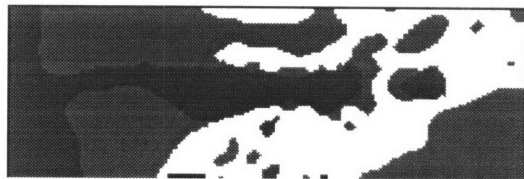
Although no quantitative information can be obtained about the segmentation results, by just looking at the segmented images, we can see that all the MEC approaches have obtained pretty good segmentation results compared to the single model methods.

(ii) Gray Whale Ear

To convince ourselves that the good segmentation performed by MEC in Figure 8.13 is not just luck, we will try all the segmentation algorithms on another real world image. This second image belongs to a CT slice of a gray whale middle ear region. Just like the Visible Man ear case, the two optimal feature sets chosen are:



Original Image



KNN



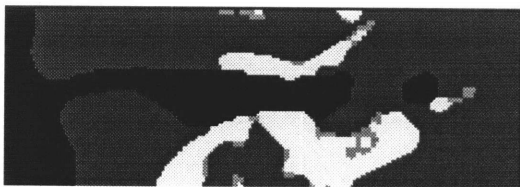
MLP



K-means



Probabilistic Co-occurrence



EM Mixture of Gaussians



Feature-based Classifier 1



SG Coincidence (5.0)



Feature-based Classifier 2



Gating Logistic (0 Hidden Nodes)



Gating Softmax (0 Hidden Nodes)

Figure 8. 13 Different segmentation results on segmenting the Visible Man X-ray CT slice at the top of the page. For the meaning of the notation below each figure, refer to Figure 8.9.

optimal feature sets for the Visible Man Ear image

Optimal Feature Set 1 : {3, 7, 8, 2}

Optimal Feature Set 2 : {1, 6, 0, 5, 1, 16, 17, 19}

The segmentation results performed by the different segmentation algorithms are shown in Figure 8.14. The top figure in 8.14 is the original image. An expert user randomly chooses 20 learning sets and 20 testing sets for each of the four classes before training the MEC. Although we cannot have a quantitative evaluation of these segmentation results, we can see from 8.14 that all the MEC segmentations are pretty good, relative to all other available methods.

8.4 Discussion of Results

We have presented experiments and results on both synthetic and real world images. In every case, the MEC approach consistently performs well on different image segmentation tasks. Quantitatively, we have measured the classification error of various image segmentation techniques on several different phantom images. We have seen that in every case, the MEC implementations have achieved either the lowest or one of the lowest classification error. By varying different parameters of the MEC, we have also observed that the segmentation results are not changed significantly. This observation suggests that the MEC approach to image segmentation is a robust and stable approach.

In segmenting real world images, only qualitative observation can be made because the underlying class labels are generally unknown. For segmentation tasks on these real world images, we have also observed that the MEC implementations have achieved encouraging segmentations on these images, in comparison with the other single model methods we have available.

