A STUDY OF MICROSCOPIC IMAGES OF HUMAN BREAST DISEASE USING COMPETITIVE NEURAL NETWORKS

by

Randy Allan, MD, FRCP(C)

A Thesis
Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

Thesis Advisor: W. Kinsner, Ph.D., P.Eng.

0-612-76743-4

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES
*****
COPYRIGHT PERMISSION PAGE

A STUDY OF MICROSCOPIC IMAGES OF HUMAN BREAST DISEASE USING COMPETITIVE NEURAL NETWORKS

BY

RANDY ALLAN

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University

of Manitoba in partial fulfillment of the requirements of the degree

of

Master of Science

A STUDY OF MICROSCOPIC IMAGES OF HUMAN BREAST DISEASE USING COMPETITIVE NEURAL NETWORKS

by

Randy Allan, MD, FRCP(C)

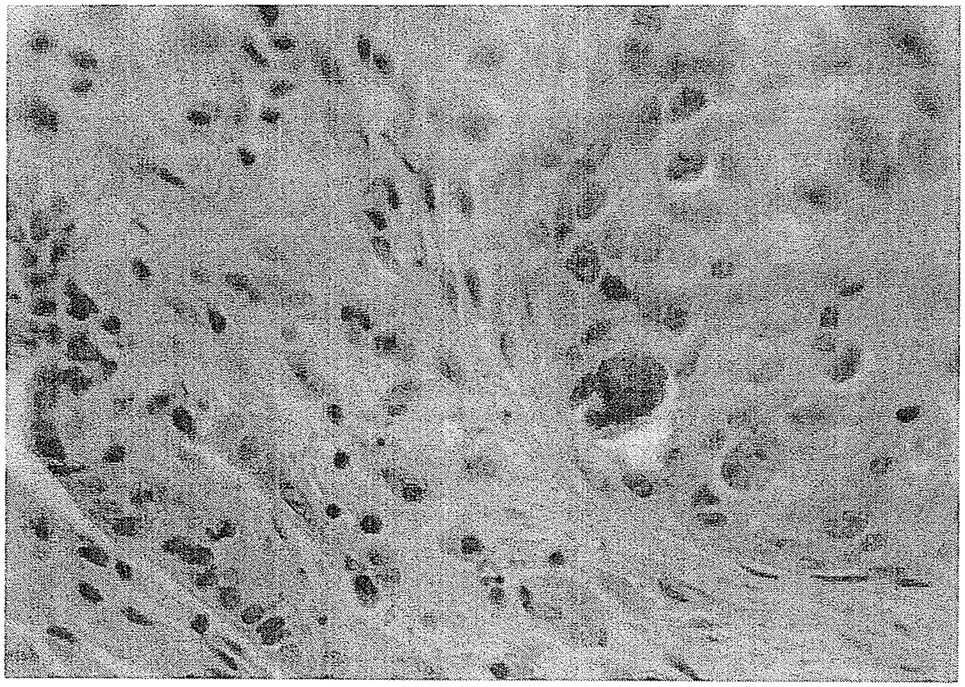
A Thesis
Submitted to the Faculty of Graduate Studies
In Partial Fulfillment of the Requirements
For the Degree of

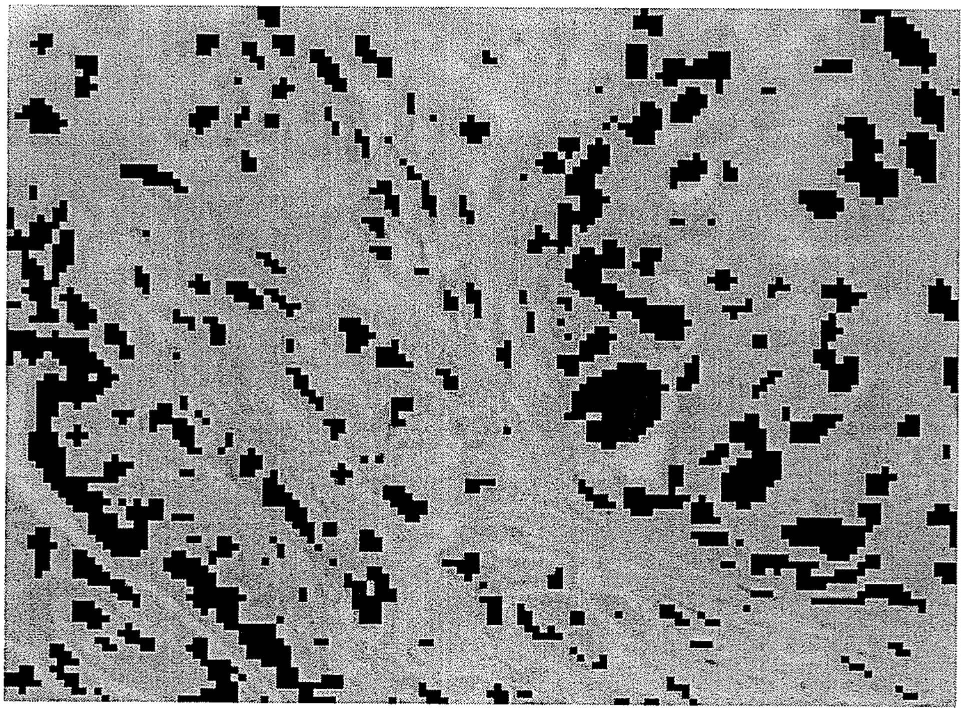
MASTER OF SCIENCE

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, Canada

Thesis Advisor: W. Kinsner, Ph.D., P.Eng.

(x + 92 + A31 + B6) = 139 pp.





ABSTRACT

Competitive neural networks offer a unique opportunity to extract features from medical images objectively. The advantages to this approach include automated or semi-automated analysis of medical images. Such an automation leads to improved precision and accuracy of diagnostic interpretation, while semi-automation achieves much the same goal with the help of a physician and could serve as a natural stepping-stone to full automation.

This thesis shows that competitive neural networks can extract general features from digital photomicroscope images of four types of human breast disease (fibrocystic change, fibroadenoma, infiltrating duct carcinoma and infiltrating lobular carcinoma). The competitive neural networks employed in this thesis include (i) the basic competitive neural network with conscience, (ii) one-, two- and three-dimensional self-organizing feature maps, and (iii) learned vector quantitizer. The networks were trained with training sets taken from digital images of representative areas of breast disease in microscopic sections. The vectors varied from four to as many as 10000 units in length. The pixel blocks were always square, containing from 2x2 up to 100 x 100 pixels. The number of features extracted were as few as four to as many as 72. The training time varied from as little as 50 epochs to as many as 2,000,000. (approximately two months time on the PC). Each network was compared to the others for performance, as well as to the performance of a pathologist.

Looking at the final weights of the networks, the weights showed average levels of gray scale values that corresponded to thresholding at low (dark) gray scale values, intermediate values and high (light) values, as well as gradients of values that corresponded to various edges. These features were extracted regardless of supervision or lack of supervision of the neural networks. To visual inspection, there are no obvious and sharp distinctions between benign and malignant diagnoses although the graphs of the patterns of feature distribution show clear differences between some diseases.

ACKNOWLEDGEMENTS

First and foremost I would like to thank Dr. Witold Kinsner for his preceptorship and guidance during the many years of course work and thesis work and for suggesting the use of competitive and hierarchical neural networks in this research. The magnitude of his patience, understanding and guidance cannot be understated.

I am indebted to Dr. David Scuse of the Department of Computer Science and Dr. Howard Card and Prof. Waldemar Lehn of the Department of Electrical and Computer Engineering for their helpful comments. Dr. Scuse was particularly helpful with the use of materials from his laboratory.

I would like to thank the members of the Department of Cytopathology of the Health Sciences Centre for the long-term loan of the Zeiss Axioskop microscope.

Fellow graduate students helped me many times along the way, and I express my gratitude to Ken Ferens, Armein Langi, Jason Toonstra and Richard Dansereau, as well as others who have since moved on.

Finally I would like to thank my friends and my children, all of who supported me fully as we came through this endeavor together.

TABLE OF CONTENTS

Page

Chapter I: Introduction

Chapter II: Literature Review of Pathology Image Analysis

Chapter III: Image Information from Microscopic Images

Chapter IV: Neural Networks and Classification of Images

Chapter V: Design of Experiments

Chapter VI: Experimental Results and Discussion

Chapter VII: Conclusions and Recommendations

References

Appendix A: Matlab Code for the Thesis

Appendix B: Additional Images of Feature Extraction

List of Figures

Page

Tables

List of Abbreviations

ANN	Artificial neural networks
BP	Backpropagation
CAS	Cellular analysis system
CCD	Charge-coupled device
CNN	Competitive neural networks
DNA	Deoxyribonucleic acid
FA	Fibroadenoma
FC	Fibrocystic change
HNN	Hierarchical neural network
IDC	Infiltrating duct carcinoma
ILC	Infiltrating lobular carcinoma
JPEG	Joint Photographers Expert Group
LVQ	Learned vector quantization
NN	Neural network
RGB	Red-green-blue
SOFM	Self-organizing feature map
TIFF	Tagged Information File Format

CHAPTER I
INTRODUCTION

1.1 Motivation

The motivation for this thesis comes from the desire to apply a search for objective features in the currently subjective task of pattern recognition in medical image interpretation. In short, diagnostic errors will be reduced by automation of microscopic slide interpretation. A potential additional benefit is improved productivity and efficiency for the histology laboratory.

1.2 Problem Definition

The problem is the irreducible error rate that comes from the current state of practice of tissue diagnosis. The problem can be seen as one of the limitation of human construction for handling this kind of problem, and in two parts. First, the human eye can only discern about 60 gray levels, whereas the electronic standard is 256, corresponding to 8 bits. Secondly, the human brain cannot carry out the complex mathematical image analysis systematically that can be done electronically. The challenge is to reduce this error rate significantly and efficiently through a form of automation.

1.3 Background

The different fields of medical image interpretation bring as much objectivity to this task as is humanly possible. Initially important features and patterns seen in tissue pathology are discerned in a critical research setting, and presented to the professional community as a peer-reviewed publication. Having been established as “the truth” in this way, the ensuing task is for tissue pathologists around the globe, as varied as they are in training, temperament and experience, to apply these findings to the daily work load of tissue diagnosis in a reproducible manner that is faithful to the canon received from the peer-approved research process. In this entire process, there are many opportunities for errors to occur that result in misdiagnosis, a deviation from the objective truth, at the bedside.

The current state of practice of histodiagnosis is for one pathologist to examine slides of tissue and render a written diagnosis. Another pathologist may or may not review the accuracy of the first pathologist’s interpretation, either in real time or at a later date. The referring clinician may or may not question the diagnosis and ask for a review of some kind. This essentially summarizes the current state of quality assurance in histopathology.

One issue that is not generally discussed in histopathology quality assurance, but is germane to this thesis, is that errors can be made at the time of the research that describes how to diagnose a particular disease. This research process calls for good cognition on the part of the researcher. The features of a disease should be thoroughly understood by him and then these impressions conveyed accurately and clearly into the literature. Critiquing this kind of research can be very difficult if it is one opinion against another with no other more objective method available. In this thesis we pursue objective microscopic feature extraction that has statistical diagnostic significance.

The idea that there are unique *measured* features in representative images of tissue diagnosis was something accidentally chanced upon by pathologists applying digital image analysis. Pixel patterns and arrangements, hence *electronic* features, were found to correlate with benign or malignant diagnoses [HGBS93] These features are usually not discernible as anything in particular to the human eye. Such features are often called *malignancy-associated changes.*

Competitive learning, especially unsupervised competitive learning, offers a remarkable tool with which it is possible to investigate the idea that there are unique and reproducible features that can be extracted by an electronic instrument, even if these features are not appreciated by the human eye. The cold hard discipline of mathematics can be brought to this task with minimal to no intervening prejudice on the part of a human interpreter.

This thesis is the first attempt to discover so-called malignancy-associated changes using a competitive neural network.

The Kohonen learning law [Hayk94], as implemented in an artificial neural network offers a ready-made and flexible tool with which to carry out this task. Essentially, the investigator employing this approach must be prepared to start over when it comes to thinking about image features and their role in pattern classification. A mathematical algorithm, in the form of the learning law, must develop independently *de novo* its own diagnostic approach to the medical image. The investigator must let the neural network operate on its own terms as much as possible in order to minimize the incorporation of human subjectivity. The process is stopped periodically to assess the progress of the neural network by relating its discoveries back to the original images, and then it is allowed to continue.

1.4 Thesis Statements and Objectives

The central objective of this thesis is the *qualitative* study of the ability of competitive neural networks to extract features from captured electronic images of diseased human breast tissue.

This is a contribution to a long-standing and ambitious project of the biomedical community of professionals: the ultimate usurpation of tissue diagnosis from humans to electronic means. It hypothesizes that if this goal is to be achieved then the process must be: (1) a rigorously scientific and mathematical one that allows artificial intelligence to operate on its own terms rather than as a mere adaptation of human ones, (2) one that satisfies medical professionals that artificial intelligence is approaching the problem correctly, as well as performing a satisfactory number of correct classifications, and (3) one of discovery so that the computer can reveal to the pathologist image features that it discovers that may be novel and useful, while tissue diagnosis is still a human responsibility.

Competitive neural networks (CNNs) are the experimental platform of this thesis. CNN have several attractive attributes that lend themselves to digital image analysis of medical images. They offer an opportunity to observe a form of artificial intelligence that can study microscopic tissue images independently. Supervision and guidance of the networks by a tissue expert can be varied from absent to intense. A great advantage is the fact the neural network can identify a feature objectively. Medical history does record more than a few instances of errors in the description of certain diseases (e.g., *Helicobacter pylori*) and disagreements about some entities lingers on (e.g., *dysplastic nevus*).

CNNs offer the opportunity to apply Beer's law to instrumentation and anatomical images. This is novel. Until now Beer's law has mostly been seen as the use of light to measure the concentration of a substance in a confined space. Measuring *multiple* confined spaces, in a mathematical relationship with each other, is relatively new in laboratory medicine. This is necessary because an image is a two-dimensional function.

Most peculiarly, CNNs can define features in ways that may be alien to the human observer. Within the pixel block (that the network is told to use as a frame for feature extraction) the pattern of pixels may not be recognizable to the human expert. Yet artificial intelligence may perceive such a pattern as malignant or premalignant change.

Microscopic images (i.e., images obtained through a microscope) are presented to competitive neural networks. Most of these networks are unsupervised. The ability of these networks to identify significant features in the images, and the effect of increased training on this ability, is appraised qualitatively from the standpoint of a trained pathologist. In addition, one type of CNN, the supervised Learned Vector Quantization (LVQ) network, is quantitatively appraised for its ability to classify. The ability of this type of network to accept supervision raises the possibility of a much more powerful network and one with a higher probability of finding distinguishing diagnostic features. Although considered, the use of hierarchical neural networks (HNN) to test the validity of the extracted features or develop a classifier of images is not broached due to time limitation.

1.5 Thesis Organization

Chapter II is a general exposition of the literature of the work that has gone on before. It proceeds historically, leading from the tree trunk of quantitative microscopy to the new branch of feature extraction with CNNs.

Chapter III discusses the issue of image information. As a clinical and general pathologist, the author takes the point of view that this anatomical pathology problem can be addressed as a clinical pathology problem by extending Beer's law to two-dimensions.

Chapter IV discusses CNN as a tool for feature extraction. Their practical utilities, and theoretical limitations, are discussed.

Chapter V presents the scientific method here. The CNNs extract features, and then the results are presented visually.

Chapter VII discusses the results. The issues are what the experiments were able to accomplish, and what objectives they fell short of.

Chapter VIII concludes the thesis with the author's observations on how the CNNs discover features, and how this knowledge can be used to design further experiments to make a clinically-relevant CNN-based system. Practical recommendations for extending this research are made.

CHAPTER II
LITERATURE REVIEW OF PATHOLOGY IMAGE ANALYSIS

2.1 Introduction

There are three important general themes in the literature that are germane to this thesis. First of all, that microscopic images can be quantified, usually with digital image processing. Secondly, that interest in the use of neural networks (NN) by the medical community has increased, commensurate with the growth of this field directed by neurocomputing professionals. Finally, that there is a great interest in developing products to meet real and perceived automated image processing needs, culminating on September 3, 1998, with the US Federal Food and Drug Administration announcing official approval of the *Autopap* cervical screening instrument for *primary* cervicovaginal smear screening. The latter is important to this thesis because the *Autopap* uses artificial neural networks (ANN). Considering these themes together, one gets an idea of the forces that carry the field of quantitative microscopy along, and this thesis with it.

The next three sections review, in order, the general history of quantitative microscopy up until the introduction of NN into this field (around 1990), a general history of NN, and finally the work on the application of NN to quantitative histology and cytology.

2.2 A Brief History of Quantitative Microscopy

Although the microscope was invented in the 17th century, it was not until the early 20th century that quantization of microscopic biological images was attempted. Jacobj [Jaco25] studied the relationship of cell and nuclear sizes of normal tissues. It was in the 1920s that the idea of quantitative microscopic pathology was born. Ehrich [Ehri36] performed similar studies but did so on normal and neoplastic human cells. Caspersson [CaSa42] did extensive and rigorous studies using fluorescence, but took a week to complete all measurements on a single cell. Needless to say the methods employed were all manual.

It was Mellors who first conceived the idea of an automatic scanning microscope for diagnostic purposes [MeSi51]. The 1950s were a time of exploration of electronic analysis of cells and particularly their DNA content, although not with the application of the modern computer. Mellors, working with the father of cytopathology George Papanicolaou and others, described the principles of quantitative analysis of cells in cervical smears and in pleural fluids [MeGP54]. From Mellor's ideas came the Cytoanalyzer, the first semiautomated analyzer for the detection of cancer cells in cervical smears [Toll55].

The Cytoanalyzer, and its follow-up model Cytoanalyzer II, are important in the course of the development of quantitative image analysis. Their early development, so soon after the establishment of screening cervical cytology, reflects the driving compulsion to automate this branch of pathology. The "Holy Grail" of cervical screening automation has affected progress in all aspects of the field ever since. There has been a heavy emphasis on the analysis of individual cells. Segmentation of nuclei and cytoplasm from the background and each other was a central issue of the 1960s and 1970s and continues as such to this day. This problem defeated the Cytoanalyzers and led to abandonment of this instrument [BaRC92].