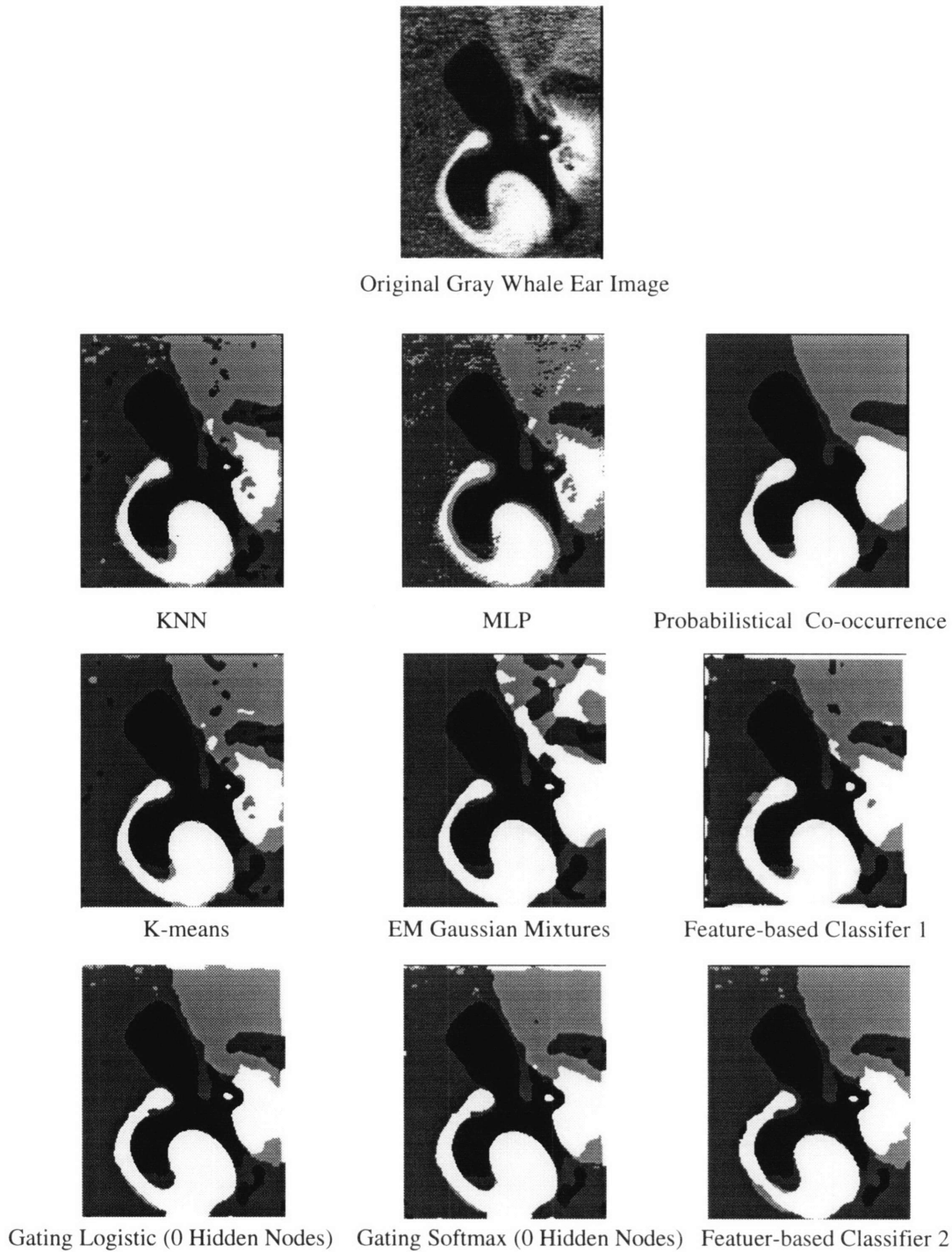


Figure 8.14 Image segmentation by various techniques on the CT gray whale ear image at top of the page. For the meaning of the notation below each figure, refer to Figure 8.9.

Segmentation of real world images always raises the question of how do we know when we have found the true region boundaries? The fact is, unless we have *a priori* knowledge about the image, there is no way we can verify our segmentation results. For example, for certain medical images, a histology study of the actual organ, cells, or other objects in the image could verify within certain error bound the accuracy of a segmentation task. However, this procedure is very time consuming and is generally unavailable to most image processing practitioners or to most physicians.

Another way to obtain the *a priori* knowledge of the true underlying class labels is to construct phantom images, as we have done in this thesis. By consistently verifying that a segmentation technique performs better than other techniques by the use of phantom images, image processing practitioners can build up their faith in that technique. Of course, if the qualitative evaluation of this technique on real world images is not good, even if it performs well on phantom images, people would probably be reluctant to use this method.

In this chapter, we have verified by using several different types of phantom images that the MEC approach consistently performs well with respect to the other segmentation techniques considered in this thesis. Furthermore, we have shown that the qualitative observation of the MEC segmentation results on real world images are very encouraging.

Chapter 9

CONCLUSION

A multi-experts approach to image classification and segmentation has been proposed. This thesis has argued that this approach is more appropriate than many single-model approach for processing real world images. Real world images are likely to be results of various image formation processes. Modeling all these process simultaneously is a very challenging task and no method proposed so far can claim success in this respect. This thesis presents a possible approach to deal with such a task through the multi-experts architecture. The key problem addressed by this thesis is how to combine different experts to yield sensible combined results.

This thesis has proposed two classifier designs for combining different methods (or experts) in the multi-experts approach. These two classifiers are based on two recently proposed statistical techniques. The first one is a gating network approach based on the work of Jordan and Jacobs (1994). The idea behind this approach is to weigh the outputs of different experts by appropriate priors. These priors are determined by a gating network which partitions the input space to allow the experts better suited for certain inputs to have higher weights for those inputs. At the same time, the experts which perform poorly on certain input patterns automatically receive lower weighting factors. The second proposed multi-experts classifier is based on Wolpert's (1992) idea of stacked

generalization. The stacked generalizer makes generalization to yield the overall outputs by observing the predictions made by the classification experts.

Experimental comparisons have been made on the performances of many traditional image segmentation methods as well as the Multi-Experts Classifier (MEC). We have quantitatively tested these methods on various phantom images and have observed that the MEC, on average, does better than any of the other methods. The quantification of “better” is measured by the number of corrected labeled pixels. These preliminary results are very encouraging for the multi-experts approach to image segmentation (or pixel classifications).

Future Works

To the fields of image classification and segmentation, this thesis is like an explorer who has just landed on a big tropical island. He is only able to gather only a few items on the island to bring back with him because his ship is not big. There could be many valuable resources on this island, but the explorer is not able to find out about them because his stay on the island constrained by time. Nevertheless, based on those resources he can see, the explorer has a strong feeling that this island has tremendous development potential.

This thesis has only explored a small territory in using the multi-experts approach to image classification and segmentation. As mentioned in the introduction, robust processing of real world images is likely to require more than one single model because real world images are rarely entirely composed of stationary signals. Real world images have regions that are results of different image formation processes such as shading, texture, motion, etc. There are many possible directions for future works, which include the following:

- Perhaps, the most straightforward extension to this work is to explore the incorporation of additional experts. Several issues have to be resolved in this respect, this bullet point considers one of them. For the gating network approach, the current inputs to the gating networks are the *optimal* feature sets. These features are the entire

set of inputs to all the experts available. Using these features make sense because the function of the gating network is to partition the entire input space to the different experts. Now, for example, if the probabilistic co-occurrence method is to be included as an additional expert. It is not immediately clear how to add features to the gating network to accommodate for the co-occurrence method. In the case of the K-means, on the other hand, the inputs to the K-mean could be changed to the optimal feature sets. No additional to the gating network input is required. Nevertheless, investigation has to be conducted on how to add new input features to the gating network as new experts are added.

- For the stacked generalization approach of the MEC, when a new expert is added, no difficulty as the gating network approach is encountered here. However, as more and more experts are added while the learning set remains constant, the resulting learning problem could be quite difficulty. Therefore, investigation has to be conducted to explore how to deal with this increased complexity in the learning algorithm. In the case of the coincidence tensor approach of the stacked generalizer, more experts means that the dimension of the coincidence tensor is increased more and more. If the learning set remains constant, many elements of the tensor would be zero. Sparse entries could lead to error in generalization. Therefore, research is definitely needed to deal with this problem.
- From the parallel nature of the gating network and the serial nature of the stacked generalizer, one can see that these two methods can be combined. For example, results from the stacked generalizer could be considered as another expert. In the gating network framework, if there are N experts originally, an additional stacked generalizer would increase the number of inputs to the gating network to $N+1$.
- A more theoretical question is what is the limit of the multi-experts approach. By the *limit*, we mean if there exists a minimum error that this approach cannot eliminate either by further partitioning the input space or by learning from the output of additional experts. A related question is, how do we know whether by incorporating

an additional expert, the performance of the overall system would be increased or decreased?

This thesis has only demonstrated the application of the multi-experts approach to the image classification and segmentation tasks. As a general framework for approaching pattern recognition or vision tasks, the multi-experts approach could serve to advance algorithms for more robust pattern recognition or image processing. The potential of this approach is yet to be fully realized.

Appendix A

A PROBABILISTIC APPROACH TO CO-OCCURRENCE MATRIX

In Chapter 2 and Chapter 5, features based on co-occurrence matrices have been discussed. This appendix details a novel approach proposed by Lohmann (1995) using multinomial distributions to parametrize the co-occurrence matrix coefficients.

The key argument that Lohmann used to support his approach is that all of the popular features derived from the co-occurrence matrices fail to retain the sufficiency to represent a textural image without losing any information. In other words, the extracted features cannot by themselves be used to derive the original co-occurrence matrix. These features can definitely not represent the texture information completely. An assertion of his paper is that the four co-occurrence matrices contain sufficient information to represent textural images sufficiently. Justification of his new approach is given by the observing the striking similarity between an original texture and a simulated texture using the co-occurrence matrices. Initial results on Brodatz textures and LANDSAT images are very encouraging and have aroused the author to adopt it for robust texture classification. Most impressive about this approach is that no feature-selection is required while the technique seems to work very well with a variety of images. In the following paragraphs, a brief description of this new approach is given. The background of using co-occurrence matrix for classifying images has been given in the second chapter of this thesis.

For simplicity of notation, consider a texture with only one co-occurrence matrix with the normalized coefficients c_{ij} such that $\sum_{i,j} c_{ij} = 1.0$. Let x_{ij} be the number of pairs of pixels with gray values (i, j) or (j, i) in a population of n pixels, then $x_{ij} = [n \times c_{ij}]$ and $\sum_{i,j} x_{ij} = n$. For classifying any pattern into N classes, the natural probability model to use is multinomial density. For this problem, the multinomial density can be used to model the probability of selecting x_{ij} number of pairs given the co-occurrence matrix. If c_{ij} is the co-occurrence matrix coefficient for the entire image, then the likelihood of finding the feature vector \mathbf{x} is:

$$p(\mathbf{x}) = n! \prod_{i=1, j=1}^m \frac{c_{ij}}{x_{ij}!} \quad (\text{A.1})$$

A given feature vector \mathbf{x} can be regarded as evidence supporting or refuting a classification hypothesis. To quantitatively measure the strength of the evidence, one can apply Bayesian concepts. Let z represents a class or state variable identifying \mathbf{x} to a certain class k , then,

$$p(z = k | \mathbf{x}) = \frac{p(\mathbf{x} | z = k) p(z = k)}{\sum_m p(\mathbf{x} | z = m) p(z = m)} = \frac{p(\mathbf{x} | z = k) p(z = k)}{p(\mathbf{x})} \quad (\text{A.2})$$

Usually the prior probability $p(z)$ is known and is taken to be equal across all classes, i.e. $p(z=k) = 1/K$ for all K classes. To classify \mathbf{x} into one of K classes, Lohmann applies a concept in information theory known as *mutual information*, $I(\mathbf{x}, z)$, which is essentially the log likelihood of the Bayesian probability $p(z=k|\mathbf{x})$. If $I(\mathbf{x}, z) > 0$, the \mathbf{x} supports classification hypothesis z . If $I(\mathbf{x}, z) < 0$, the \mathbf{x} refutes hypothesis z . Finally, if $I(\mathbf{x}, z) = 0$, the \mathbf{x} neither supports nor refutes z . Using the multinomial distribution as the parametric model for modelling the co-occurrence matrix coefficient features,

$$p(\mathbf{x} | z = k) = n! \prod_{i=1, j=1}^m \frac{c_{ij,k}}{x_{ij}!} \quad (\text{A.3})$$

where $c_{ij,k}$ represents the normalized co-occurrence matrix coefficient for class k .

Now, we can express the *mutual information* as the following:

$$I(\mathbf{x}, z = k) = \log \frac{p(\mathbf{x}|z = k)}{p(\mathbf{x})} = \sum_{i,j} x_{ij} (\log c_{ij,k} - \log c_{ij}) \quad (\text{A.4})$$

This quantity can be readily calculated provided that $q_{ij,k}$ and c_{ij} are both greater than 0. For dealing with multiple evidence from several co-occurrence matrices of a given image, Lohmann assumes that the coefficients between different co-occurrence matrices are independent. For two co-occurrence matrices, the *mutual information* can be expressed as,

$$\begin{aligned} I(\mathbf{x}^1, \mathbf{x}^2, z = k) &= \log \frac{p(\mathbf{x}^1, \mathbf{x}^2|z = k)}{p(\mathbf{x}^1, \mathbf{x}^2)} = \log \frac{p(\mathbf{x}^1|z = k)p(\mathbf{x}^2|z = k)}{p(\mathbf{x}^1)p(\mathbf{x}^2)} \\ &= \log \frac{p(\mathbf{x}^1|z = k)}{p(\mathbf{x}^1)} + \log \frac{p(\mathbf{x}^2|z = k)}{p(\mathbf{x}^2)} = I(\mathbf{x}^1, z = k) + I(\mathbf{x}^2, z = k) \end{aligned} \quad (\text{A.5})$$

where \mathbf{x}^1 and \mathbf{x}^2 are the feature vectors for two different co-occurrence matrices. Although the coefficients' independence assumption is highly doubtful, Lohmann observes that the error incurred by this assumption is small and can be tolerated. Generalization for more than two co-occurrence matrices is obvious.

As we have mentioned briefly earlier, the most impressive aspect of this novel approach is that no “optimal” features set is required be chosen first before a good classification results can be obtained. In this respect, this approach is like the non-parametric approaches, but it is definitely a parametric method. But the parametrization operations are done automatically without any intervention from the algorithm designer.