In an important article from the mid-1960s, Prewitt (of the Prewitt edge detection algorithm) discussed the application of the science of digital image analysis to cell images [PrMe66]. Interest was greatest not so much in the edge detection algorithm as the use of thresholding, with the pixel histogram, to segment images.

An important early instrument that came out shortly after Prewitt's paper was the Taxonomic-Intracellular Analytic System (TICAS) [WBBO68]. It was intended as a research tool and used a LINC-8 computer. This instrument gave numerous cellular measurements and saw service up until the early 1980s.

As mentioned above, the 1970s initiated intense research into segmentation of microscopic images. Generally speaking, various groups developed algorithms centred around basic digital image processing techniques. Watanabe used a combination of density thresholding and gradient approach [Wata76]. Pycock and Taylor used contour tracing at a grey-level threshold determined by syntactic analysis of a histogram of grey levels [PyTa80]. Aggarwal and Bacus used a multi-spectral approach for image segmentation. They employed a two-dimensional histogram of optical densities of two pictures of the same scene but at different wavelengths [AgBa77]. Garbay *et al.* used a region-growing algorithm that combined local colour similarity and global shape criteria [GaCB86]. Work on segmentation continues to this day. For example, an algorithm for using a two-dimensional histogram based on gray-level threshold and nuclear texture to segment the nuclei of breast aspirate material has recently been described [AlKi96].

Most research groups at this time pursued a policy of developing feature extraction algorithms once the image and its scenes had been segmented. Usually this meant isolation of individual nuclei and then performing a panel of measurements upon each. The general theme was (and still often is) one of trying to emulate features as appreciated by trained pathologists. As this field of research has matured, features can be broadly categorized into densitometric, morphometric and texture. Some features have gained more popularity than others; some have been replaced by others found to be similar in quality but more successful in classification.

Classification methods revolved around classifying cells and classifying the specimens they are derived from. The methods utilized for cell classification in what could be called the pre-neural network era included Bayesian classifiers [OlPT77], linear discriminant analysis [Holm78], piece-wise linear discriminant functions [PyTa80], polynomial discriminant functions [RESB79], and hierarchical classifiers [Zahn79].

Classification at the specimen level has always been more of a philosophical issue built around the intent and capabilities of the system under development. Rendering specific diagnoses would be an ideal but is difficult to achieve. Most systems have taken a lower key and more practical approach, desiring only to use the statistics of the classified cell population to set-up a cut-off point for further human analysis (*i.e.*, screening test only).

By the late 1970s and early 1980s, several microscopic image analysis systems had been developed. Noteworthy examples (of which there are many) are the BioPEPR, intended for large-scale cancer screening programs and developed in the Netherlands [Zahn79], CERVIFIP, a multi-purpose image analysis system from Edinburgh [TuHu80], CYBEST, a Japanese image analysis system aimed at uterine cancer detection [Muka83], and the SAMBA, a system designed in France for low-resolution cell localization and cell aggregate analysis, and high resolution cell feature analysis [Bruga79].

2.3 A Brief Background of Neural Networks

From the development of the first useful electronic computer (ENIAC) in 1946 up until the late 1980s, essentially all information processing applications used a single approach: Procedural (programmed) computing. This involves the solution of a problem with a specific algorithm and a known and finite set of rules. This has the advantage of reliable and precise solutions. However, it has the disadvantages of costly and time consuming development, and the possibility of no programmed solution being possible. An artificial intelligence approach, where the electronic instrument had some heuristic abilities such as generalization, always seems to have been desirable. Still, the world has had to wait for several decades for neurocomputing to mature to a point where it could solve problems.

The seminal event in the development of NN was the description of the McCulloch-Pitts binary neuron [McPi43]. Taking inspiration from the natural neurological system as a means of developing artificial intelligence, and looking for a mathematical and hence algorithmic explanation of its fundamental building block, the neuron, McCulloch and Pitts developed a mathematical model of how a neuron works and described how a neural network could be designed and used to solve mathematical problems.

In 1949, Hebb, a Canadian psychologist, described a learning law for the synapses of neurons based on re-enforcement principles [Hebb49]. Although not formulated with neural networks or even computers in mind, it provided a mathematical concept that made it the first learning law applied to neurocomputing. Shortly after, in 1951, Marvin Minsky constructed the first neurocomputer [Mins54].

Rosenblatt and others developed a more successful neurocomputer in 1957 [Rose58]. His Mark I Perceptron neurocomputer was the first implementation of the perceptron, one of the first successful NN. It was intended for pattern recognition work and was able to manage linearly separable problems. It was also subject to supervised learning and had its own, now abandoned, learning law.

In the 1960s, there was much enthusiasm for neurocomputing but this ground to an almost-complete halt by the end of the decade [Hech90]. One reason was a failure to generate new ideas and algorithms to drive the science forward and make it capable of tackling more difficult problems. But perhaps more importantly the publication of the book *Perceptrons* by Minsky and Papert, discrediting neural computing, was of devastating impact. Although there seems to have been a mission on the part of Minsky (he became apostate to a discipline he had helped found) to strangle the infant science, the scientific argument against it centred on Minsky's mathematical proof that neurocomputing could not solve non-linearly separable problems [MiPa69].

The period from the late 1960s to the early 1980s have been called quiet ones for neural network research. The 1980s was a second renaissance for the field. Rumelhart and others refined the backpropagation (BP) algorithm to solve non-linear problems, and made it practical [RuMc86], following the original work by Werbos [Werb74].

Hopfield generally promoted the science of neurocomputing and developed a type of neural network named after him [Hopf84]. But most germane to this thesis, practical competitive learning was realized in a neural network model in the 1980s.

The concept of competitive learning antedates 1962 and can be found in the literature under the rubric "unsupervised clustering". Stark and others described a competitive, unsupervised learning law in 1962 [StOW62]. Grossberg, a prolific NN researcher, described the fundamental NN learning law that eventually evolved into the one used by the networks in this thesis and most other competitive neurocomputing applications [Gros78]. Influenced by Grossberg, Von der Malsburg developed a self-organization and competitive learning law that was non-local, and kept the sum of all weights constant [Vond73].

Von der Malsburg's work was extended by Grossberg to eliminate the non-local character of the law [Gros76]. But it was Kohonen, with his theory of equiprobable representatives of a fixed probability density function, who essentially created the current competitive learning law and algorithm that is the heart of most such neural networks, and bears his name [Koho84]. This learning law became practical and useful when Desieno introduced the conscience function to ensure equiprobability [Desi88].

2.4 Neurocomputing in Microscopic Digital Image Analysis

The late 1980s and 1990s was a period of discovery of neural computing by the pathology image analysis community. The introductory articles by Dytch, Wied and Bibbo of the late 1980s seemed to initiate general interest in researching this area [DyWB88] [DyWi90a] [DyWi90b] although Bartels, the doyen of quantitative microscopic pathology, mentioned neurocomputing in passing earlier in a published review of machine learning [BaWD88]. Examining Bartels' article, it is clear that, for computerized classification, interest at that time was strongest in rules-based and expert systems.

It is clear from reading the literature and studying events of the time that clandestine neural network research in the commercial field was very active, and successful. In 1990, the PAPNET cervical screening instrument was announced to the world. Using two BPNN to analyze low resolution and high-resolution scenes, it represents a remarkably short development period from description of the refined BP algorithm to its application in an engineered product. The details of the use of neural networks by PAPNET remain proprietary to this day. Unfortunately, the developer of the PAPNET, Neuromedical Systems of Suffern, NY, went bankrupt. Its intellectual property was purchased and shelved by a competitor.

In the early 1990s, and arguably even to this day, most interest has centred on the use of BP. Its attractions to the medical profession are: (i) as a classifier (where other classifiers have failed or proven unreliable), and (ii) as a predictor/prognosticator. One of the earliest applications in laboratory medicine was a study by Astion and Wilding where they used nine variables to predict the presence or absence of breast cancer in patients with breast disease [AsWi92]. Strangely, the nine variables they chose were either weakly related to breast cancer or not related at all. The NN outperformed a coin flip. The intent of Astion's work was to demonstrate the predictive power of the BP algorithm.

Radiology and histopathology, the two main imaging branches of medicine, have taken great interest in neural networks in the 1990s. Becker at the Armed Forces Institute of Pathology has led much work in this field. In 1992, with O'Leary and Mikel, he used classical feature analysis with a commercial image analysis system and a three layer BPNN to separate tubular carcinoma from sclerosing adenosis, with a 92% success rate [OlMB92]. Dawson saw the potential for combining information from the Cellular Analysis System 100 (CAS 100), a popular image analysis system of the time, with a three layer BPNN developed on Brainmaker, a popular BP neural software package of the time. She was able to get the network to emulate human grading of breast cancer nuclei with no more than 70% agreement [DaAW91]. Wolberg and Mangasarian, part of a group of non-pathologists doing extensive work into image analysis of breast aspirates at the University of Wisconsin, used BPNN. They studied nine features graded from one to ten that were fed into their network [WoMa90]. The network only misclassified 2 of 70 aspirates in cross-validation. Still, this group appears to have ceased using NN, presumably because Mangasarian developed his own classifier based on a linear programming approach [Mang68]. Palcic compared a BPNN to discriminant function analysis and a decision tree classifier [PMST92]. He used a large collection of rehydrated cervical cells from the files of the British Columbia Cancer Agency. The nuclei were segmented using their own algorithm ([MaPa89a] and [MaPa89b]) and then measured using the CAS 100 image analysis instrument. Fifty-seven features were extracted from 4686 nuclei. Each classifier was challenged to classify each nucleus according to the now-defunct squamous dysplasia system as well as a modification that is akin to the now-accepted Bethesda system. The results were roughly comparable for each classifier. The NN was able to distinguish 99.4% of all high-grade squamous intraepithelial lesions from others.

Many publications have revolved around reporting the success of using the PAPNET cervical smear screening instrument [Sher94], [Mang93], [Mang93], [BoKo93], [BoKo94], [Rose93], [KhKO93], [KiSh92]. With the apparent demise of this instrument and no description of the algorithms used, these articles do not contribute much to our understanding of the general science of image analysis with NN.

Some sophistication about NN beyond simple straight application of feed-forward networks is beginning to form in the medical community. *Hierarchical neural networks (HNN)* are one such example. They could be defined through the use of two or more distinct classes of neural networks organized in sequence to augment classification [LKKK94]. A common example is the use of a competitive neural network (CNN), such as the self-organizing feature map neural network (SOFM), as the first component of the hierarchy, with its output presented to a feed-forward neural network, such as BP, to complete classification.

Mehdi used hierarchical specialist NN to improve upon the classification of the cervical nuclei training set of Palcic (see above) [MeSH94]. He used specialist NN to classify according to the Bethesda system and achieved an insignificant improvement over Palcic's networks. He also had greater classification accuracy for nuclei when using the squamous dysplasia system as a target. His positive conclusion of matching the performance of trained histotechnologists must be looked at critically from our current vantagepoint. He attempted to emulate a classification system that has always had poor precision and has been abandoned for that reason. The performance of the specialist NN probably reflects the shortcomings of this system, not the networks. His hidden layer is very large (thirty nodes) and may have performed better than smaller networks because it memorized patterns. Finally, as the aim is to improve upon, not match human performance and error rate, one must be cautious of his overall system. However, his claim to speedier performance of the HNN over a solitary network is an important contribution, although a different panel of image features (particularly a smaller one) could speed up a solitary network without necessarily sacrificing accuracy.

Schnorrenberg created a modular NN that combined an unspecified feed-forward NN with a receptive field squashing function to locate immunochemically stained breast nuclei on a frozen section slide [ScTP00]. Pixel blocks were reduced to their single value decomposition so that the feed-forward network could handle the image data.

2.5 Competitive Neural Networks in Microscopic Image Analysis

The medical image analysis community, or at least the pathology image analysis community, seems unduly focused on the BPNN, or it is at least fixated on NN as classifiers and predictors. There is a tendency among articles published by leading researchers to describe NN in general terms and confine themselves to using the BP feed-forward network as the archetypal example. In the introductory article by Dytch [DyWi90a] and a similar up-date by Becker four years later [Beck94], SOFM and the contribution of Kohonen are only mentioned in passing. Nonetheless, some articles are appearing that applying these networks to the medical imaging challenge.

Karakitsos used Learned Vector Quantization (LVQ) as a feed-forward network to classify thyroid needle aspirates [Kara99]. The data presented to this network were classical features extracted using the Optimas commercial image analysis system. Twenty-five measurements of classical cellular features were formed into a training vector for each cell. Standard deviation and mean value for each patient were calculated from these so that a training vector of 50 values could be added to a training set. The 198 patients formed a training set of 198 vectors, and the network was trained for 50 to 200 epochs. The program package used was the one developed by Kohonen himself with the LVQ1, LVQ2 and LVQ3 algorithms. A two-class network (specialist in the sense that it discriminates benign from malignant) and a four-class network (four different disease categories) were created. The overall accuracy of the two-class network was 97.7%.

Dhawan used the concepts of competitive learning and SOFM to enhance image segmentation, and used photographs of cutaneous lesions to demonstrate this, but he did not use NN per se [DhAr93].

2.6 Summary

I have presented a historical overview of digital image processing of microscopic tissue images from the perspective of our thesis topic. Thus, I have followed tried-and-true paths in this field, branching off into newer and less-proven technologies as we followed the time line. Finally, I have reached a point where the author must break his own trail, which is the application of competitive learning, through neural networks, to digital image analysis. An approach to the problem of microscopic digital analysis, its tie-in to the highly instrumentalized area of clinical pathology, and its significance for tissue diagnosis, will be discussed next.

CHAPTER III
IMAGE INFORMATION FROM MICROSCOPIC IMAGES

3.1 Introduction

The first engineering issue to be addressed is the application of sound optical principles to the acquisition of accurate and relevant data for further study.

The task of harvesting useful numerical training sets from microscopic images is complex and intimidating. For the instrument or the pathologist, rendering a correct diagnosis will often involve the analysis of several different microscopic fields at different magnifications. Multiple slides, sometimes from very different anatomical and tissue sites, must be studied. Even though only one tissue stain set is usually used (hematoxilyn and eosin) this still means a complex cognitive analysis of the absorption and reflectance by the stained tissue of all wavelengths of light.

3.2 Beer's Law

The task of sampling light for measurement, both by an instrument and by the human eye, can be seen as a form of application of Beer's law. Beer's law plays an important role in this thesis. It is the scientific principle where instrumentation for clinical pathology and microscopic anatomical pathology intersect. Beer's law states that the concentration of a substance is directly proportional to the amount of light absorbed, or inversely proportional to the logarithm of the light transmitted, and can be formulated as

$$\begin{aligned} A &= abc \\ &= \log_{10}(100/\%T) \\ &= 2-\log_{10}(\%T) \qquad (3.1) \end{aligned}$$

where A is the absorbance, a is the absorptivity, b is the light path of the solution in cm, c is the concentration of the substance of interest, and $\%T$ is the percent transmittance, and $\log_{10}(\bullet)$ is the base-10 logarithm.

The current preferred electronic method of image capture is the charge-coupled device (CCD). This electronic chip is covered with a two-dimensional array of wells (sometimes referred to as pixels) embedded in the silicon. Each well captures all incident light landing upon it. The amount of light captured over a time interval is measured simultaneously in all wells and set as the gray scale. The saturation point of the wells is set to 255 (corresponding to the maximum range of 8 bits) and the equivalent of the human glare limit. In some cameras three chips are used, each corresponding to one of the three parts of the red-green-blue colour system. The incident light is split using a three-way prism prior to entering a well.

Each well of the CCD can be thought of as a cuvette. With a chemical analyzer, the concentration of a particular chemical in a quantity of bodily fluid is measured. This is the most common application of Beer's law. But this is a point process, with no dimensionality. Image sampling is a two-dimensional extension of Beer's law. The more light that is absorbed by the two stains at a particular point in the tissue, the lower the gray-scale value of the cell. This is how Beer's law is applied in the setting of image analysis.

Ideally, only one wavelength of light, as allowed by a narrow band-pass filter, is sampled. Where the intensity of the light is insufficient to excite the CCD, a wide-pass or long-pass filer is used.

3.3 The Nyquist Sampling Theorem

Nyquist's theorem governs the way that light transmitted by the glass slide of tissue should be sampled. The sampling theorem, stated for the usual one-dimensional signal processing case, is as follows: for an analog signal *x(t)* to be recovered completely from its sampled version *x(nT)* (thus preserving all information from the original signal), then the following relationship must hold

$$f_s = 1/T_s > 2f_{max} \qquad (3.2)$$

where f_s is the frequency of sampling, T_s is the sampling period, and f_{max} is the highest frequency component in the signal. The signal must be sampled at a rate greater than 2 samples per cycle of the highest frequency component.

In the image sampling case of this thesis, the maximal dimension of the section of each image sampled by its corresponding CCD well must be less than one-half of the shortest wavelength entering the well. The light sampled is that which is transmitted by the stained tissue. (The dye staining the tissue absorbs some light from the microscope lamp but reflects and transmits the rest). At a magnification of 400x, the diagonal of the sampled area of an image is approximately 240 nm with the experimental system used here. Consequently, a low-pass optical filter of 500 nm must be used to prevent undersampling of the shorter blue wavelengths and satify the Nyquist sampling requirement (3.2).

Using this approach, the captured image of the tissue on the glass slide is a two-dimensional array of gray-scale values of transmitted/reflected light that shows reasonable fidelity to the information present. Each captured image contains the amount of light energy absorbed by the stained tissue on the slide at a particular location, and organized into a two-dimensional relationship.

3.4 Image Noise

Image noise is usually a real problem in most digital image processing problems. Usually an image averaging process is applied to reduce image noise, or a median filter can be applied.

Most noise in microscopic image analysis is white ("salt-and-pepper") noise. Because the competitive neural networks (CNN) are performing a type of statistical averaging process, it was felt white noise would not interfere with a *visual* interpretation of the results in any meaningful way. A median filter was not used because it might reduce the already-low contrast in the image even further. Hence, no noise reduction technique was employed here. In a future work using a classifier, comparative results between methods with and without noise reduction should be used.

3.5 Image Preprocessing

Preprocessing is any manipulation of raw data prior to its presentation to a classifier. It can be applied in order to speed up data processing, focus the classifier on a particular part of a problem, and eliminate noise.

In this thesis preprocessing is essentially dispensed with. The concern here is for fidelity of the image data. The problem for the neural networks (NN) is the determination of possibly subtle image features. Any preprocessing could result in the loss of information that may be necessary for the neural network to perform its task.

3.6 File Format

A definitive discussion of the topic of image file formats is beyond the scope of this thesis, as there are hundreds of formats. This area is in a great deal of flux, even today. There are a great many image file formats, usually of commercial origins and hence often proprietary. Although most of the key formats are described completely, little information is available in the public domain about some proprietary file formats. However, a rudimentary overview of this topic as it relates to this thesis is requisite, as well as a discussion of the issues that are most contingent upon this thesis.

Image file formats can be categorized broadly and loosely into two types: *vector graphics,* and *raster* or *bitmap*. Vector graphic files store graphical information as mathematical representations of various shapes and geometric structures. They are useful for image creation. Although useful for graphical artists and animators for this reason, they have no place in the scientific endeavour of image measurement and will not be considered further.

Raster (bitmap) files store the information about each pixel in an image individually. Thus, each pixel serves as a cuvette to record and store the quantity of light energy that arrived at it, as an 8-bit gray scale value. With some raster file formats, this amounts to a two-dimensional array of cuvettes that measure light energy. Thus, they are suitable for image capture and many are suitable for image measurement.

For the purposes of this thesis we are interested in file formats that act in the simple fashion described above. Some raster files use mathematical algorithms to compress the image data to save storage or transmission bandwidth. These are often unsuitable for image measurement. One common example in the Joint Photographers Expert Group (JPEG) file format. This format processes the raw raster data first by using a discrete cosine transform, and then truncating the length of each irrational number from each pixel to no more than 100 decimal places. Often the number of decimal places is even fewer. This image compression results in loss of information. There are also other lossless compression techniques, as discussed next.

In this thesis we use the Tagged Information File Format (TIFF). This format, now widely used, is a mere fifteen years old. Aldus Corporation (now part of Adobe) developed it originally in conjunction with several other major corporations. It is thus in the public domain. There are several variations of this file format due to its complexity, but it can store the raw raster data without data loss. A header contains various information about the image and points to the image file directory(s), which contain more specific information. These in turn point to file locations that contain the raster information. The image file format is flexible and enhancible. We use a version from *Matlab* that stores each RGB channel separately.

3.7 Summary

Extraction of information from a microscopic image is a complex task that requires attention to detail and respect for laws of physics and information. It also requires knowledge of the limitations of the computational environment and especially of the file format. Dealing with colour also imposes additional challenges. There are currently no standards for microscopic image information extraction, although reading the literature does give an impression of certain preferred approaches.

The required Nyquist sampling theorem is used in acquiring microscopic images. The Beer's law is incorporated into the acquisition process, thus posing a histopathology problem as a clinical pathology one.

CHAPTER IV NEURAL NETWORKS AND CLASSIFICATION OF IMAGES

4.1 Overview

The experiments in this thesis are conducted on competitive neural networks (CNN), a subclass of neural networks (NN). A discussion of the background and fundamentals of neural computing is warranted to familiarize the reader with the tools used here.

Neurocomputing could be said to have arisen from the push-and-pull of two forces at work. The push force has its origins are in the McCulloch-Pitt binary neuron, a mathematical model of how natural neurons work developed in 1943 (coincidentally around the time of the development of the first computers) [Hayk94]. Scientists and engineers are often inclined to take inspiration from nature to spur on new developments. McCulloch and Pitts' mathematical model of the neuron serves as the starting point for the development of a distinct form of computing: *neurocomputing*.

Computing has been long dominated by procedural computing. This is the computational application of an algorithm to the solution of a problem. Its meticulous nature is a strength and a weakness. The weakness is the brittleness that comes with an algorithm. Some problems are best approached with flexible solutions that adapt to the problem through training.

The NN is an adaptive computer structure. Generally, neurocomputing is the technological discipline concerned with parallel distributed adaptive information processing systems that develop information processing capabilities in response to exposure to an information environment [Hech90]. A definition of a NN is:

> "A parallel distributed information processing structure consisting of processing elements interconnected via unidirectional signal channels called connections. Each processing element has a single output connection that branches into as many collateral connections as desired. Each carries the same signal. The processing element output signal can be of any desired mathematical type. The information processing that goes on within each processing element can be defined arbitrarily with the restriction that it must be completely local…" [Hech90].

There are many different types of NN. It is beyond the scope of this thesis to discuss them all. Common features of most NN are: processing elements or nodes, connections, local memory, transfer functions, and input signals consisting of statistically significant training sets. The training set is presented one vector at a time to a processing layer through connections governed by a transfer function, which in turn is subject to a learning law. There may be another transfer to an output layer and often is if the NN is to serve as a classifier.

4.2 Learning Laws

There are many different learning laws and some are very popular and useful, others have fallen into disuse. Learning may be supervised or unsupervised. Supervised learning occurs when the expected output is given to the NN, while unsupervised learning occurs when it is not given. Unsupervised learning usually employs a competitive form of learning, where the nodes of the NN compete to be activated, as described next.

4.3 Kohonen Learning Law and Competitive Neural Networks

One of the NN used in this thesis is a CNN that learn using the Desieno modification of the *Kohonen Learning Law*. This is the focal point in this thesis.

The basic structure of the Kohonen layer is shown in Fig. 1. The layer consists of N processing elements, each receiving n input signals x_1,x_2 ,...x_n.

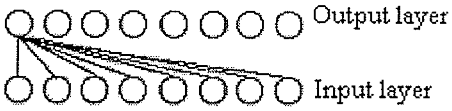


Fig. 1 Competitive neural network – first connection shown only.

Each Kohonen processing element calculates its input intensity I_i in accordance with the following formula

$$I_i = D(\mathbf{w}_i, \mathbf{x}) \tag{4.1}$$

where $\mathbf{w}_i = (w_{i1}, w_{i2}, \ldots, w_{in})^T$ is the weight vector, $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the input vector and D(•,•) is the distance measurement function.

Once each Kohonen unit has calculated its input intensity I_i, a competition takes place to see which unit has the smallest input intensity, and is hence the winner of the competition. Its output z_i is set to 1, all other units are set to zero.

At this point, the Kohonen learning law is applied. The Kohonen learning law can be stated as

$$w_i(n+1) = w_i(n) + \alpha(\mathbf{x} - w_i(n+1)\,)\, z_i \qquad (4.2)$$

where w is the weight vector of the network, x is the input signal vector, z is the output, $i = 1,2,\ldots, N$, n is the number of training iterations and α is a real constant greater than 0 and less than or equal to one. The weight vectors $\mathbf{w}_\mathrm{i}$ arrange themselves in the n-dimensional real space $\mathbf{R}^\mathrm{n}$ such that they are approximately equiprobable, in the nearest neighbour sense, with respect to **x** vectors drawn from $\mathbf{R}^\mathrm{n}$ in accordance with the probability function ρ. [Hech90]. Many researchers developed competitive learning laws but in his version Teuvo Kohonen emphasized that the derived vectors (the output of this type of network) should be equiprobable representatives of a fixed density function.

A problem with the original Kohonen law is that a few processing elements often won too many, or almost all, competitions. The downside of this is that significant features in low numbers might not have a representative vector. Duane Desieno's solution to this was to introduce a "conscience" that monitors the history of success of a processing element, forcing it to drop out of competition for a period of time if it won too much. The locally calculated value for each processing unit is given by

$$f(n+1)_i = fi(n) + \beta(z_i - f_i(n)) \tag{4.3}$$

where f is the fraction of time a processing element wins the competition, z is an output signal (zero or one), and β is a small positive number.

The new bias value is then recalculated using

$$b_i = \gamma(1/N - f_i) \tag{4.4}$$

where b is the bias value, γ is a positive constant (of about 10), N is the number of Kohonen processing units, and f is the fraction of time a processing element wins the competition.

At this point, unlike the usual situation in which the weight is updated in the one processing unit with $z_i = 1$, a separate competition is held to determine which processing element will have its weights adjusted. The competition is held on the basis of which processing element has the smallest value of

$$\mathrm{D}(\mathbf{w}_i, \mathbf{x}) - b_i \tag{4.5}$$

Ties are broken on the basis of smallest index. The processing element that wins this competition then updates its weights in accordance with the usual Kohonen learning law.

Kohonen's theory of equiprobability has a vital role in establishing the central idea of this thesis in that this type of neural network should be able to identify significant features, even if the number of representative vectors of these features in the training set is low. Without the conscience, the competitive network may be inclined to overemphasize a particular feature at the expense of all others. This is akin to a pathologist using nuclear hyperchromasia alone to diagnosis malignancy, without reference to any other features.

4.4 Limits of the Kohonen Learning Law

Limitations to the Kohonen learning law can be divided into the theoretical and the practical. From the theoretical vantagepoint, the Kohonen learning law is a subset of information theory and thus imperfect for the process of extracting all pertinent information while ignoring impertinent noise. Still, it is a straightforward algorithm that is easy to implement; thus it represents a potentially useful tool for extracting relevant features. It is an open question as to whether or not it can extract clinically useful information.

From a practical vantagepoint, the networks are prone to "twisting" and thus not organizing themselves in the most beneficial manner. The Kohonen "movie" of the twisting network that is trying to organize a two-dimensional array of random dots is well known. In the experiments in this thesis, the networks would not necessarily organize themselves from light to dark features in the most efficacious or obvious way. For instance, in the case of the three-dimensional network, the lightest features may be found in the centre on one side of the cube and the darkest features on an adjacent side, rather than on the opposite side. It is a matter of trial-and-error until experimental results come up that match the best possible organization. Guiding the networks by "seeding" locations in the networks with preferred target outcomes is to no avail, even with short training periods.

The features often do not match between two different networks conducting otherwise identical experiments on different diagnoses. For instance, one network might find its darkest thresholding on node #5, another on node # 62. The bearing this has on experimental outcomes is unknown.

4.5 Learned Vector Quantization (LVQ)

This algorithm can be considered an extension of the self-organizing feature map (SOFM) algorithm. In this algorithm, a linear classifier is applied as an additional layer to the competitive layer. This layer is subject to supervised learning. Unlike the SOFM, where the network is told to look for significant features without further guidance, the LVQ network is told the class from which the column vector of the training set comes from. Thus, the LVQ network can develop a representative vector of column vectors coming from a particular region of an input space through the application of the principle of Voronoi tessellation. If a Voronoi vector $\mathbf{w}_c$ is closest to an input vector $\mathbf{x}_i$ and they are of the same class, it is adjusted as follows

$$\mathbf{w}_c(n+1)= \mathbf{w}_c (n) + \alpha_n[\mathbf{x}_i - \mathbf{w}_c (n)] \tag{4.6}$$

where α_n is between zero and one, w is the weight vector of the network, $i = 1,2,\ldots, N$, n is the number of training iterations and x is the input signal.

If not of the same class, then

$$\mathbf{w}_c(n+1)= \mathbf{w}_c(n) - \alpha_n[\mathbf{x}_i - \mathbf{w}_c(n)] \tag{4.7}$$

The weights between the input layer and the competitive layer in this network can be extracted, converted to pixels block, and examined.

This network is practical and significant for two reasons to this thesis: firstly, it reduces the search effort for significant features. Although it was stated earlier that a hands-off approach by the human operator would best promote objectivity; this is not always the best approach. Most pathologists in most common settings will readily agree upon certain fundamental points about tissue morphology and its pathobiological significance (benign or malignant). By including this information in the learning by the network, the search for features should be made easier. Secondly, it introduces a classifier. The performance of the CNN in extracting important features can be evaluated directly by its ability to classify correctly images once the training set is resubmitted as a test set.

4.6 Summary

Neural networks have been overviewed with an emphasis on the CNN. The emphasis here is on objective feature extraction. There are different types of CNN each offering a special property for experimentation in this field. In the next chapter, each of the different CNNs will be presented with numeric information as training and verification sets made from pixel blocks extracted from the microscopic images of breast disease. The performance of each type can be evaluated over different lengths of training and compared to each other.

CHAPTER V
EXPERIMENT METHODOLOGY

5.1 Introduction

The intent of this work is to design experiments capable of searching a large range of the parameters of competitive neural networks (CNNs) in order to find image features characterized as pixel blocks. The range of parameters to be varied in all network types are the size (number of nodes) of a network, shape of the network, the number of epochs the network is to be trained, and the size of the pixel blocks to be sampled. The shape of the pixel blocks is kept constant (square).

There were six series of experiments: (i) one series involved a basic CNN, (ii) one each of the one-, two-, and three-dimensional neural networks, and (iii) two examples of learned vector quantizer(LVQ).

For the learned vector quantizers, one grouped benign and malignant diagnoses such that these were the two classes of interest, the other assigned class to each diagnosis.

5.2 Materials

Images were taken from archival material (glass slides) from the collection of the author. There were two glass slides from four different breast diseases each: infiltrating duct carcinoma, infiltrating lobular carcinoma, fibroadenoma and fibrocystic disease. The sections on the slides were removed at surgical biopsy, fixed in 10% aqueous buffered formalin, embedded in paraffin cut at 4 μm and stained with hematoxylin and eosin. They were judged typical and representative of their diagnosis. Fresh material and Feulgen-staining were not used due to limitation of resources and a desire to emulate the standard clinical approach of the pathology laboratory. Normal breast tissue was unavailable at the time of this research.

Breast disease was chosen as the subject of study because low power examination with the microscope is diagnostic and, therefore, it is easy to select representative regions of interest. High power examination of these representative regions precedes image capture of representative cells and their nuclei as a subregion of interest.

The neural networks were trained on these subregions of interest to discover statistically significant features.

5.3 Features

Features are defined in this thesis as statistically significant patterns of gray-scale pixels in a square block. The features are statistically significant because each of these pixel blocks is the visual manifestation (gray scale) of the weight vectors (normalized values) of the trained competitive NN. As described in Eq. 4.2, these weight vectors are distributed throughout $\mathbf{R}_n$ with a density proportional to the probability density function. These are not necessarily "roundness", "mean optical density", "kurtosis" or any other term used in the literature to describe what can be called *classical* features. Classical features are those visual features that the human eye recognizes readily and can describe to others subjectively (e.g., roundness, darkness, squareness and so on).

The initial size of the pixel blocks to be sampled is set at 4 x4. This size was selected because it was found useful by Langi *et al* [LFKK94]. Other sizes used were 10 x 10, 20 x20, 50 x50 and 100 x 100. A large size such as 100 x 100 was selected as the upper limit because the size of this pixel block captured the relationship between small groups of cells and was still computationally manageable.

5.4 Selection of Network Parameters

The initial number of network nodes chosen was 24 for the basic CNN, one-dimensional SOFM and LVQ. This would allow for distribution of the features of each of the four diagnoses amongst six nodes each *if* the features were all unique. As well, 24 is close to 25 (for a 5 x 5 two-dimensional SOFM) and close to 27 (for 3 x 3 x 3 three-dimensional SOFM). Thus, the SOFMs could be kept perfectly symmetrical and seek out about the same number of features. The codebooks for each SOFM would be roughly the same size. There was no way of knowing at the beginning how many nodes would be necessary. When initialized, the values of all weights were randomized.

A larger 5 x 5 x 3 three-dimensional SOFM was also created. The middle node of the "top layer" of 5 x 5 set of nodes for this NN had all of its weights set to zero. It was an absolute and uniform dark feature. The weights of the middle node of the opposite 5 x 5 set ("bottom layer") were all set to 255, making it represent an absolute and uniform white feature. The weights of a node in one corner of the 5 x 5 set with the absolute dark feature were also set to the same values as a representative pixel block taken from a nucleus of infiltrating duct carcinoma. The same was done for an adjacent corner node with infiltrating lobular carcinoma, the next corner node for fibroadenoma and the last corner node for fibrocystic change. The remainder of the weights were randomized.

As described above, the LVQ networks were of two type, one with two classes of benign and malignant, and one with four classes of all four diagnoses.

Networks were trained for 1,000, 5,000, 10,000, 50,000, 100,000, 200,000, 10^6 and 2 x 10^6 epochs.

For one-dimensional NNs (i.e., competitive, one-dimensional SOFM and LVQ), node #1 is set as the node to the left, counting up to node #24 on the right. The weights of the node in turn represent the feature corresponding to that number. For two-dimensional SOFMs, the nodes were designated 1 to 25 by reading right to left and top to bottom. For three-dimensional SOFMs, the nodes were designated 1 to 27 by reading left to right and then top to bottom for the first layer, then the second layer, then the third layer.

5.5 Image Acquisition

Twenty high-power microphotographs of four examples of human breast disease, infiltrating lobular carcinoma (ILC), infiltrating duct carcinoma (IDC), fibrocystic change (FC) and fibroadenoma (FA), are taken using a Zeiss Axioskop microscope and a Pixera 1260x960 charge-coupled device (CCD) camera (Figs. 2 and 3).