Appendix B

THE INCOMPLETE DATA PROBLEM AND EXPECTATION MAXIMIZATION

This appendix describes the incomplete data problem and its application for finding the maximum likelihood (ML) estimates of incomplete data through expectation maximization (EM). This EM approach was first formulated by Dempster, Laird and Rubin in their landmark paper [Dempster, *et.al.*, 1977] and has been the most successful method for approaching the incomplete data problem. Such approach has been extensively used by many researchers for finding the ML estimates of incomplete data in various problems. A few recent applications can be found in [Jordon, *et.al.*, 1993a, 1993b; Liang, *et.al.*, 1994; Zhang, *et.al.*, 1994]. This thesis has used EM to find the ML parameter estimates of various image models, Details of which can be found in Chapter 4.

Consider an image with observed intensity vector \mathbf{y} , for every pixel i there exists both the observed intensity value y_i and the unobserved class label z_i . Since \mathbf{y} does not directly give information about the values of \mathbf{z} , the observable \mathbf{y} is often called the incomplete data. The set of random variables $\mathbf{x} = \{ \mathbf{y}, \mathbf{z} \}$, which includes both the observed \mathbf{y} and the unobserved \mathbf{z} , is called the complete data. We model the incomplete data \mathbf{y} by the probability density function (pdf) $f(\mathbf{y}|\Phi)$ where Φ is the set of parameters to be estimated for the pdf. The ML approach for estimating the parameters through the incomplete data \mathbf{y} can be expressed by the following estimator:

$$\Phi^* = \arg \max_{\Phi} f(y|\Phi).$$

Dempster and his colleagues in [Dempster, *et.al.*, 1977] presents solution to this ML problem in an iterative fashion through two steps, the (E)xpectation step and the (M)aximization step. Assume an initial estimate of the parameters is Φ^0 and p stands for the current iteration:

E-step: *estimate the expectation of the log-likelihood of the complete data using the current parameter estimates $\Phi^{*(p)}$. Let $Q(\Phi|\Phi^{*(p)})$ represent this expectation:*

$$Q(\Phi|\Phi^{*(p)}) = E[\log(f(x|\Phi)|y, \Phi^{*(p)})]$$

M-step: *maximize the expectation of the log likelihood by finding the next best parameter estimate $^{(p)}$:*

$$\Phi^{*(p+1)} = y)$$

The paper [Dempster, *et.al.*, 1977] also presents derivation of the EM convergence results. The E and M steps turn out to monotonically increase the $E[\log(f(x|\Phi))]$ in every iteration.

Appendix C

A PROBABILISTIC MODEL FOR THE GATING NETWORK

In this appendix, the gating network discussed in Chapter 3 will be discussed in a probabilistic framework. This framework is derived from first principles based on Bayes' theorem. Much of the discussion here is quite intuitive and has very likely already appeared quite extensively in the statistical and learning community such as in [McCulloch and Nelder, 1984; Dobson, 1989].

The problem we are concerned about here is the N-way classification of a given input vector \mathbf{y} . Assume that each of the N different classes is parametrized by a set of parameters, $\Phi_n \in \{ \Phi_1, \Phi_2, \dots \Phi_N \}$. The probability of input \mathbf{y} in state n can be represented by the state variable z , where $0.0 \leq z \leq 1.0$. We can view the classification problem as finding the best state z with parameters Φ_n^* to be associated with the given input \mathbf{y} , which can be expressed as :

$$z^* = \arg \max_{z, \Phi} p(z|\mathbf{y}, \Phi) \quad (\text{C.1})$$

where $p(z|\mathbf{y}, \Phi)$ is the posterior probability of the data in state z . Let's consider a very general class of probability distribution for the input data \mathbf{y} -- the exponential family which includes the normal, multinomial, Poisson, gamma, and many of the other familiar distributions. The general form of the exponential family distribution can be expressed as [Jordan and Jacobs, 1994]:

$$f(y|\mu, \sigma) = k \exp\{(g(\mu)y - b(\mu))/\sigma + c(y, \mu)\} \quad (C.2)$$

where k is a normalizing constant, μ represents model parameters called the “natural parameters”, and σ represents some dispersion constants of the distribution. $g(\cdot)$, $b(\cdot)$, and $c(\cdot)$ are functions that depend on the type of exponential distributions involved. With the data distribution function, we can write the posterior probability as :

$$\begin{aligned} p(z|y) &= \frac{f(y|z, \Phi_n) p(z, \Phi_n)}{\sum_j f(y|z, \Phi_n) p(z, \Phi_n)} \\ &= \frac{e^{\{(g(\mu_n)y - b(\mu_n))/\sigma_n + c(y, \mu_n)\}} e^{\log p(z, \Phi_n)}}{\sum_j e^{\{(g(\mu_n)y - b(\mu_n))/\sigma_n + c(y, \mu_n)\}} e^{\log p(z, \Phi_n)}} \end{aligned} \quad (C.3)$$

To illustrate concretely what the above expression is, we consider the Gaussian distribution family. Here $\Phi_n = \{ \mu_n, \sigma_n \}$. To avoid excessive notations, we assume that the covariance matrices are diagonal :

$$f(\Phi_n|x) = \frac{e^{-1/2\sigma_n^2 (y - \mu_n)^2} e^{\log p(z, \Phi_n)}}{\sum_j e^{-1/2\sigma_n^2 (y - \mu_n)^2} e^{\log p(z, \Phi_n)}} \quad (C.4)$$