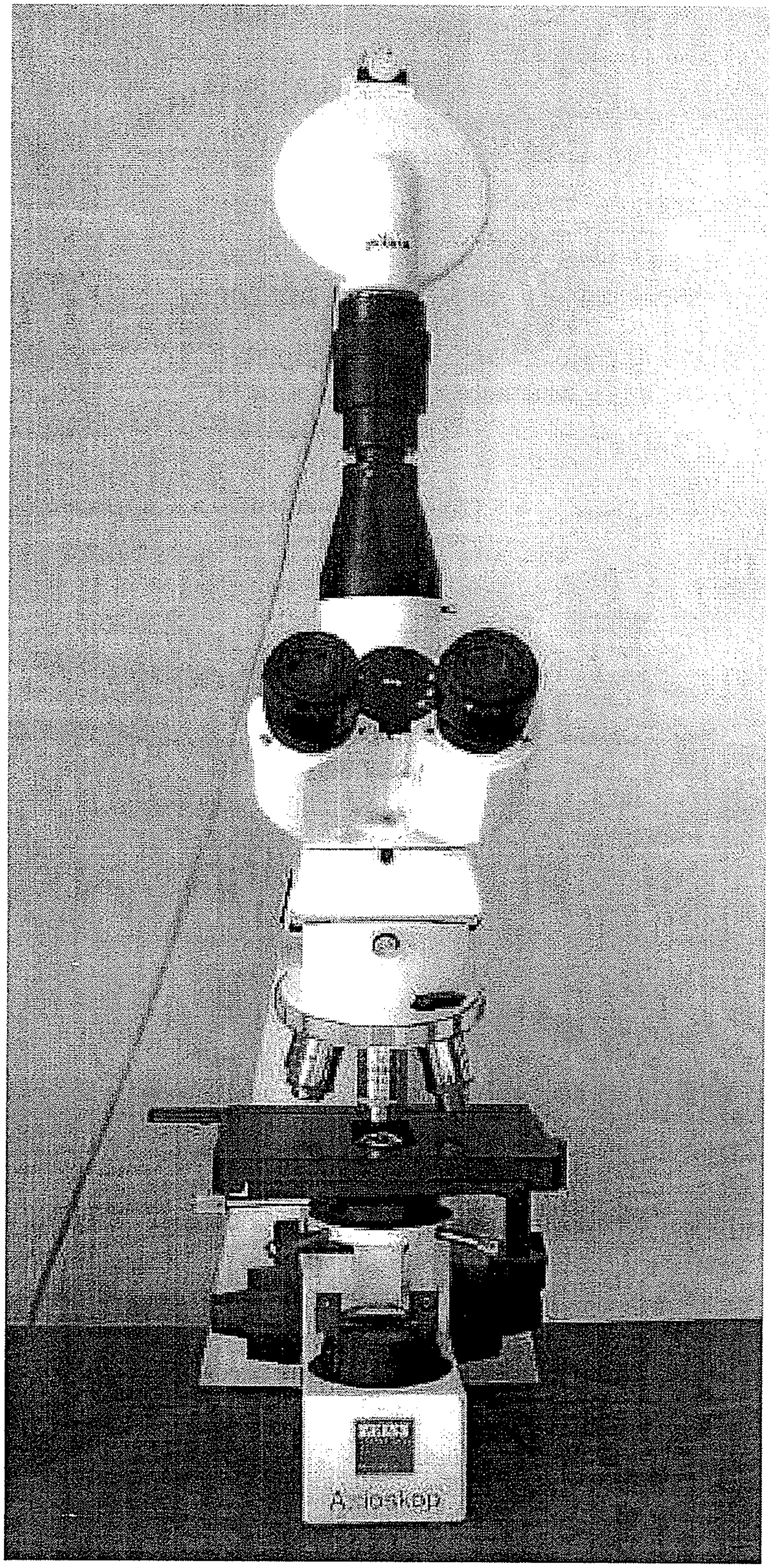


Fig. 2 Zeiss microscope – front view.

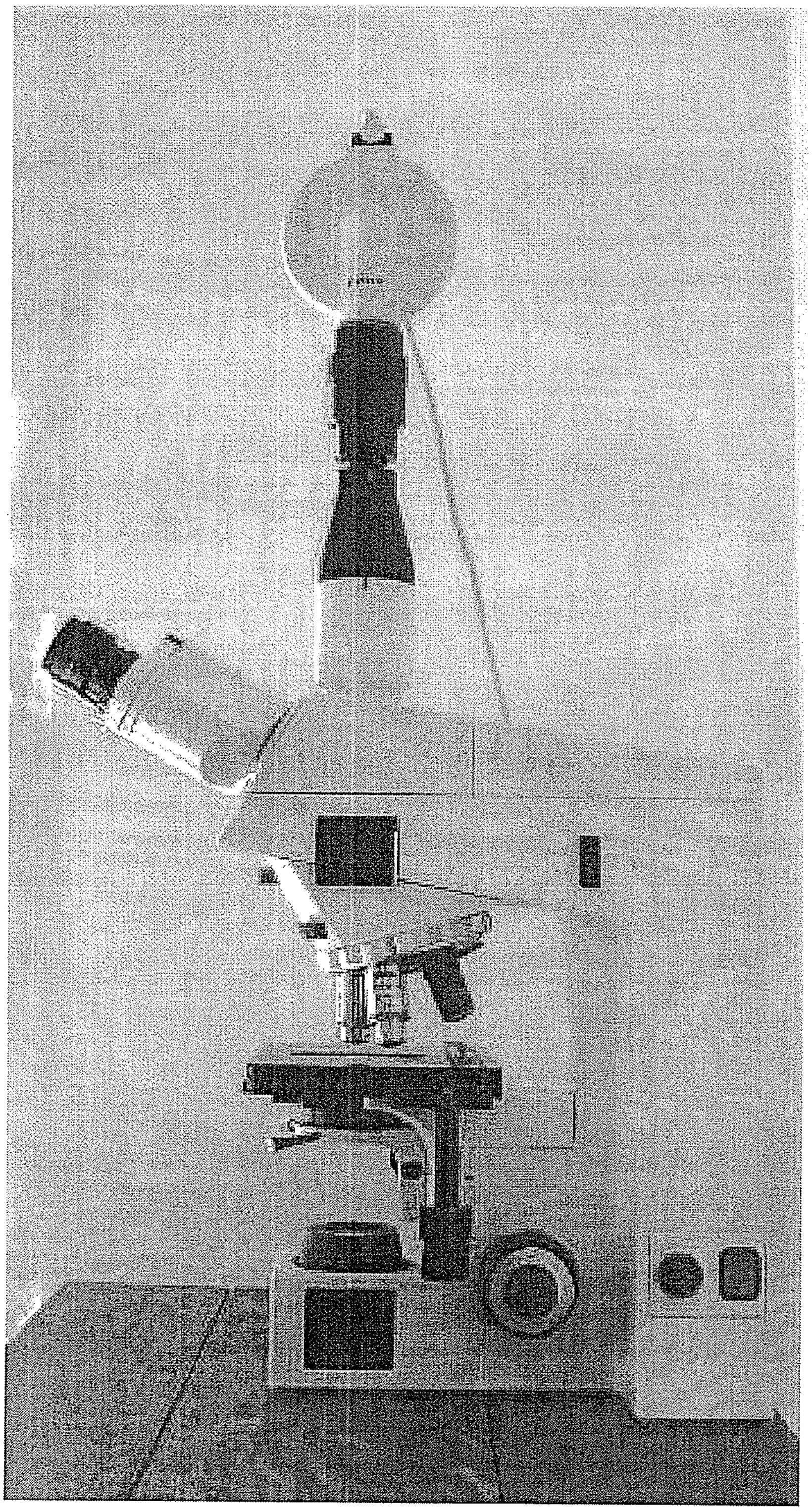


Fig. 3 Zeiss microscope – side view.

The magnification is 400x. A 500 nm low-pass filter is used and the images are captured as gray-scale TIFF files. The illumination is by a halogen bulb with its intensity set so that the upper end of the histogram of the image is just touching 255. The illumination intensity is maintained throughout the image acquisition.

5.6 Image Sampling

The 1260x960 pixel images are segmented into as pixel blocks. Each pixel holds a value between zero and 255. The darker each pixel in the block is, the more hematoxilyn and hence the more deoxyribonucleic acid (DNA) is present. This approximates Beer's Law as a means of measuring DNA concentration in the tissue on the microscopic slides. Each block is sampled by shifting left to right and top to bottom across the image (Fig. 4).

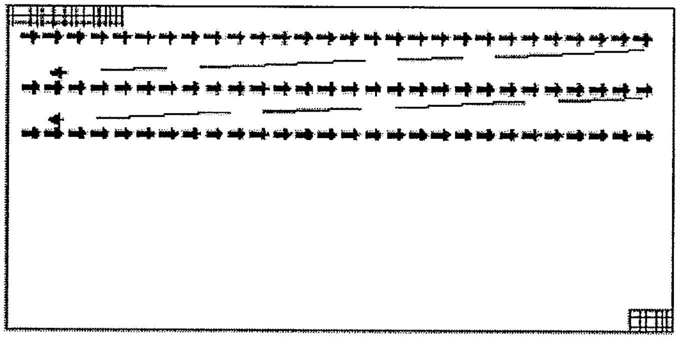
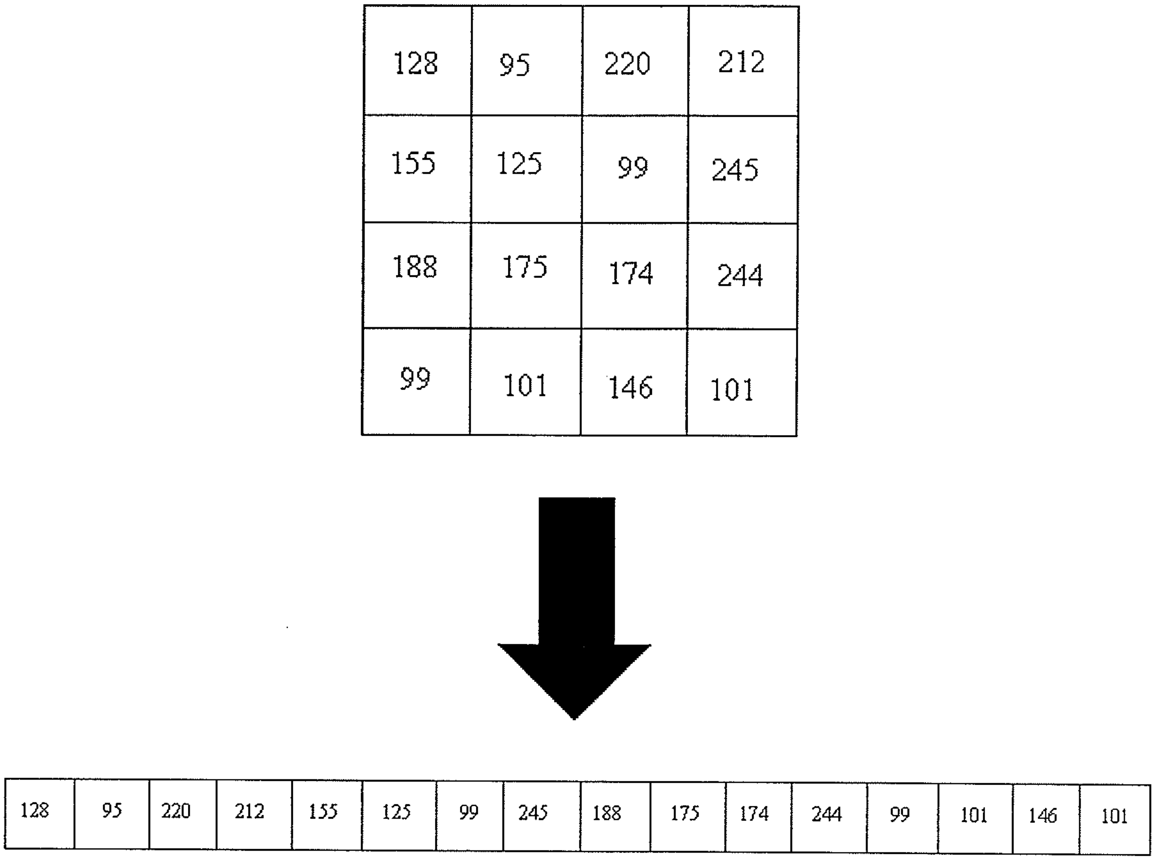


Fig. 4 Segmenting an image into blocks (vectors).

Shifts were half frame for large pixel blocks (the cut-off for this was arbitrarily set at 20 x 20) and a full frame for smaller ones. Each pixel block is then converted into a one-dimensional vector (Fig. 5). Each vector is one column entry in a training set.



128	95	220	212
155	125	99	245
188	175	174	244
99	101	146	101

128	95	220	212	155	125	99	245	188	175	174	244	99	101	146	101

Fig. 5 Organization of 4 x 4 pixel blocks into neural network training vector (not normalized)

5.7 Competitive Neural Networks as Classifiers

The training sets are presented to the different types of CNN, as shown in Figs. 6, 7, 8 and 9).

Each figure shows a schematized representation of a training set of n input signals being presented to a Kohonen layer of N processing units. Each processing unit wins the competitions a certain number of times after so many epochs of training, and have its weights w_i set at the end of training. w_{i1} represents the set of final weight values for the processing unit on the extreme left (and hence Feature # 1), w_{i2} will be the same for the next processing unit (Feature #2) and so on.

Figures 7 and 8 illustrate the regional updating of the processing units that occurs with the self-organizing variation of the Kohonen learning law. In Fig. 7, the region is one-dimensional, and discretely Gaussian around the processing unit with $z_i = 1$. Figure 7 shows the corresponding case for the two-dimensional case, with its two-dimensional discrete Gaussian shape. Figure 9 is an example of the three-dimensional SOFM, the largest map with six pre-biased weightings.

Figure 10 illustrates the LVQ with four classes.

Examples of some normalized expected weight values are shown to facilitate understanding of the figures.

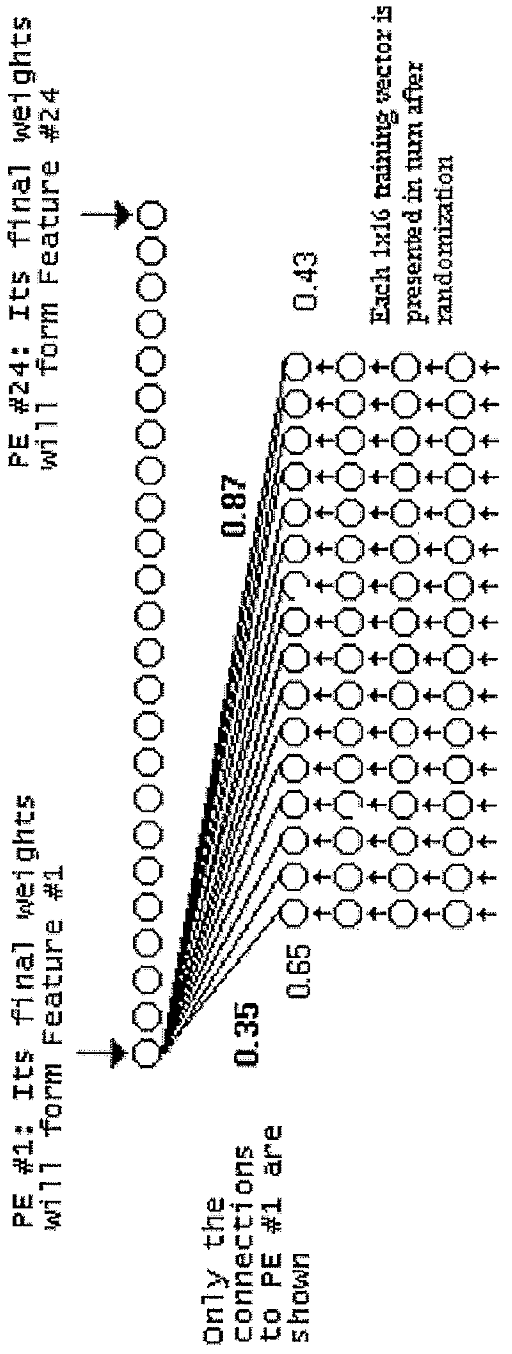

PE #1: Its final weights will form Feature #1
PE #24: Its final weights will form Feature #24
Only the connections to PE #1 are shown
0.35
0.87
0.65
0.43
Each 1x16 training vector is presented in turn after randomization

Fig. 6 Competitive neural network with Desieno conscience

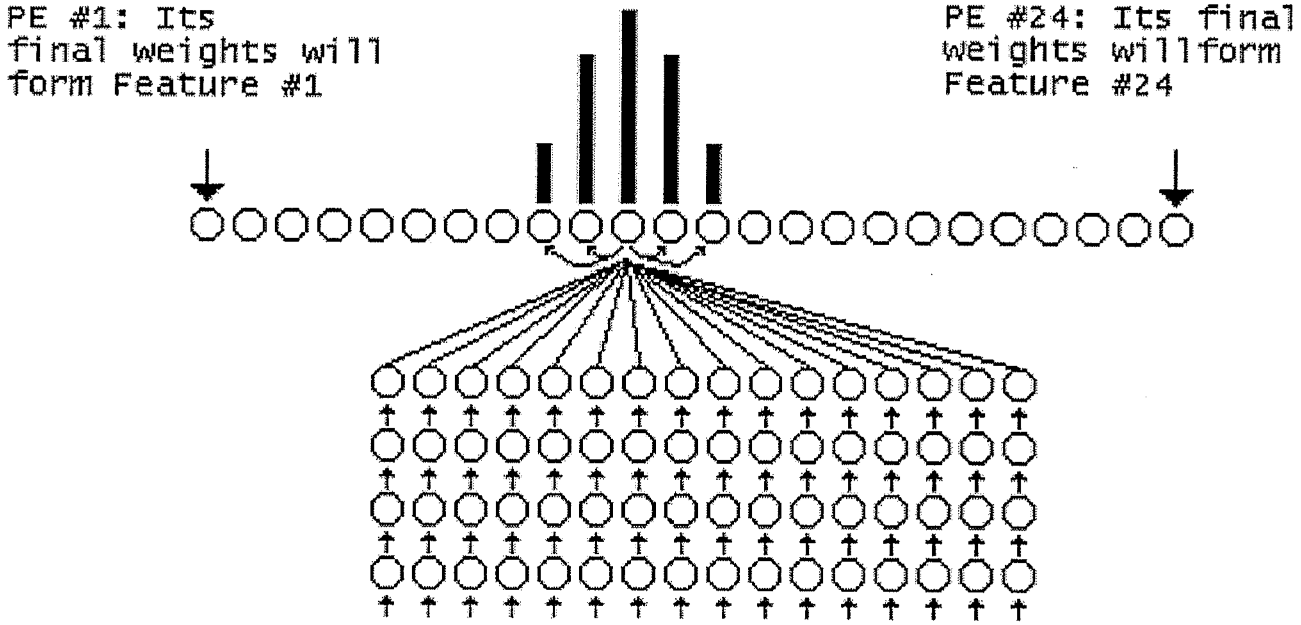

PE #1: Its final weights will form Feature #1
PE #24: Its final weights willform Feature #24