For $N = 2$, the above equation is reduced to the following familiar expression when we compute the posterior probability of input y being in class 0 :

$$\begin{aligned}
p(z_0|x, \Phi_0) &= \frac{e^{-\frac{1}{2\sigma_0^2}(y-\mu_0)^2} e^{\log p(z_0, \Phi_0)}}{e^{-\frac{1}{2\sigma_0^2}(y-\mu_0)^2} e^{\log p(z_0, \Phi_0)} + e^{-\frac{1}{2\sigma_1^2}(y-\mu_1)^2} e^{\log p(z_1, \Phi_1)}} \\
&= \frac{1}{1 + e^{\{(y-\mu_0)^2/\sigma_0^2 - (y-\mu_1)^2/\sigma_1^2\} \log p(z_1, \Phi_1) / p(z_0, \Phi_0)}}} \quad (C.5) \\
&= \frac{1}{1 + e^{\mathbf{w}_1^T y + w_0}}
\end{aligned}$$

where \mathbf{w}_1^T is the transpose of $(\mu_0/\sigma_0 - \mu_1/\sigma_1)$ and w_0 is the constant $\{(\mu_1^2/\sigma_1^2 - \mu_0^2/\sigma_0^2) - \log p(z_0, \Phi_1) / p(z_1, \Phi_0)\}$. The simple expression (5) above is the well known *logistic* function. $p(z_1|y, \Phi_1)$ has a similar expression with appropriate subscript changes.

For N-way classification where $N > 2$, a natural choice for the state likelihood is the multinomial distribution as discussed in Chapter 3 of this thesis. Consider m trials, the likelihood of the states has the form:

$$f(z_1, z_2, \dots, z_N | y, \Phi) = m! \prod_{i=1}^N \frac{p(z_i | y, \Phi)^{z_i}}{z_i!} \quad (C.6)$$

where $\sum_{i=1}^N p(z_i) = 1.0$, $\sum_{i=1}^N z_i = m$, and for classification problems, m is one. The exponential form of this likelihood is :

$$f(z_1, z_2, \dots, z_N | y, \Phi) = \exp \left\{ \ln \frac{m!}{z_1! z_2! \dots z_N!} + \sum_{i=1}^N z_i \ln p(z_i | y, \Phi) \right\} \quad (C.7)$$

The posterior probability in class $z_i = k$ is derived in [Jordan and Jacobs, 1994]. The probability of state z_i can be expressed as the so-called *softmax* function:

$$p(z_i | y, \Phi_i) = \frac{e^{\mathbf{w}_{i1}^T y + w_{i0}}}{\sum_j e^{\mathbf{w}_{j1}^T y + w_{j0}}} \quad (C.8)$$

This function is a reasonable expression when we consider expression (C.3), which is already in the form of (C.8). When we incorporate the constant exponential factors w_{i0} into the vector dot product term by augmenting the \mathbf{y} vector by a constant 1 element, we get the following simple form for the softmax function :

$$p(z_i|\mathbf{x}, \Phi_i) = \frac{e^{\mathbf{w}_i^T \mathbf{y}}}{\sum_j e^{\mathbf{w}_j^T \mathbf{y}}} \quad (\text{C.9})$$

This softmax function soft-partitions an input space because a given \mathbf{y} has non-trivial membership in all of the classes since Equation (C.9) is never zero for any \mathbf{w}_i . Equation (C.9) gives a probability measure of how likely input \mathbf{y} is in each of the K classes.

Appendix D

EXAMPLE OF A LEARNING SET INPUT FILE

Here is a sample learning sets input file for evaluating the performances of different segmentation algorithms on the image *whale.abd*. The first column of the learning pairs section represents the class label, the second column represents the learning pair number in the class. Finally, the coordinate of the learning point is written, the value of which is obtained directly from the image.

image file:	whale.abd
number of classes:	4
number of learning set for class 0:	20
number of learning set for class 1:	20
number of learning set for class 2:	20
number of learning set for class 3:	20
1 (1) (33, 16)	
1 (2) (22, 42)	
1 (3) (45, 35)	
1 (4) (68, 26)	
1 (5) (83, 21)	
1 (6) (96, 41)	
1 (7) (104, 23)	
1 (8) (106, 110)	
1 (9) (68, 112)	
1 (10) (25, 109)	
1 (11) (28, 82)	
1 (12) (70, 84)	
1 (13) (44, 88)	
1 (14) (36, 52)	
1 (15) (64, 62)	
1 (16) (88, 41)	
1 (17) (87, 82)	
1 (18) (104, 66)	
1 (19) (121, 51)	
1 (20) (74, 123)	
2 (1) (201, 56)	
2 (2) (184, 40)	
2 (3) (164, 60)	
2 (4) (153, 92)	
2 (5) (136, 95)	

2 (6) (173, 121)
2 (7) (186, 86)
2 (8) (203, 104)
2 (9) (222, 86)
2 (10) (226, 55)
2 (11) (194, 34)
2 (12) (157, 30)
2 (13) (146, 48)
2 (14) (184, 19)
2 (15) (226, 17)
2 (16) (240, 21)
2 (17) (237, 88)
2 (18) (217, 94)
2 (19) (157, 16)
2 (20) (193, 108)
3 (1) (53, 142)
3 (2) (45, 164)
3 (3) (27, 152)
3 (4) (18, 185)
3 (5) (42, 191)
3 (6) (79, 176)
3 (7) (93, 149)
3 (8) (92, 199)
3 (9) (105, 177)
3 (10) (116, 192)
3 (11) (110, 222)
3 (12) (85, 224)
3 (13) (46, 220)
3 (14) (42, 246)
3 (15) (27, 215)
3 (16) (20, 234)
3 (17) (100, 238)
3 (18) (113, 206)
3 (19) (28, 193)
3 (20) (54, 234)
4 (1) (218, 156)
4 (2) (190, 155)
4 (3) (164, 166)
4 (4) (148, 193)
4 (5) (148, 151)
4 (6) (174, 144)
4 (7) (188, 199)
4 (8) (216, 191)
4 (9) (185, 233)
4 (10) (157, 235)
4 (11) (211, 242)
4 (12) (211, 215)
4 (13) (241, 206)
4 (14) (236, 154)
4 (15) (199, 200)
4 (16) (196, 150)
4 (17) (149, 138)
4 (18) (139, 171)
4 (19) (185, 147)
4 (20) (200, 225)