Fig. 7 One-dimensional self-organizing feature map

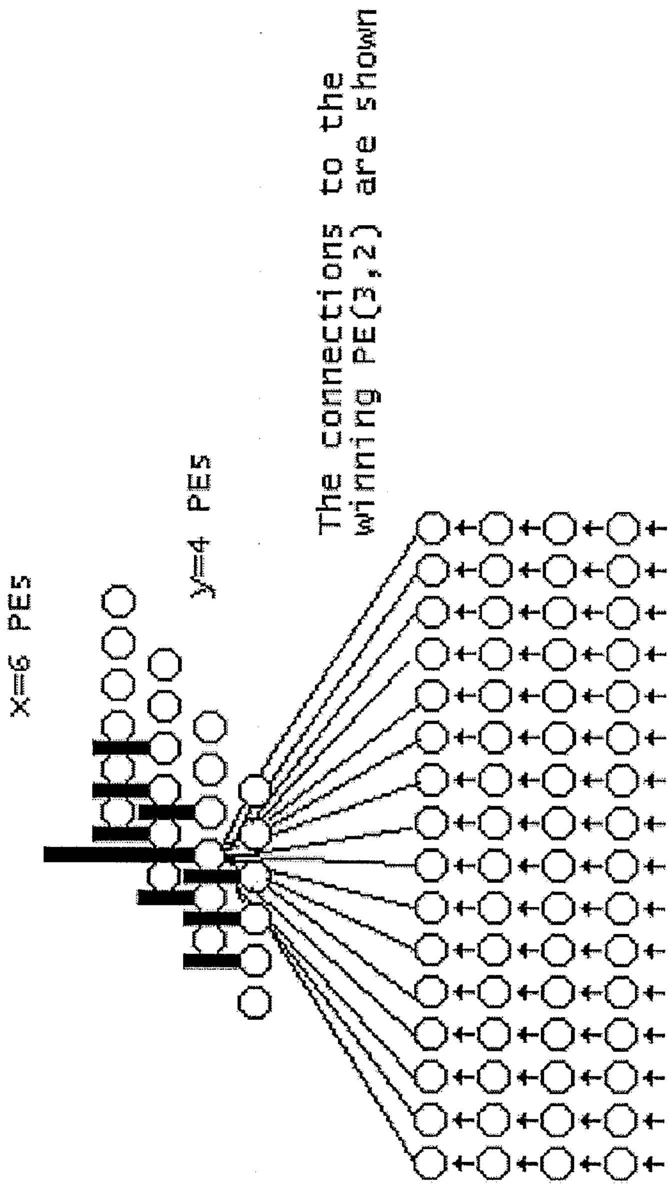


Fig. 8 Two-dimensional self-organizing feature map

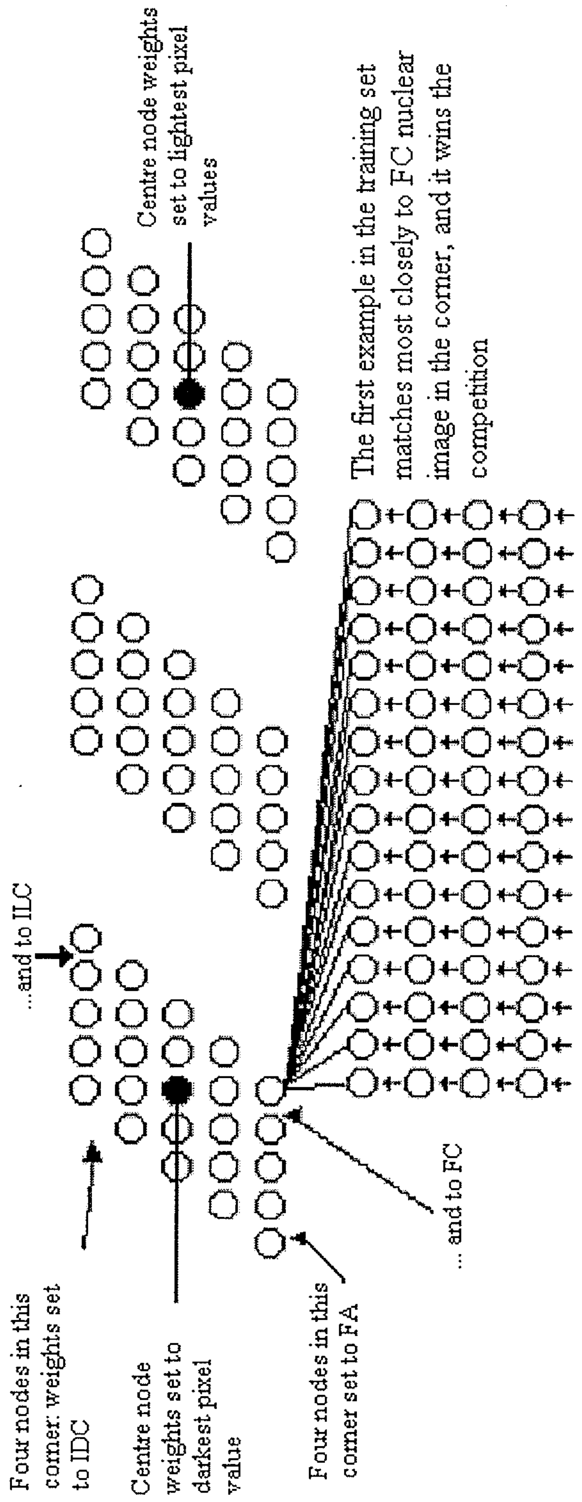

Four nodes in this corner: weights set to IDC
...and to ILC
Centre node weights set to darkest pixel value
Centre node weights set to lightest pixel values
Four nodes in this corner set to FA
... and to FC
The first example in the training set matches most closely to FC nuclear image in the corner, and it wins the competition

Fig. 9 Three-dimensional self-organizing feature map

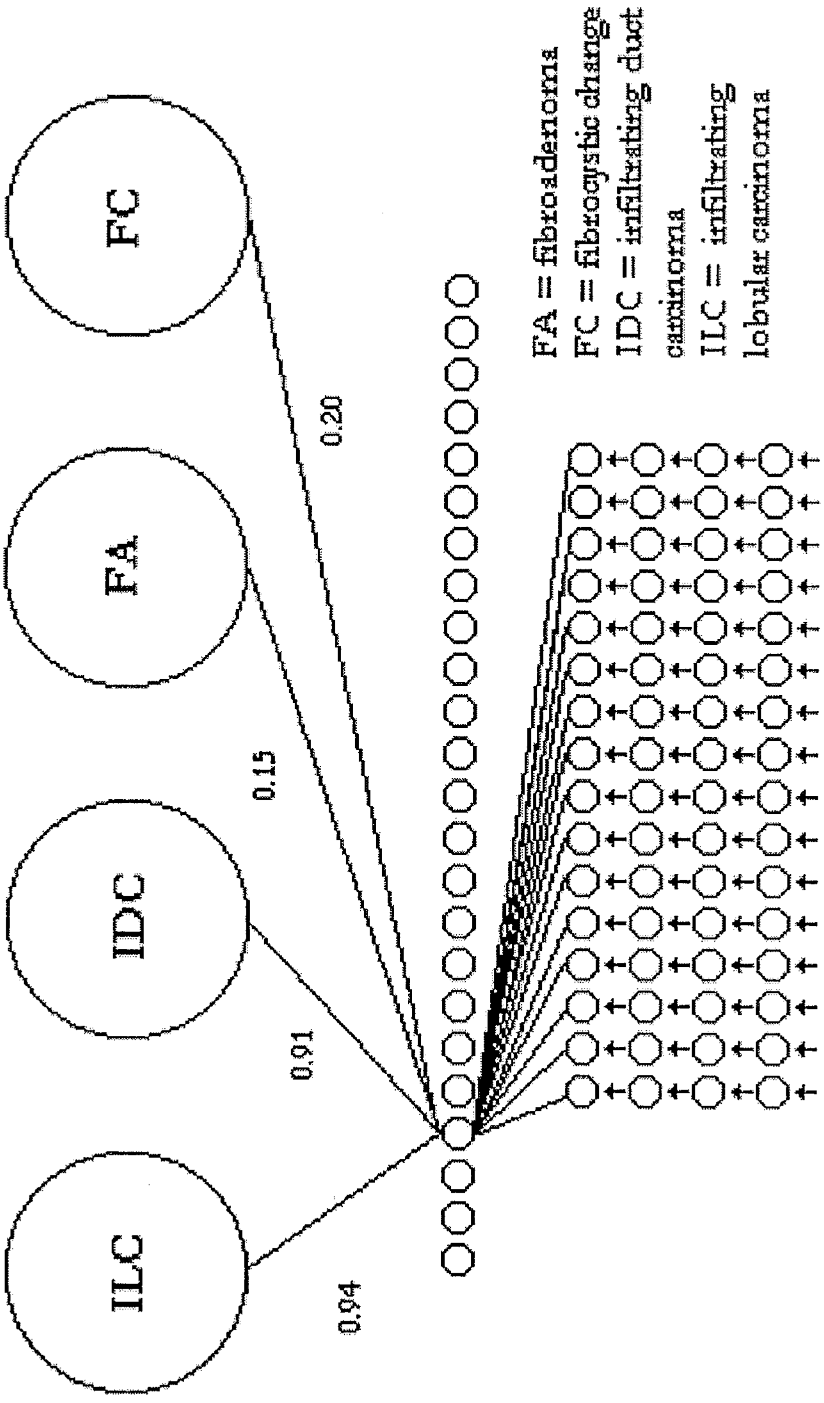

ILC
IDC
FA
FC
0.94
0.91
0.15
0.20
FA = fibroadenoma
FC = fibrocystic change
IDC = infiltrating duct carcinoma
ILC = infiltrating lobular carcinoma

Fig. 10 Learned Vector Quantizer

5.8 Computer Resources

The NNs were programmed using the *Matlab* programming language (student version 5.3, with the image processing and neural network toolboxes) on a PC with an AMD 800 MHz microprocessor with 768 MB RAM and a 20 GB hard drive.

5.9 Network Training

Each NN was trained for a certain number of epochs and then stopped. The number of epochs varied from 1000 to 2000000. The training set was presented as the verification set. There were effectively seven networks (CNN, 1D SOFM, 2D SOFM, 2 3D SOFM and 2 LVQ) to train with eight stoppages each to generate 56 trained networks for each size pixel block.

5.10 Network Testing

The training set was then given to the trained NN as the test set. Where a vector from the test set mapped to a node it was identified as having that feature. The pixels in the image where that vector was derived from were set to zero (blacked-out) to identify the location of that feature in the image. The 8 trained competitive neural network generated 24 images flagged with a feature for each of the 80 original images for a total of 15,360 result images, as did the 1D SOFM and both LVQ networks. The 2D SOFM generated 16,000 result images, the 3 x 3 x3 SOFM generated 17,280 result images, and the 5 x 5 x 3 SOFM generated 48,000 result images. For the five different sampling methods of pixel blocks there were a total of 713,600 result images to appraise.

5.11 Appraisal

The images from the blacked out images were examined visually for their importance. The numeric degree of positivity was assessed. Numbers of features in each image were counted. Comparisons of the absolute counts (absolute positivity) for each class were done.

5.12 Summary

Using eight slides representative of four different breast diseases, combined with six different competitive neural networks, several hundred thousand result images were generated and examined for visual and statistical significance, as described in the next chapter.

CHAPTER VI EXPERIMENTAL RESULTS AND DISCUSSION

6.1 Introduction

Each NN is discussed in turn. The results of the 1D SOFM will be discussed in the most detail as an archetypal competitive NN. The results and shortcomings peculiar to other NNs are stressed in the discussion in these sections.

6.2 The Competitive Neural Network with Desieno Conscience

After 1000 epochs of training, the 10 x 10 input network created a codebook of 24 features of which three output nodes shared the search space roughly equally. Twenty-one nodes were "dead", thus only three probability functions represented by the normalized weight vectors that formed the pixel blocks were used to represent the hypersphere $\mathbf{R}_n$, which is the search space. Feature #3 mapped to pixel blocks that had the high average gray scale value (a light feature), Feature #20 mapped to those that had a low average gray scale value, and Feature #15 mapped to an intermediate average gray scale value. The results per image showed some images (e.g., light-staining stroma) flagging 100% for Feature #3 and others (e.g., hypercellular areas of either malignancy or benign tissue) flagging 90% of their area to Feature #20.

This network was most noteworthy for its instability. After 5000 epochs there were again only a light, an intermediate and a dark feature to represent the $\mathbf{R}_n$. Twenty-one nodes were dead. These were represented by Features # 2, 11 and 24 respectively.

Table 1: Location of major feature - 10 x 10 training set competitive.

	Feature Type		
No. of Epochs of Training	Light	Intermediate	Dark
1000	3	15	20
5000	2	11	24
10000	5	15	22
50000	11	8	19
100000	4	13	23
200000	6	23	18
1000000	6	15	21
2000000	5	19	14

As can be seen from Table 1, the node identifying each feature tends to change with further training and did not show any stability. In this example, there is a tendency for features to distribute from light to dark from node #1 to node #24 throughout the epochs of training. However, sometimes they would crossover with the node identifying the dark feature being situated nearer to the node identifying the light feature (e.g., epochs 5,000, 100,000 and one million, as shown in Table 2).

Table 2: Location of major feature - 20 x 20 training set competitive.

	Feature Type		
No. of Epochs of Training	Light	Intermediate	Dark
1000	21	11	1
5000	15	19	1
10000	24	22	5
50000	24	15	11
100000	22	5	9
200000	18	20	2
1000000	16	21	2
2000000	23	10	3

Similarly, in the 20 x 20 training set, stability was poor and tended to change throughout training. Like the 10 x 10 training set, despite long training, additional nodes did not win competitions. In the case of the 20 x 20 dark features tended to map to the lower number nodes and light features to the higher numbers. The basic orientation of the map was set at the outset and did not tend to change.

Similar results were obtained for the other training set sizes of 4 x 4, 10 x 10 and 50 x 50.

The visual results were very similar to the digital image processing function known as "thresholding". The greatest value third of the gray scale range of pixels (for light in this case, 255 down to 204) is set to black. The visual examples of these are presented in Figs. 11, 12 and 13. Fig. 11 shows pixel blocks with the very highest average value (lightest). These pixel blocks are set to zero (black) to illustrate them. The lumena of this example of fibroadenoma, virtually empty space, is marked while darker areas (*e.g.*,cytoplasm, nuclei) are left at their original gray scale value. Fig. 12 illustrates an intermediate thresholding. Some of the cytoplasm of this example of fibrocystic change is marked as the lightest areas in Fig. 11 were. Cytoplasm is darker than the lumena but lighter than the nuclei. In Fig. 13, there is an almost complete marking of the nuclei of this example of infiltrating duct carcinoma, the darkest structure type in the image. The cytoplasm of the cells, the stroma and the very light cracks in the tissue section are not marked.

The instability of the network was probably due to a weakness in Kohonen's learning law. The pure competitive network probably gave too much statistical weight to these three features compared to other possible features (as discovered by the other networks in the following sections). This is a known shortcoming of the Kohonen learning law. It handles statistical outliers poorly. This network may have found other features to be too subtle and considered them outliers. Allowing the network to keep training merely made it move the three significant features to different locations in the output vector (hence the instability), but it did not improve performance. As is shown in the upcoming sections, regional functions (to promote self-organization and ordering) overcome this problem.

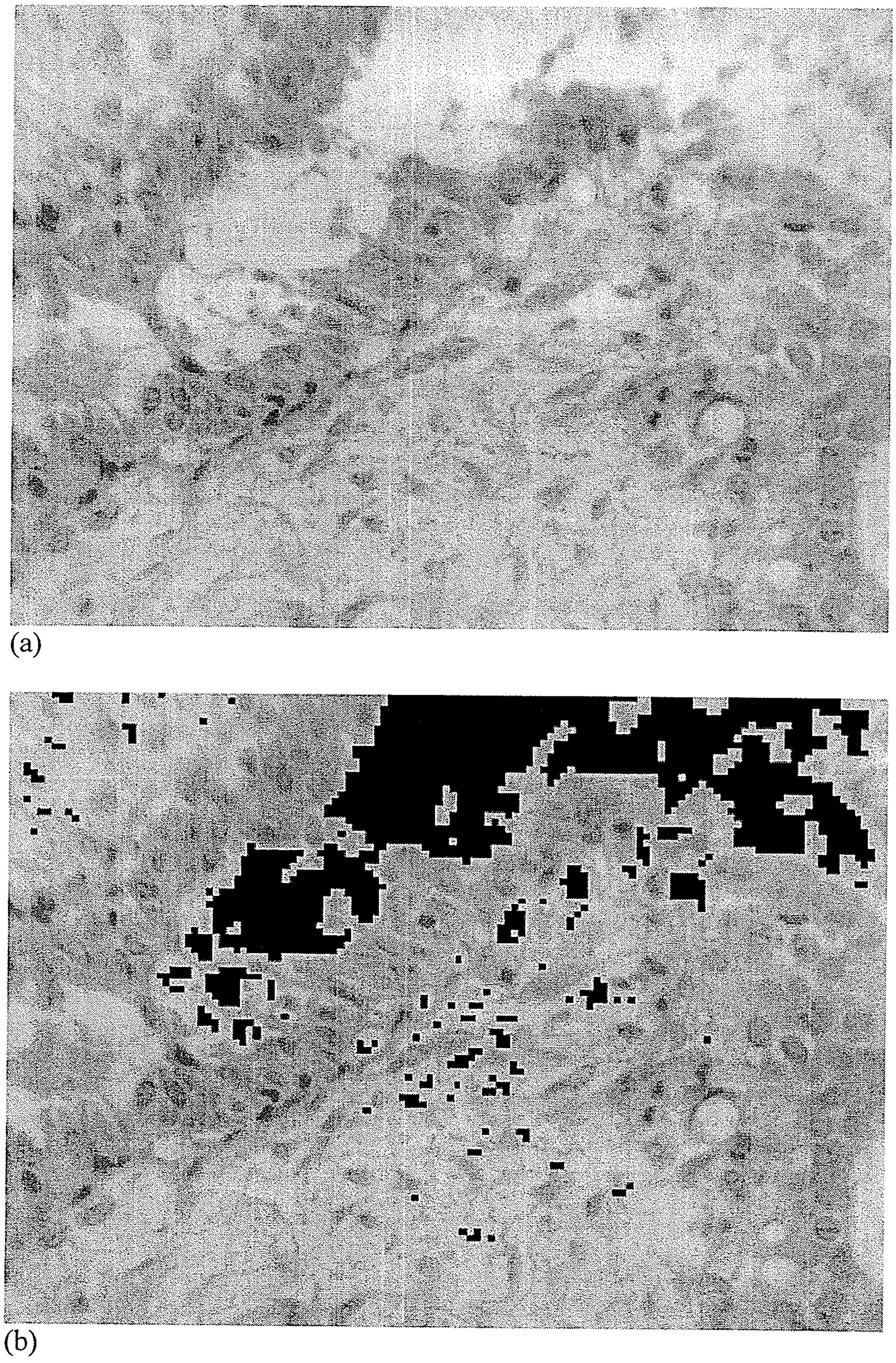
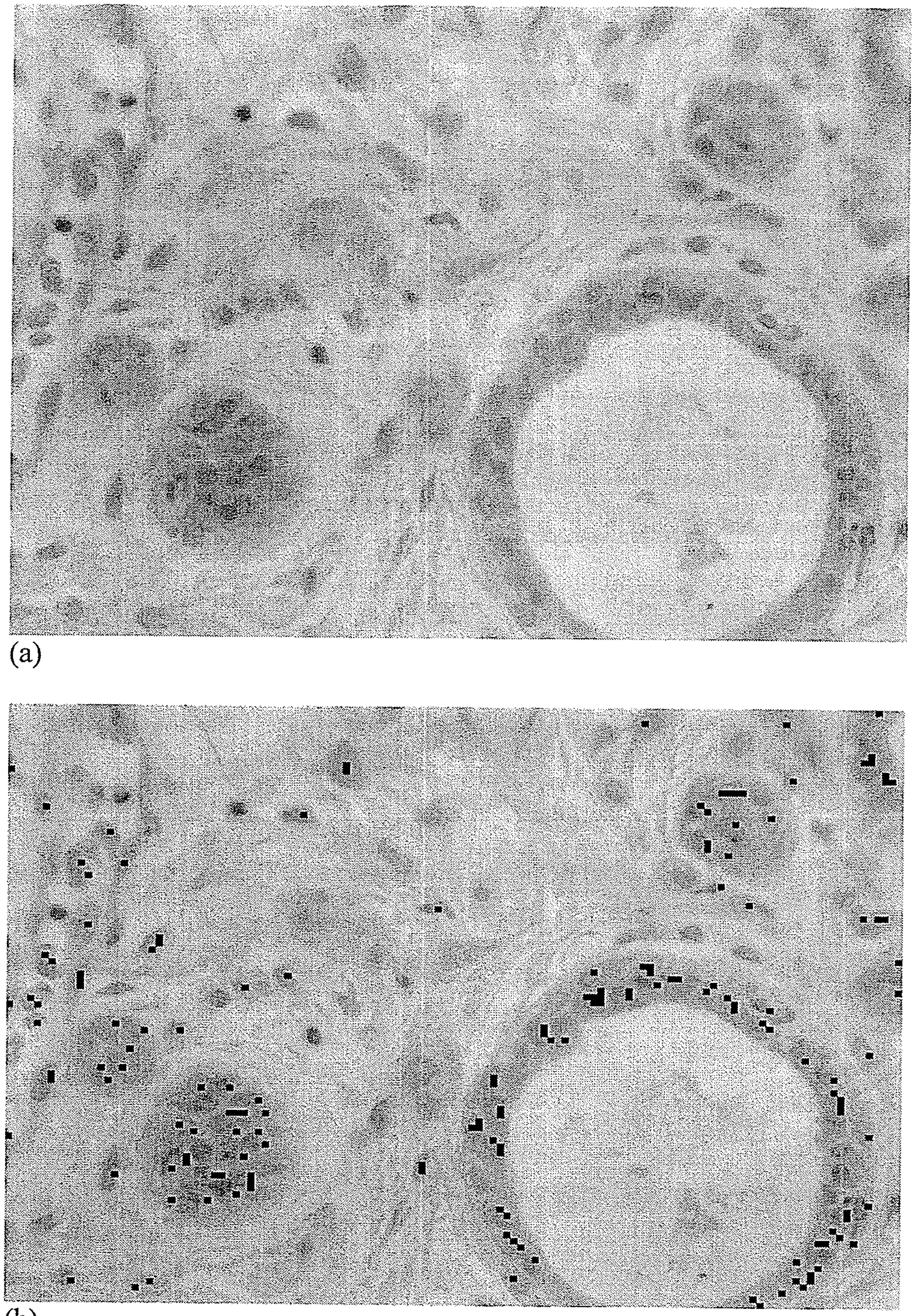


Fig. 11 Light thresholding. (a) duct and stroma of fibroadenoma viewed at 400x. (b) same image – lightest 20 x 20 pixel pixels are given a value of 0 (black) to identify them. As expected, the empty lumen has most of this type of pixel block, as well as the lightest areas in the stroma.


(a)

(b)

Fig. 12 Intermediate gray thresholding. (a) neoplastic ducts and glands in fibrocystic disease seen at 400x (b) a portion of the cytoplasm of the cells, which is darker than stroma or lumena but lighter than the cell nuclei, is marked by 20 x 20 pixel blocks set to 0 (black).

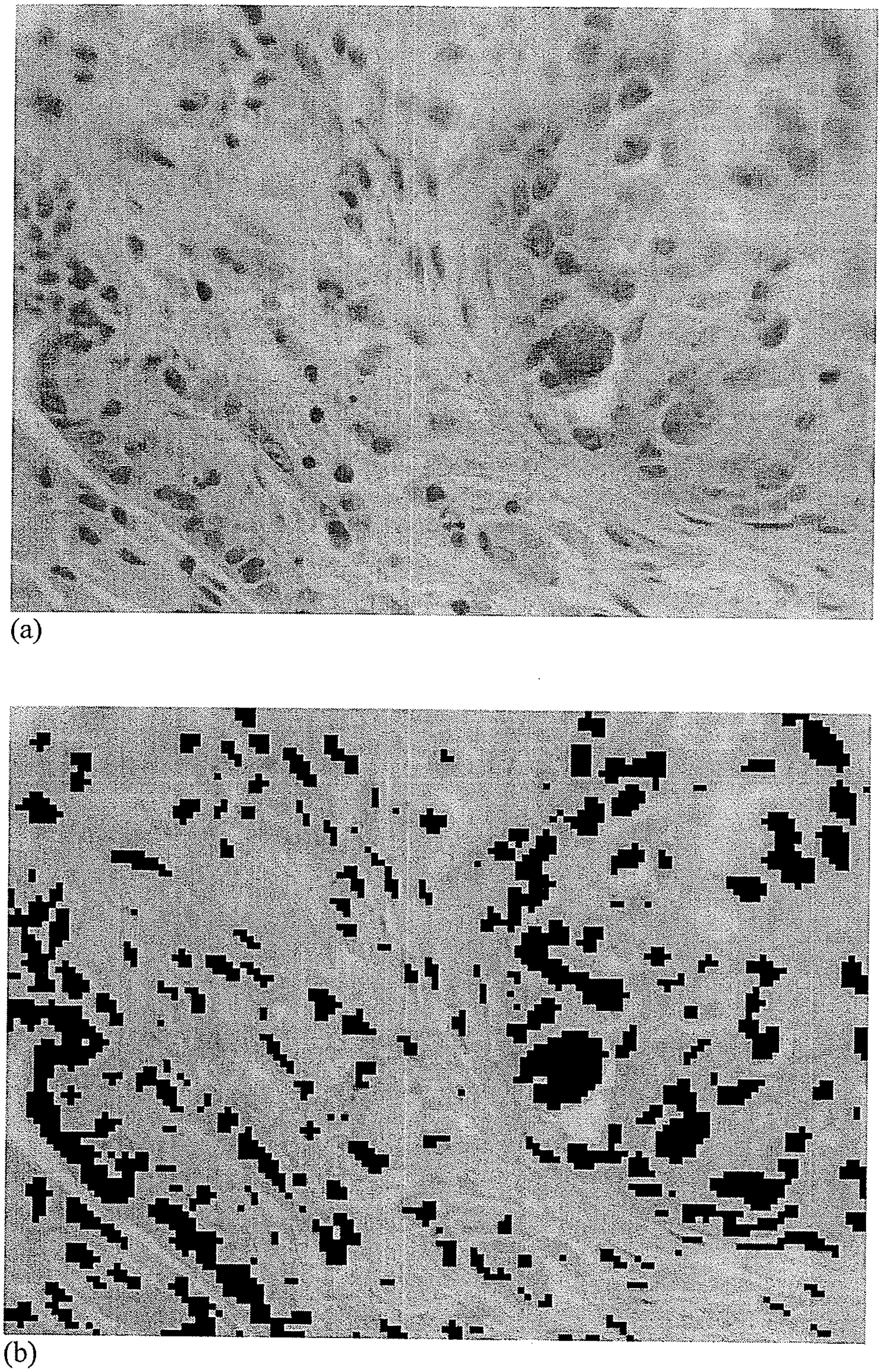


Fig. 13 Dark thresholding.(a) infiltrating duct carcinoma seen at 400x. (b) the nuclei, usually the darkest features in an microscopic section, are identified here almost in total by 20 x 20 pixel blocks that identify darkness. Few nuclei are missed, and this pixel block seems very specific for cell nuclei.

6.3 The One-Dimensional Self-Organizing Feature Map

Self-organization with local area updating showed the ability to use all the nodes in the NN and identify many more features. The 10 x 10 training set would organize itself after 1000 epochs with the feature with the greatest average grayscale value (light) mapping to the first node, the weights of each node having a progressively lower average gray scale value until the lowest value occurred at node 24. This general trend persisted throughout all the epochs of training and is a reflection of how self-organization, through local area updating, tries to order the network. In this case, it runs from light to dark from one end to the other, a perfectly natural approach to organization. With further training, the location of certain features could be seen to move to different locations in the same image (e.g., the result image for a particular feature for a particular original data image would look different after further training).

With the larger training sets (50 x 50 and 100 x 100), after only 1000 epochs, the one-dimensional SOFM would behave like the competitive neural network with Desieno conscience. Only three major features were identified. However, with further training all nodes became employed to demonstrate features.

A great difficulty was doing an analysis of all the images. For this network alone, for all five pixel block samplings, 76,800 result images were generated. When trying to identify a feature specific for a diagnosis, direct visual inspection is much faster than computer-generated statistical analysis of all the data.

Examining the images for a specific diagnostic feature, that showed up in one class of images *alone*, did not generate any such examples. All classes would flag for at least a few pixel blocks for every single feature. This occurred regardless of the length of training or the size of the pixel block.

Statistical analysis of the 10 x 10 one-dimensional SOFM gives some additional insight into the diagnostic strengths and limitations of this approach. An analysis of this is generally representative of the results of the other pixel block sampling sizes.

Table 3 shows data from the first image of fibroadenoma. It has a lot of white background (myxoid stroma), with most intermediate gray and dark pixels present in the few epithelial cells that are present in the neoplastic glands.

Table 3: Distribution of features in first image of fibroadenoma.

	No. of Epochs							
Feature type	**1000**	**5000**	**10000**	**50000**	**100000**	**200000**	**1000000**	**2000000**
1	<0.001	0.003	0.003	0.003	0.003	0.003	0.003	0.003
2	0.39	0.099	0.012	0.012	0.012	0.012	0.012	0.012
3	0.52	0.42	0.403	0.403	0.403	0.403	0.403	0.403
4	0.125	0.085	0.084	0.083	0.083	0.083	0.083	0.083
5	<0.001	0.002	0.002	0.002	0.002	0.002	0.002	0.002
6	<0.001	0.002	0.001	0.001	0.001	0.001	0.001	0.001
7	<0.001	0.003	0.002	0.002	0.002	0.002	0.002	0.002
8	<0.001	0.004	0.002	0.002	0.002	0.002	0.002	0.002
9	<0.001	<0.001	0.002	0.002	0.002	0.002	0.002	0.002
10	<0.001	0.002	0.003	0.003	0.003	0.003	0.003	0.003
11	0.002	0.004	0.004	0.004	0.004	0.004	0.004	0.004
12	0.002	0.007	0.008	0.009	0.009	0.009	0.009	0.009
13	0.04	0.065	0.062	0.061	0.061	0.061	0.061	0.061
14	0.052	0.051	0.049	0.049	0.049	0.049	0.049	0.049
15	0.005	0.007	0.006	0.006	0.006	0.006	0.006	0.006
16	0.002	0.003	0.004	0.004	0.004	0.004	0.004	0.004
17	<0.001	0.002	0.003	0.003	0.003	0.003	0.003	0.003
18	<0.001	0.002	0.003	0.003	0.003	0.003	0.003	0.003
19	<0.001	0.003	0.004	0.004	0.004	0.004	0.004	0.004
20	<0.001	0.004	0.004	0.004	0.004	0.004	0.004	0.004
21	0.163	0.148	0.148	0.149	0.149	0.149	0.149	0.149
22	0.041	0.059	0.057	0.056	0.056	0.056	0.056	0.056
23	0.009	0.017	0.016	0.016	0.016	0.016	0.016	0.016
24	<0.001	0.008	0.007	0.007	0.007	0.007	0.007	0.007
Min	0.002	0.002	0.001	0.001	0.001	0.001	0.001	0.001
Max	0.52	0.42	0.403	0.403	0.403	0.403	0.403	0.403
Mean	0.113	0.043	0.037	0.037	0.037	0.037	0.037	0.037
Std Dev	0.170	0.091	0.086	0.086	0.086	0.086	0.086	0.086

As can be seen from the table, the distribution of feature vectors concentrates on finding the lightest (features 2,3 and 4 after 1,000 epochs), darkest (21,22 and

23) and intermediate (11 to 16) and rapidly reaches a constant stable state around 10,000 epochs. Each cell in the table represents the normalized ratio of all pixel blocks that map to that particular feature. A few minor adjustments to the distribution are all that occur despite much longer training. At 10,000 epochs all the features are utilized to some degree, although few pixel blocks may map to some features (about 0.1 % in the case of feature 6). The lack of further significant weight change is partly because CNNs cannot be overtrained and tend to reach a stable state asymptotically. They can also get caught in local minima resulting in a poorly functioning network that cannot find significant features. It is difficult to say if any suboptimal performance by this network is due to getting caught in local minima during training. CNNs cannot escape local minima with further training. This has been demonstrated empirically by the so-called "Kohonen movie" where Kohonen showed a SOFM trying to map random dots in a two-dimensional space but sometimes would end up twisting itself into a knot rather than spreading itself evenly out across the space. No amount of further training could untie the knot.

Where the SOFM was well-ordered with an orderly change in average weight (output) vector values (hence an orderly change in the gray scale value of the pixel blocks/features formed from these normalized output vectors) local minima due to problems with handling gray scale are unlikely. However, it is difficult to say if the SOFM was seeking subtler perhaps spatial features that frustrated it and resulted in it being caught in a local minima.

The distribution for an image from a malignant diagnosis, the first image of infiltrating duct carcinoma, is presented in Table 4.

Table 4: Distribution of features in first image of infiltrating duct carcinoma.

	No. of Epochs							
Feature Type	**1000**	**5000**	**10000**	**50000**	**100000**	**200000**	**1000000**	**2000000**
1	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
2	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
3	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
4	0.021	0.022	0.021	0.021	0.021	0.021	0.021	0.021
5	0.022	0.023	0.023	0.023	0.023	0.023	0.023	0.023
6	0.029	0.028	0.029	0.029	0.029	0.029	0.029	0.029
7	0.033	0.033	0.035	0.035	0.035	0.035	0.035	0.035
8	0.037	0.038	0.036	0.036	0.036	0.036	0.036	0.036
9	0.044	0.044	0.044	0.044	0.044	0.044	0.044	0.044
10	0.049	0.049	0.049	0.049	0.049	0.049	0.049	0.049
11	0.059	0.057	0.057	0.057	0.057	0.057	0.057	0.057
12	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057
13	0.054	0.054	0.054	0.054	0.054	0.054	0.054	0.054
14	0.051	0.054	0.054	0.054	0.054	0.054	0.054	0.054
15	0.042	0.042	0.043	0.042	0.042	0.042	0.042	0.042
16	0.056	0.053	0.052	0.053	0.053	0.053	0.053	0.053
17	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055
18	0.06	0.06	0.06	0.06	0.06	0.06	0.06	0.06
19	0.072	0.072	0.072	0.072	0.072	0.072	0.072	0.072
20	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075
21	0.068	0.068	0.068	0.068	0.068	0.068	0.068	0.068
22	0.065	0.065	0.065	0.065	0.065	0.065	0.065	0.065
23	0.049	0.049	0.049	0.049	0.049	0.049	0.049	0.049
24	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Min	0.002	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Max	0.075	0.075	0.075	0.075	0.075	0.075	0.075	0.075
Mean	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048
Std Dev	0.018	0.018	0.018	0.018	0.018	0.018	0.018	0.018

The distribution of features is markedly different for this image. It is more cellular (because it is cancer) and has less background transilluminating as well as less stroma. Thus, there are fewer regions that map to the light features. The more even distribution of the intermediate and darker features means that there is less change in the distribution of the features than there is with the image of fibroadenoma, as the network converges towards a solution.

Although there is no single distinguishing feature for any category of diagnosis, the possibility must be considered that the distribution of features for a particular diagnosis may be characteristic of that diagnosis. If there are contradicting examples then this approach may be unreliable.