References

- T. Q. Bartlett, M. W. Vannier, D. W. McKeel, Jr., M. Gado, C. F. Hildebolt, and R. Walkup, "Iterative Segmentation of Cerebral Gray Matter, White Matter, and CSF: Photographic and MR Images," *Computerized Medical Imaging and Graphics*, 18(6), pp. 449-460, 1994
- J. Besag, "Spatial Interaction and Statistical Analysis of Lattice Systems", *Journal of the Royal Statistical Society (B)*, 36, pp. 192-236, 1972
- J. Besag, "On the Statistical Analysis of Dirty Pictures," *Journal of the Royal Statistical Society (B)*, pp. 259-302, 48(3), 1986)
- J. C. Bezdek, L. O. Hall, L. P. Clarke, "Review of MR Image Segmentation Techniques Using Pattern Recognition," *Medical Physics*, 20(4), pp. 1033-1048, Jul./Aug., 1993
- L. Breiman, "Stacked Regressions", *Technical Report, Statistics Department*, University of California at Berkeley, 1992
- L. Breiman, "Bagging Predictors," *Technical Report, Statistics Department*, University of California at Berkeley, 1994
- L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*, Wadsworth International Group, Belmont, CA, 1984
- J. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationships to Statistical Pattern Recognition", in *Neuro-computing: Algorithms, Architectures, and Applications*, eds. F. Fogelman-Soulie, and J. Herault, New York, Springer-Verlag, 1989
- P. Brodatz, *Textures -- a Photographic Album for Artists and Designers*, Dover, New York, 1966

- W. Buntine, "Decision Tree Induction Systems: a Bayesian Analysis," *Uncertainty in Artificial Intelligence 3*, L. N. Kanal, *et.al.*, (Eds.), North-Holland, pp. 109-127, 1989
- I. Carlbom, D. Terzopoulos, and K. M. Harris, "Computer-Assisted Registration, Segmentation, and 3D Reconstruction from Images of Neuronal Tissue Sections," *IEEE Trans. Medical Imaging*, 13(2), pp. 351-362, June 1994
- J. M. Carstensen, "Description and Simulation of Visual Texture", *Ph.D. Thesis*, IMSOR, Technical University of Denmark, 1992
- R. Chellappa, and S. Chatterjee, "Classification of Textures Using Gaussian Markov Random Fields", *IEEE Trans. Acoustics, Speech, and Signal Processing*, 33(4), pp. 959-963, Aug. 1985
- C. H. Chen, *Nonlinear Maximum Entropy spectral Analysis Methods for Signal Recognition*, Chichester, Research Studies Press Ltd., 1982
- L. D. Cohen, "On active contour models and balloons," *CVGIP Image Understanding*, 53(2), pp. 211-218, Mar. 1991
- G. R. Cross, and A. K. Anil, "Markov Random Field Texture Models", *IEEE Trans. PAMI*, 5(1), pp. 25-39, Jan. 1983
- L. S. Davis and A. Mitiche, "MITES: a Model-driven, Iterative Texture Segmentation Algorithm," *Comput. Graphics Image Processing*, 19, pp. 95-110, 1982
- A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society (B)*, 38, pp.1-38, 1977
- J. Dengler, S. Behrens, and J. F. Desaga, "Segmentation of Microcalcifications in Mammograms", *IEEE Trans. Medical Imaging*, Dec. 1993
- H. Derin, and H. Elliott, "Modeling and Segmentation of Noisy and Textured Images Using Gibbs Random Fields", *IEEE Trans. PAMI*, 9(1), pp. 39-55, Jan. 1987
- E. R. Dougherty, and C. R. Giardina, *Image Processing -- Continuous to discrete, Volume 1: Geometric, Transform, and Statistical Methods*, Englewood Cliffs, Prentice-Hall, Inc., 1987
- A. W. Drake, and R. L. Keeney, *MIT Video Course Manual: Decision Analysis*, Massachusetts Institute of Technology, Cambridge, MA, 1978
- J. Duryea, and J. M. Boone, "A Fully Automated Algorithm for the Segmentation of Lung Fields on Digital Chest Radiographic Images," *Medical Physics*, 22(2), pp. 183-191, Feb., 1995

- B. Efron, and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, 1993
- S. Finette, A. R. Bleir, and W. Swindell, "Breast Tissue Classification Using Diagnostic Ultrasound and Pattern Recognition Techniques: I. Methods of Pattern Recognition," *Ultrasonic Imaging*, 5, pp. 55-70, 1983 (as cited in [Momenan, et.al., 1994])
- J. H. Friedman, "Multivariate Adaptive Regression Splines", *Annals of Statistics*, 19, pp. 1-141, 1991
- K. S. Fu, "Syntactic Image Modelling Using Stochastic Tree Grammars", in *Image Modeling*, ed. A. Rosenfeld, New York, Academic Press, pp. 153-170, 1981
- K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Inc., 1990
- B. S. Garra, M. F. Insana, T. H. Shawker, R. F. Wagner, M. Bradford, and M. Russell, "Quantitative Ultrasonic Detection and Classification of Diffuse Liver Disease Comparison with Human Observer Performance," *Invest. Radiology*, 24(3), pp. 196-203, Mar. 1989 (as cited in Momenan, et.al., 1994)
- S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. PAMI*, 6 (6), pp. 721-741, Nov. 1984
- R. C. Gonzalez, R.E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, 1992
- W. E. L. Grimson, and T. Pavlidis, "Discontinuity Detection for Visual Surface Reconstruction," *Comput. Vision, Graphics Image Processing*, 30, pp. 316-330, 1985
- S. Grinaker, "Edge Based Segmentation and Texture Separation," *Proc. Fifth International Conference on Pattern Recognition*, Miami Beach, Florida, Dec 1-4, 1980, pp. 554-557
- M. Grunkin, "On the Analysis of Image Data Using Simultaneous Interaction Models", *Ph.D. Thesis*, IMSOR, Technical University of Denmark, 1993
- M. Grunkin, "The MORSE Estimator for Non-causal SAR models," submitted to *IEEE Trans. Image Processing*, 1995
- R. M. Haralick, "Statistical and Structural Approaches to Textures", *Proc. IEEE*, 67(5), pp. 786-804, May 1979
- R. M. Haralick, and L. G. Shapiro, "Image Segmentation Techniques", *Computer Vision, Graphics & Image Processing*, 29, pp. 100-132, 1985

-
- M. Hassner, and J. Sklansky, "The Use of Markov Random Fields as Models of Texture", *Computer Graphics and Image Processing*, 12, pp. 357-370, 1980
- J. Hertz, A. Krogh, R. G. Palmer, *Introduction to the Theory of Neural Computation: Lecture Notes Volumn 1, Santa Fe Institute Studies in the Sciences of Complexity*, Addison-Wesley Publishing Company, 1991
- K. Hornik, M. Stinchcombe, and H. White, "Multilayer Feedforward Networks, Are Universal Approximators", *Neural Networks*, 2, pp. 359-366, 1989
- R. A. Howard, "Decision Analysis: Perspectives on Inference, Decision, and Experimentation", *Proc. IEEE*, 58(5), May 1970
- Y. Hu, and T. J. Dennis, "Textured Image Segmentation by Context Enhanced Clustering", *IEE. Proc., Vision Image, and Signal Processing*, 141(6), pp. 413-421, Dec. 1994
- S. N. Jayaramamurthy, "Texture Discrimination Using Digital Deconvolution Filters", *Proc. 7th Internation Conference on Pattern Recognition*, Montreal, Canada, July 30-August 2, 1984, pp. 1216-1218
- M. I. Jordon, and R. A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm", *Neural Computation*, 6, pp. 181-214, 1994
- M. I. Jordon, and L. Xu, "Convergence Results for the EM Approach to Mixtures of Experts Architectures", *MIT A.I. Memo No. 1458*, Nov. 18, 1993
- R. L. Kashyap, and R. Chellappa, "Estimation and Choice of Neighbors in Spatial-Interaction Models of Images", *IEEE Trans. Information Theory*, 29(1), Jan. 1993, pp. 60-72
- R. L. Kashyap, R. Chellappa, and R. Khotanzad, "Texture Classification Using Features Derived From Random Field Models", *Pattern Recognition Letters*, 1, pp. 43-50, 1982
- R. L. Kashyap, and A. Khotanzad, "A Model-Based Method for Rotation Invariant Texture Classification", *IEEE Trans. PAMI*, 8(4), pp. 472-481, July 1986
- Kass, A. Witkin, and K. Terzopoulos, "Snakes: Active contour models," *Int. Journal Comput. Vision*, 1(4), pp. 321-331, 1988
- P. A. Kelley, H. Derin, K. D. Hartt, "Adaptive Segmentation of Speckled Images Using a Hierarchical Random Field Model", *IEEE Trans. Acoustics, Speech, and Signal Proc.*, 36(10), pp. 1628-1641, Oct., 1988

- D.R. Ketten, D.K. Odell, and D.P. Domning, "Structure, Function, and Adaptation of the Manatee Ear," *Marine Mammal Sensory Systems*, J. Thomas et al (Ed), Plenum Press, New York, pp. 77-95, 1992
- J. Kilday, F. Palmeri, and M. d. Fox, "Classifying Mammographic Lesions Using Computerized Image Analysis", *IEEE Trans. Medical Imaging*, pp. 664-669, Dec, 1993
- B. Kosko, *Neural Networks and Fuzzy Systems: a Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, Englewood Cliffs, 1992
- S. W. Kwok, and C. Carter, "Multiple Decision Trees," *Uncertainty in Artificial Intelligence 4*, R. D. Shachter, et.al., (Eds.), North-Holland, pp. 327-334, 1990
- S. M. LaValle, and S. A. Hutchinson, "A Bayesian Segmentation Methodology for Parametric Image Models," *IEEE Trans. PAMI*, 17(2), pp. 211-217, Feb., 1995
- H. Y. Lee, Extraction of Textured Regions in Aerial Imagery, *Proc. Image Understanding Workshop*, Arlington, VA, June 23, 1983, ed. L.S. Baumann, Science Appl., McLean, VA, pp. 298-303, 1983 (as cited in [Reed, et.al., 1993])
- C. Li, D. Goldof, and L. O. Hall, "Toward Automatic Classification and Tissue Labeling of MR Brain Images," *Proc. IAPR Workshop on Structural and Syntactic Pattern Recognition*, Singapore, World Scientific, 1992 (as cited in [Bezdek, et.al., 1993])
- Z. Liang, J. R. MacFall, and D. P. Harrington, "Parameter Estimation and Tissue Segmentation from Multispectral MR Images," *IEEE Trans. on Medical Imaging*, 13(3), pp. 441-449, Sept. 1994
- J. S. Lim, *Two-dimensional Signal and Image Processing*, Prentice Hall, Inc., 1992
- G. Lohmann, "Analysis and Synthesis of Textures: a Co-occurrence-based Approach", *Comput. & Graphics*, 19(1), pp. 29-36, 1995
- B. Lorensen, "Marching Through the Visible Man", Web Page with URL address, <http://www.ge.com/crd/ivl/vm/vm.html>, 1994
- A. Lundervold, and G. Storvik, "Segmentation of Brain Parenchyma and Cerebrospinal Fluid in Multispectral Magnetic Resonance Images", *IEEE Trans. Medical Imaging*, 14(2), pp. 339-349, June, 1995
- J. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations", *Proc. Fifth Berkeley Symposium on Math. Statist. and Prob.*, pp. 281-297, Berkeley, CA, 1967