Apocrine metaplasia has some characteristics in common with infiltrating duct carcinoma as represented in these high power images. It is cellular with many relatively dark nuclei. Depending on how the image is captured, it may not show background stroma or a duct lumen. Thus the image will lack some of the more obvious characteristics that make this diagnosis benign, and the neural network may not capture the spatial information (roundness, uniform chromatin pattern, regular nucleoli) that is subtler but is very important as identifying this feature as benign.

The distribution of the features in Table 5 is similar to IDC in Table 4. Figure 14 compares the distribution of the features of the mature network for the three example images after two million epochs. There is no significant difference in the distribution between apocrine metaplasia and cancer.

Table 5: Distribution of features in image of apocrine metaplasia.

	No. of Epochs							
Feature Type	**1000**	**5000**	**10000**	**50000**	**100000**	**200000**	**1000000**	**2000000**
1	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
2	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
3	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001	<0.001
4	0.003	0.007	0.007	0.007	0.007	0.007	0.007	0.007
5	0.034	0.033	0.033	0.033	0.033	0.033	0.033	0.033
6	0.045	0.046	0.046	0.046	0.046	0.046	0.046	0.046
7	0.052	0.054	0.054	0.054	0.054	0.054	0.054	0.054
8	0.058	0.06	0.06	0.06	0.06	0.06	0.06	0.06
9	0.059	0.059	0.059	0.059	0.059	0.059	0.059	0.059
10	0.057	0.057	0.057	0.057	0.057	0.057	0.057	0.057
11	0.049	0.049	0.049	0.049	0.049	0.049	0.049	0.049
12	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048
13	0.052	0.052	0.052	0.052	0.052	0.052	0.052	0.052
14	0.054	0.053	0.053	0.053	0.053	0.053	0.053	0.053
15	0.042	0.043	0.043	0.043	0.043	0.043	0.043	0.043
16	0.055	0.055	0.055	0.055	0.055	0.055	0.055	0.055
17	0.054	0.054	0.054	0.054	0.054	0.054	0.054	0.054
18	0.053	0.053	0.053	0.053	0.053	0.053	0.053	0.053
19	0.062	0.062	0.062	0.062	0.062	0.062	0.062	0.062
20	0.064	0.064	0.064	0.064	0.064	0.064	0.064	0.064
21	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067
22	0.06	0.063	0.063	0.063	0.063	0.063	0.063	0.063
23	0.031	0.028	0.028	0.028	0.028	0.028	0.028	0.028
24	0.001	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Min	0.001	0.002	0.002	0.002	0.002	0.002	0.002	0.002
Max	0.067	0.067	0.067	0.067	0.067	0.067	0.067	0.067
Mean	0.048	0.048	0.048	0.048	0.048	0.048	0.048	0.048
Std Dev	0.018	0.017	0.017	0.017	0.017	0.017	0.017	0.017

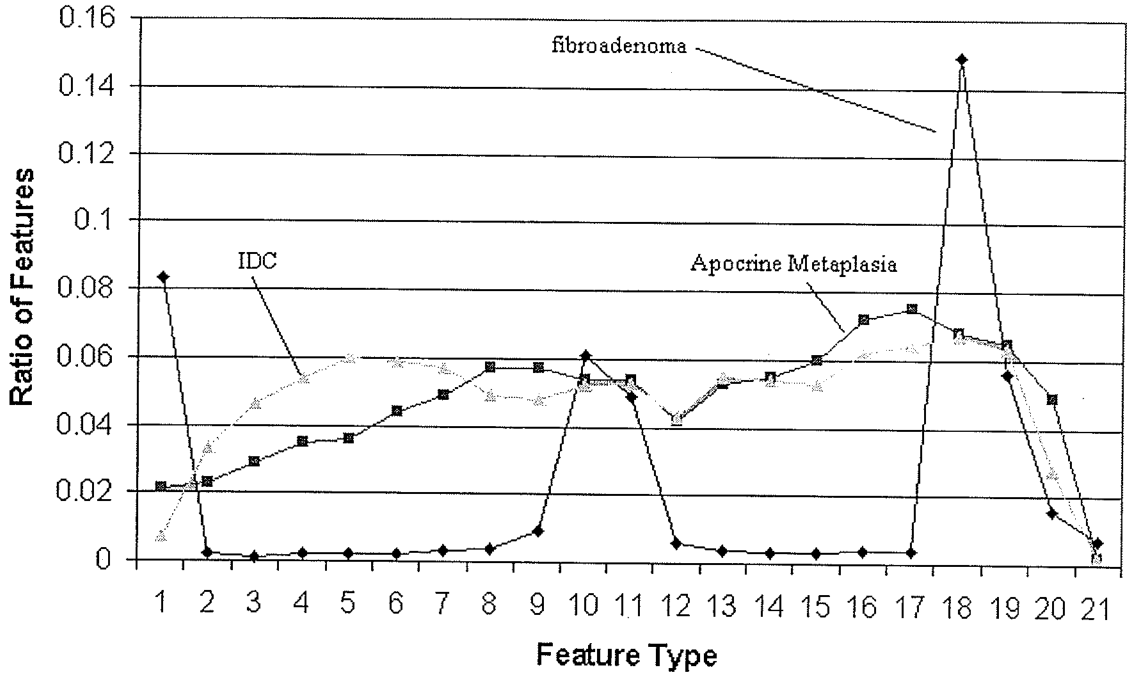
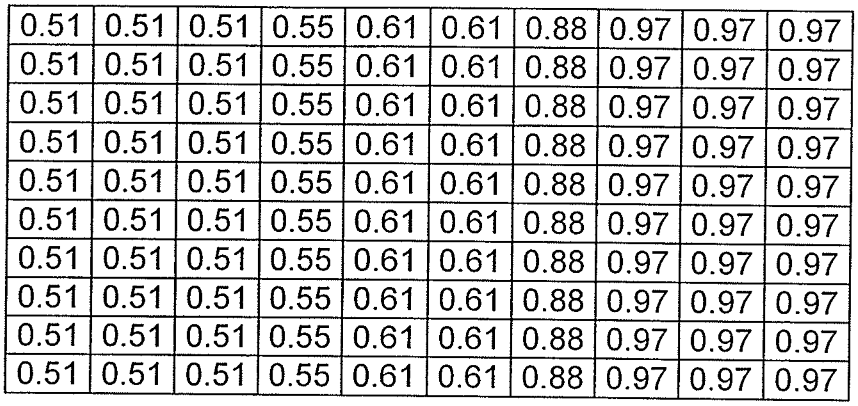


Fig. 14 Comparison of the distribution of features of IDC, apocrine metaplasia (from FC), and FA

Fig. 14 is important because it shows the distribution of features for three different histological patterns of disease: apocrine metaplasia, IDC and FA. The unique profile of FA compared to the other two patterns is noted. It could be predicted that a linear classifier could easily distinguish apocrine metaplasia from FA and IDC. However, the profiles of apocrine metaplasia and IDC are very similar even though these are very different diseases with markedly different prognoses and management regimes.

Besides being able to threshold images like the basic CNN, the one-dimensional SOFM could also detect gradients of pixels. It would flag an area in an image that showed such a gradient, like an edge or a border of a structure. As an example, the weights of node #13 after 10000 epochs of training, reconstituted as a 10 x 10 pixel block showing the normalized gray scales.



0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97
0.51	0.51	0.51	0.55	0.61	0.61	0.88	0.97	0.97	0.97

Fig. 15 Gradient of pixel values in a 10 x 10 weight vector.

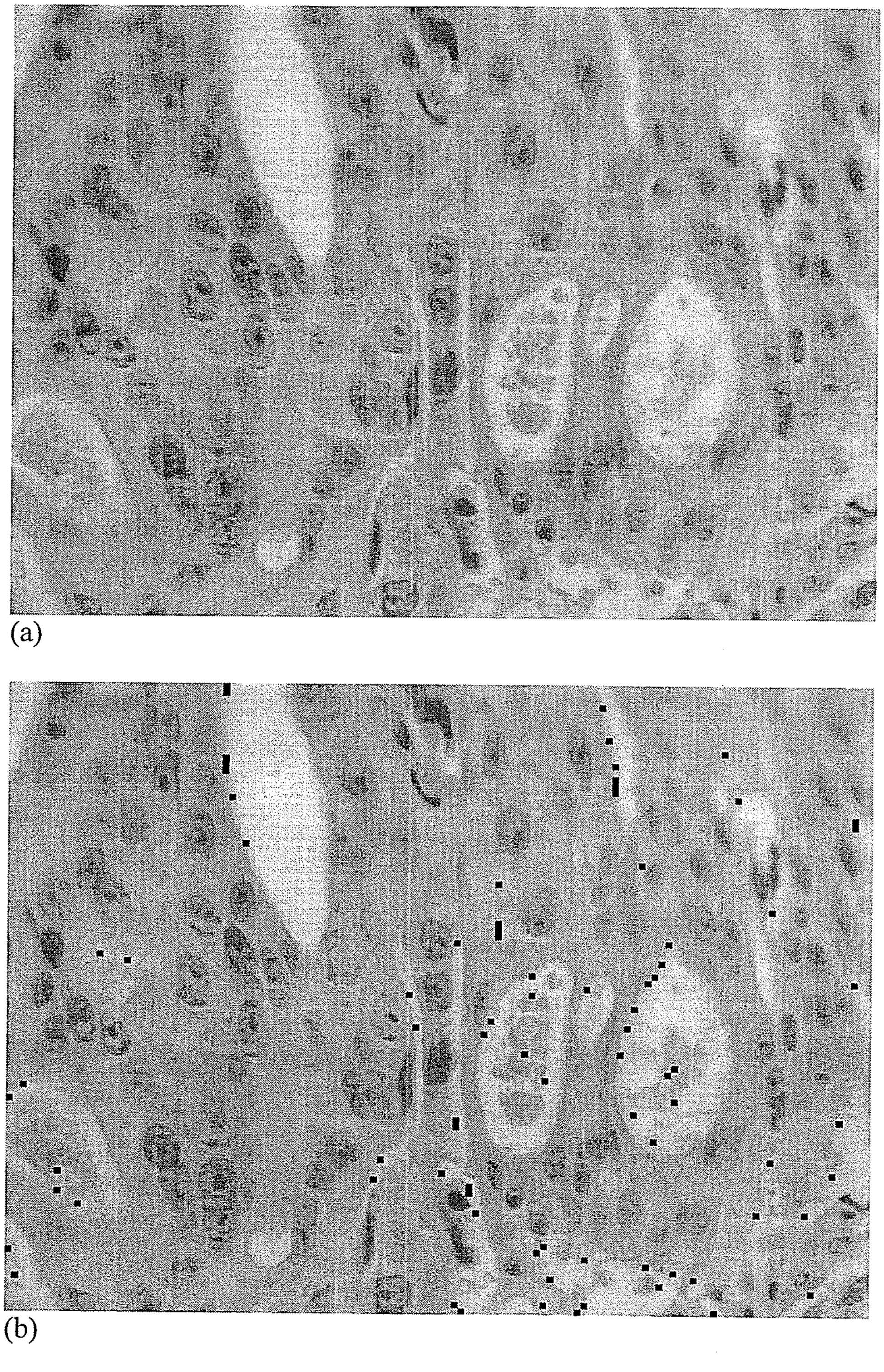


Fig. 16 Gradient detection. (a) infiltrating duct carcinoma 400x very cellular area forming glands (b) gradient of darker pixel values on the left side of the image and lighter pixel values to the right; areas showing this are blacked-out. This feature is very specific; many areas that are similar are not blacked out.

Figure 16 illustrates the detection of this gradient feature in an image of infiltrating duct carcinoma. By visual inspection, it is apparent that various portions of many malignant ducts are flagged on their left side as having the feature. This feature was not specific for any class.

All classes, with a total of twenty images each, had 24,192,000 10 x 10 pixel blocks to be categorized as features. The percentage for each class is given in Table 6.

Table 6: Comparison of distribution of all features in all images, 1D SOFM.

	Diagnosis								
Feature	FA	FC	ILC	IDC	Feature	FA	FC	ILC	IDC
1	0.04	0.03	<0	<0	13	0.02	0.02	0.02	0.02
2	0.05	0.04	<0	<0	14	0.01	0.02	0.02	0.02
3	0.09	0.03	<0	<0	15	0.02	0.02	0.02	0.02
4	0.1	0.03	0.01	<0	16	0.03	0.03	0.02	0.02
5	0.08	0.02	0.01	<0	17	0.04	0.03	0.02	0.03
6	0.04	0.01	0.01	0.01	18	0.04	0.04	0.04	0.04
7	0.03	0.02	0.01	0.01	19	0.06	0.07	0.08	0.08
8	0.03	0.04	0.01	0.01	20	0.09	0.08	0.17	0.2
9	0.01	0.03	0.01	0.02	21	0.08	0.13	0.23	0.2
10	0.01	0.03	0.03	0.03	22	0.05	0.12	0.17	0.15
11	0.02	0.03	0.03	0.04	23	0.04	0.08	0.05	0.07
12	0.01	0.03	0.03	0.03	24	0.01	0.02	0.01	0.01

The distributions do show a pattern. There are general characteristic distributions of the diagnoses, as illustrated in Fig. 17.

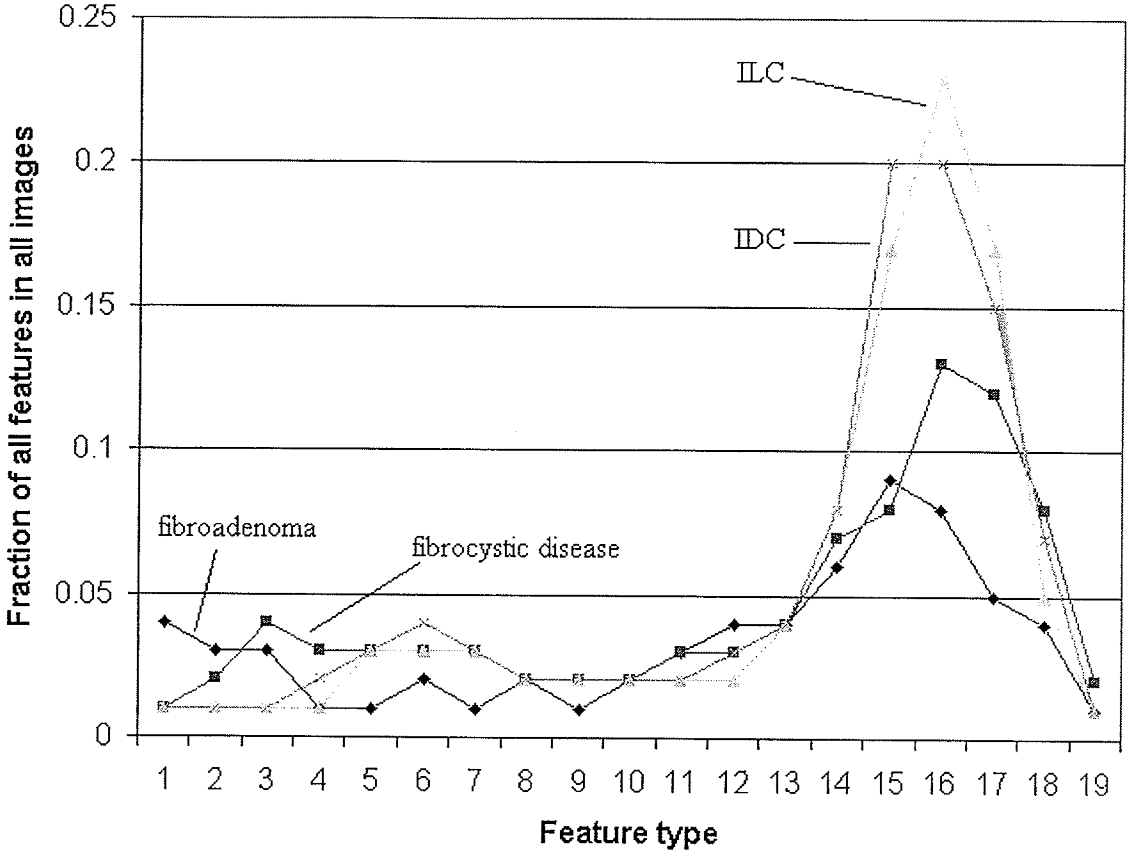


Fig. 17 Fraction of all features in all the images of a class, 1D SOFM.

6.4 The Two-Dimensional Self-Organizing Feature Map

The two-dimensional SOFM tended to organize itself with dark and light features either at opposite sides of the 5 x 5 square or in opposite corners. Sometimes the network would perform suboptimally by having light features congregating along one side and dark features congregating along an adjacent side. As a result, little space was available between the dark and light poles for the weights to develop a large range of features. As well, the nodes in the opposite corner tended to not being used.

The feature extraction was very similar to the one-dimensional case. No feature was specific for a type of diagnosis. The distribution of the features after training was similar to that seen in Tables 3, 4 and 5, although the order would be different depending on which side or corner of the 5 x 5 plane the dark and light features tended to centre themselves. Training time was roughly comparable to the one-dimensional case.

6.5 The Three-Dimensional Self-Organizing Feature Map

The 3 x 3 x 3 three-dimensional network gave a similar performance to the one- and two-dimensional SOFMs. Light and dark features tended to congregate in opposite 3 x 3 planes with intermediate gray scale features present in the middle plane. There were no features found to be specific for a diagnosis, based on visual inspection. There was no statistically significant distribution of the features after training that would help distinguish the different classes of images. Like the previous two NNs, the three-dimensional case could not make the critical distinction between apocrine metaplasia and infiltrating duct carcinoma.

The 5 x 5 x 3 three-dimensional SOFM was able to demonstrate more features than any other NN. It took the longest to train. It was still unable to discern a specific feature. Table 7 shows the relative distribution of features throughout all the images of each class.

It did demonstrate how one salt-and-pepper noise affects the network. By increasing the size of the network its sensitivity to finer, and possibly more specific, features is enhanced. However, this resulted in increased sensitivity to noise that occurred at the low and high gray scale pixel values. The sensitivity of the CCD camera is low, and low and high values might contain a disproportionate amount of noise. Although the NN averages values and thus already performs noise reduction like a median filter, it ends up searching the space created falsely by the noise at the high and low ends, prolonging the search unnecessarily.