- J. Mao, and A. K. Jain, "Texture Classification and Segmentation Using Multiresolution Simultaneous Autoregressive Models", *Pattern Recognition*, 25(2), pp. 173-188, 1992
- D. Marr, *Vision*, San Francisco, Freeman, 1982
- J. Marroquin, S. Mitter, and T. Poggio, "Computer Vision", *J. Amer. Statistical Assoc.*, 82, pp. 76-89, Mar. 1987
- P. McCullagh, and J. A. Nelder, *Generalized Linear Models*, London, Chapman and Hall, 1983
- H.E. Melton, Jr., and D.J. Skorton, "Real-time Automatic Boundary Detection in Echocardiography", *Proc. IEEE Ultrasonic Symposium*, pp. 1113-1117, 1992
- W. Mendenhall, D. D. Wackerly, R. L. Scheaffer, *Mathematical Statistics with Applications*, Duxbury Press, Belmont, CA., 1990
- N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equations of State Calculations by Fast Computing Machines," *Journal of Chem. Physics*, 21, pp. 1087-1091, 1953
- M. L. Minsky, and S. A. Papert, *Perceptrons*, Cambridge, MIT Press, 1969
- R. Momenan, R. F. Wagner, B. S. Garra, M. H. Loew, and M. F. Insana, "Image Staining and Differential Diagnosis of Ultrasound Scans Based on the Mahalanobis Distance", *IEEE Trans. Medical Imaging*, 13(1), Mar. 1994
- W. Niblack, R. Barber, W. Equitz, M. Flickner, E. Glasman, D. Petkovic, P. Yanker, and C. Faloutsos, "The QBIC Project: Querying Images by Content Using Color, Texture and Shape," *Tech. Rep. RJ 9203 (81511)*, IBM Research Division, Feb. 1993 (as referenced in [Picard and Minka, 1995]).
- P. P. Ohanian, and R. C. Dubes, "Performance Evaluation for Four Classes of Texture Features", *Pattern Recognition*, 25(8), pp. 819-833, 1992
- Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley Publishing Company, Inc, 1989
- T. N. Pappas, "An Adaptive Clustering Algorithm for Image Segmentation," *IEEE Trans. Signal Processing*, 40, pp. 901-914, Apr., 1992
- R. W. Picard, and T. P. Minka, "Vision Textures for Annotation", *Journal of Multimedia Systems*, 3, pp. 3-14, 1995
- I. Pitas, *Digital Image Processing Algorithms*, Prentice Hall International (UK) Ltd, 1993

- T. Poggio, V. Torre, and C. Koch, "Computational Vision and Regularization Theory", *Nature*, 317, pp. 314-319
- J. R. Quinlan, "Induction of Decision Trees", *Machine Learning*, 1, pp. 81-106, 1989
- T. R. Reed, and J. M. Hans du Buf, "A Review of Recent Texture Segmentation and Feature Extraction Techniques", *CVGIP : Image Understanding*, 57(3), pp. 359-372, May, 1993
- A. Pentland, R. W. Picard, S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases", *MIT Media Laboratory Perceptual Computing Technical Report No. 255*, Nov. 1993
- D. A. Pomerleau, ALVINN: an Autonomous Land Vehicle in a Neural Network," in *Advances in Neural Information Processing Systems I*, ed. D. S. Touretzky, Morgan Kaufmann, pp. 305-313, 1989
- J. C. Russ, *The Image Processing Handbook*, CRC Press, Inc., 1992
- D. E. Rumelhart, J.L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, 1986
- M. Self, and P. Cheeseman, "Bayesian Prediction for Artificial Intelligence," *Proc. Uncertainty in Artificial Intelligence*, Seattle, July, 1987
- J. F. Silverman, and D. B. Cooper, "Bayesian Clustering for Unsupervised Estimation of Surface and Texture Models", *IEEE Trans. PAMI*, 10(4), pp. 482-496, July, 1988
- M. Spann, and R. Wilson, "A Quad-tree Approach to Image Segmentation which Combines Statistical and Spatial Information", *Pattern Recognition*, 18(3/4), pp. 257-269, 1995
- G. Strang, *Linear Algebra and Its Applications, 3rd Edition*, Harcourt Brace Jovanovich, Publishers, San Diego, 1988
- D. Terzopoulos, "Regularization of Inverse Visual Problems Involving Discontinuities", *IEEE Trans. PAMI*, 8, pp. 413-424, 1986
- D. Terzopoulos, and D. Metaxas, "Dynamic 3D Models with Local and Global Deformations: Deformable Superquadrics", *IEEE Trans. PAMI*, 13(7), July, 1991
- J. S. Weszka, C. R. Dyer, and Z. Rosenfeld, "A Comparative Study of Texture Measures for Terrain Classification", *IEEE Trans. Systems, Man, and Cybernetics*, 6(4), pp. 269-285, April, 1976

- A. Whitney, "A Direct Method of Non-Parametric Measurement Selection", *IEEE Trans. Computers*, 20, pp. 1100-1103, 1971
- R. Wilson, and M. Spann, "Finite Prolate Spheroidal Sequencies and their Applications: Image Feature Description and Segmentation", *IEEE Trans. PAMI*, 10, pp. 193-203, 1988
- R. Wilson, and M. Spann, *Image Segmentation and Uncertainty*, Letchworth, Research Studies Press Ltd., 1988
- J. W. Woods, "Two-dimensional Discrete Markov Random Fields", *IEEE Trans. Information Theory*, 18, pp. 232-240, Mar. 1972
- D. H. Wolpert, "Stacked Generalization", *Neural Networks*, 5, pp. 241-259, 1992
- J. Zhang and J.W. Modestino, "A Model-Fitting Approach to Cluster Validation with Application to Stochastic Model-Based Image Segmentation," *IEEE Trans. PAMI*, 12(10), pp. 1009-1017, Oct. 1990
- J. Zhang, J.W. Modestino, and D.A. Langan, "Maximum-Likelihood Parameter Estimation for Unsupervised Stochastic Model-Based Image Segmentation, *IEEE Trans. Image Processing*, 3(4), pp. 404-420, July, 1994