Table 7: Distribution of features for all classes, 3D SOFM, after 100,000 epochs.

	Class								
Feature Type	FA	FC	ILC	IDC		FA	FC	ILC	IDC
1	<0.001	<0.001	<0.001	<0.001	39	0.013	0.016	0.016	0.016
2	<0.001	<0.001	<0.001	<0.001	40	0.013	0.015	0.014	0.015
3	<0.001	<0.001	<0.001	<0.001	41	0.015	0.014	0.015	0.014
4	<0.001	<0.001	<0.001	<0.001	42	0.014	0.015	0.015	0.015
5	<0.001	<0.001	<0.001	<0.001	43	0.016	0.015	0.015	0.016
6	<0.001	<0.001	<0.001	<0.001	44	0.015	0.015	0.014	0.014
7	<0.001	<0.001	<0.001	<0.001	45	0.015	0.016	0.014	0.016
8	<0.001	<0.001	<0.001	<0.001	46	0.014	0.015	0.015	0.014
9	<0.001	<0.001	<0.001	<0.001	47	0.013	0.014	0.014	0.015
10	<0.001	<0.001	<0.001	<0.001	48	0.014	0.015	0.014	0.015
11	<0.001	<0.001	<0.001	<0.001	49	0.015	0.016	0.014	0.016
12	0.002	<0.001	<0.001	<0.001	50	0.016	0.016	0.014	0.016
13	0.005	0.001	<0.001	<0.001	51	0.017	0.013	0.013	0.014
14	0.014	0.001	<0.001	<0.001	52	0.017	0.015	0.015	0.014
15	0.026	0.005	<0.001	<0.001	53	0.018	0.019	0.019	0.019
16	0.025	0.008	0.001	<0.001	54	0.017	0.018	0.018	0.018
17	0.032	0.009	0.002	<0.001	55	0.017	0.019	0.022	0.023
18	0.038	0.01	0.002	<0.001	56	0.018	0.031	0.028	0.027
19	0.036	0.012	0.002	0.002	57	0.019	0.032	0.032	0.032
20	0.034	0.012	0.005	0.006	58	0.021	0.034	0.034	0.034
21	0.032	0.012	0.006	0.005	59	0.025	0.036	0.038	0.038
22	0.032	0.012	0.008	0.009	60	0.027	0.047	0.056	0.056
23	0.03	0.012	0.011	0.012	61	0.03	0.049	0.064	0.064
24	0.026	0.012	0.012	0.012	62	0.031	0.051	0.066	0.063
25	0.023	0.013	0.013	0.013	63	0.029	0.048	0.063	0.061
26	0.019	0.014	0.016	0.016	64	0.027	0.048	0.055	0.055
27	0.015	0.013	0.013	0.013	65	0.019	0.048	0.048	0.048
28	0.013	0.012	0.012	0.012	66	0.008	0.022	0.022	0.022
29	0.013	0.012	0.013	0.013	67	0.003	0.008	0.008	0.008
30	0.001	0.012	0.012	0.012	68	<0.001	<0.001	<0.001	<0.001
31	0.004	0.011	0.011	0.012	69	<0.001	<0.001	<0.001	<0.001
32	0.005	0.011	0.011	0.011	70	<0.001	<0.001	<0.001	<0.001
33	0.007	0.012	0.012	0.012	71	<0.001	<0.001	<0.001	<0.001
34	0.009	0.014	0.013	0.013	72	<0.001	<0.001	<0.001	<0.001
35	0.009	0.015	0.014	0.014	73	<0.001	<0.001	<0.001	<0.001
36	0.01	0.011	0.012	0.011	74	<0.001	<0.001	<0.001	<0.001
37	0.011	0.011	0.011	0.011	75	<0.001	<0.001	<0.001	<0.001
38	0.012	0.012	0.012	0.012					

	FA	FC		ILC	IDC
Min	0.001	0.001		0.001	0.002
Max	0.038	0.051		0.066	0.064
Mean	0.018	0.018		0.019	0.020
Std Dev	0.009	0.013		0.016	0.016

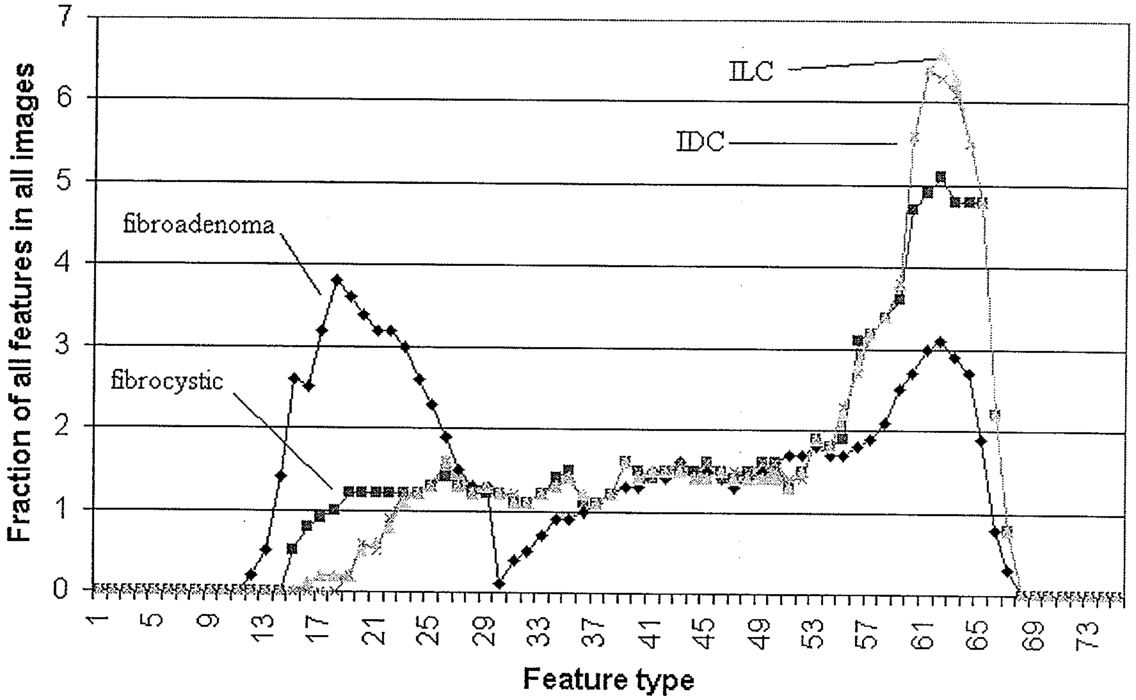


Fig. 18 Fraction (x100) of all features in all the images of a class, 3D SOFM

One problem identified with the three-dimensional SOFM was its occasional tendency to organize itself in a manner that was not the most efficient use of space. The best performance of this network (best and widest distribution of features) would be achieved when the features with the darkest and lightest average gray scale values were most distant from each other. This was achieved when these features were found either in opposite corners or opposite sides, preferably in the centers of opposite 3 x 3 or 5 x 5 planes. As mentioned in Chapter V, to try to assure this outcome, theses nodes in the three-dimensional SOFM could be biased instead of randomized before training. However, despite this, the darkest and lightest regions may form in close proximity to each other, resulting in many underutilized nodes in remoter corners of the SOFM (Fig. 19).

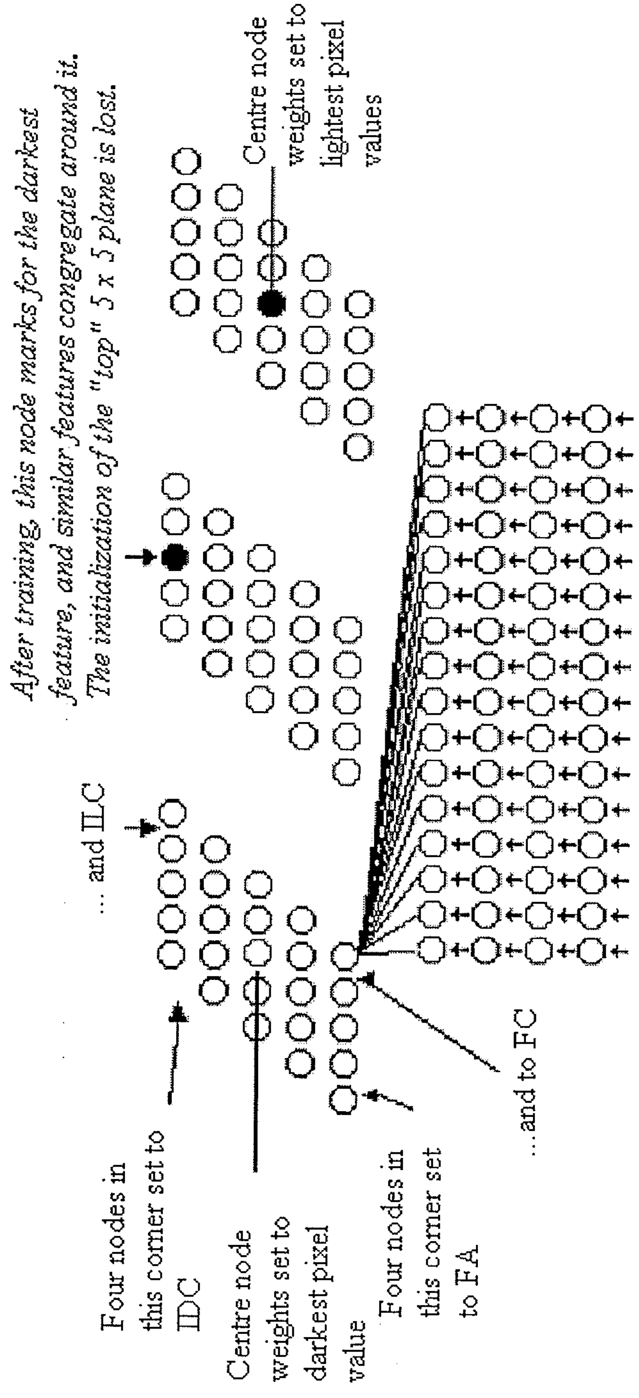

After training, this node marks for the darkest feature, and similar features congregate around it. The initialization of the "top" 5 x 5 plane is lost.
Centre node weights set to lightest pixel values
... and ILC
Four nodes in this corner set to IDC
Centre node weights set to darkest pixel value
Four nodes in this corner set to FA
...and to FC

Fig. 19 This 3D SOFM is "bent". The most extreme features are too close together and nodes in the top plane are under used.

6.6 The Learned Vector Quantizer

The 24 node learned vector quantizer performed the same as the 1D SOFM of the same size. Distribution of features was virtually the same as in Fig. 17. Long training did not generate any features specific for a class. After any length of training, regardless of whether the LVQ used two classes (benign or malignant) or all four classes, there was no specific feature detected.

In the set of diagnostically representative images of infiltrating duct carcinoma, with a learned vector quantizer looking for 24 features after 50,000 epochs, feature # 21 covered 28 percent of the surface area of all IDC images, as opposed to 23% of the area of all the surface area of all the ILC images and 17% of FC images and 5% of the FA images. This would seem to have predictive value for malignancy and even specifically infiltrating duct carcinoma. However, the range of values for the set of images in a class was: 0 to 25% for fibrocystic change, 0 to 15% for fibroadenoma, 3 to 45% for infiltrating lobular carcinoma and 7 to 57% for infiltrating duct carcinoma. Closer direct visual inspection showed that this feature was identifying nuclei. It missed some malignant nuclei and flagged many benign nuclei such as fibroblasts and endothelial cells. It was able to identify IDC solely by appreciating the tendency of the disease to be hypercellular and hyperchromatic.

The ability to make a diagnosis based on statistical information is dependent on the image selected by the operator. Selection of the stroma or carcinomatous tissue would have favoured a diagnosis of benign disease.

6.7 Summary

The NNs are able to identify interesting features in all images. They tend to be able to detect significant gray scale ranges of pixels, akin to thresholding, distributed throughout the range of the part of the 8-bit gray scale present in the sampled images. The more sophisticated NNs (SOFM and LVQ) also detected many gradients of gray scale in the images. These were represented as a gradient of average values in the pixel block that represented the feature.

The profiles of the distribution of features throughout all images show a tendency to match particular classes. However, some images containing patently benign features (e.g., apocrine metaplasia) could not be distinguished from malignancy by any of the NNs used, and in their operating modes used. Nevertheless, the work has demonstrated the behaviour of the NNs under specific architectures and parameter settings.

CHAPTER VII
CONCLUSIONS AND RECOMMENDATIONS

7.1 Conclusions

Competitive neural networks are able to identify interesting visual features in representative images of fibrocystic change (FC), fibroadenoma (FA), infiltrating lobular carcinoma (ILC) and infiltrating duct carcinoma (IDC). Testing a variety of different neural networks (NNs) with many different parameter settings demonstrated the ability to threshold and gradient detect throughout equiprobable domains of the hypersphere $\mathbf{R}_n$ (search space) of the set of images that represented breast disease.

There was no feature detected that was specific for a particular disease, or benignancy or malignancy.

NNs could not discern features that could distinguish between cells of different histogenesis in the same image.

There are some statistical associations between the relative distributions of the features and the diagnoses. However, a significant pitfall was the inability of the NN to distinguish some images with very different diagnoses. Selection of images by the operator has a great bearing on the networks' ability to distinguish different classes.

Self-organizing feature maps (SOFMs) can fail to organize themselves in the most efficient manner. This occurs even when nodes are biased to assist training.

Once a SOFM starts organizing itself inefficiently, it is best to stop training and start over.

Salt and pepper noise affects training by enlargening the search space unnecessarily and thus prolonging the search for equiprobable output vectors and by wasting vectors to cover the noise space. This could be eliminated by truncating the image histogram at the high and low ends.

The Desieno conscience NN was never able to find anything more than three significant features, all thresholding in type.

Learned vector quantizers (LVQs) do not perform any better than SOFMs for trying to find specific features. However, they are not as prone to inefficient organization during training as the higher-order SOFMs.

7.2 Recommendations

The problem of raising the specificity of at least one or even some of the extracted features can be parsed into qualitative and quantitative approaches. However, these two approaches are not entirely separate, as will be shown.

The *qualitative* approach will be require refining of, and eventually moving away from, the "brute force" approach applied in this thesis. The first step will be noise reduction by applying either truncating the histograms of the images or applying a smoothing function to all images.

Anticipating that this will not yet result in highly specific feature extraction, the next approach is to refine the algorithm. Pixel block sampling should tend be larger (to capture more spatial information) and perhaps be of a more natural shape than square. Experimentation should be limited to employing LVQ so as to benefit from human guidance of clinical data and increase the efficiency of the experimentation at this juncture. The experiments will benefit from the improvements in random access memory and microprocessor speed that have occurred lately, as well as the uses of clustered microprocessing to further enhance computational speed.

It is conceivable that such a combination of qualitative changes may be fortunate to yield features of high predictive value. But incorporation of *quantitative* techniques, with careful planning, should make a very significant difference.

A quantitative approach means the counting of the number of pixel blocks identified as a feature in different representative images and making statistical comparisons. This should be done at intervals of epochal training to gage the potential of the NN. If an extracted feature can be shown to have a statistically significant association with a particular diagnosis, the NN that generated it can be further worked with to improve its specificity. A difference of greater than 95% confidence limit of the number of blocks in representative images could be a sufficient, and realistic, way of judging the specificity of a feature.

Statistically significant features that fall short of the 95% figure could still be used to guide further experimentation. In *genetic algorithm* theory, such a network could be seen as having an elevated fitness and thus be subjected to mutation and then further training. The new selected networks may even extract a 100% specific feature, but it is more likely that the new networks would approach the 95% confidence limit after pixel blocks are counted. The process could be fully automated in the *Matlab* programming code, with instructions to stop when the above criteria are met.

The numbers of examples of breast disease presented to the NN must be increased to the point where sufficient examples of all breast disease are available to it. In the design of a new experiment, specialist LVQs should be employed that try to distinguish between a particular disease and *all* other disease that could occur in breast tissue.

Changes should be made with the resources used from the histology laboratory. Tissue should be fresh, cut at a uniform 4 μm, and stained with Feulgen-stain. This would better emphasize the information in the nucleus and eliminate the noise that is present in eosin-stained tissue.

Ideally, a special microscope should be used with a red laser light source and 10 nm band-pass filter to enhance image sampling.

REFERENCES

[AgBa77] R.K. Aggarwal and J.W. Bacus, "A multi-spectral approach for scene analysis of cervical cytology," *Journal Histochemistry and Cytochemistry*, vol. 25, pp. 668-680, 1977.

[AlKi96] R. Allan and W. Kinsner, "A two-dimensional histogram of brightness-texture space for nuclear-cytoplasmic segmentation," *Journal of Computer-Assisted Microscopy*, vol. 8, no. 1, pp. 31-40, 1996.

[AsWi92] M.L. Astion and P. Wilding, "The application of backpropagation neural networks to problems in pathology and laboratory medicine," *Arch. Pathol. Lab. Med*, vol. 116, pp. 995-1001, 1992.

[BaRC92] H. Banda-Gamboa, I. Ricketts, A. Cairn, K Hussein; JH Tucker; N Husain, "Automation in cervical cytology: An overview," *Analytical Cellular Pathology*, vol. 4, pp. 25-48, 1992.

[BaWD88] P.H. Bartels, J. Weber and L. Duckstein, "Machine learning in quantitative histopathology," *Analytical and Quantitative Cytology and Histology*, vol. 10, no. 4, pp. 299-306, 1988.

[Beck94] R.L. Becker, "Computer-assisted image classification: Use of neural networks in anatomical pathology," *Cancer Letters*, vol. 77, pp. 111-117, 1994.

[BoKo93] M. Boon and L.P. Kok, "Neural network processing can provide means to catch errors that slip through human screening of Pap smears," *Diagnostic Cytopathology,* vol. 4, pp. 411-416, 1993.

[BKNH94] M. Boon and L.P. Kok, M. Nygaard-Nielsen, K. Holm, K. Holund, “Neural network processing of cervical smears can lead to a decrease in diagnostic variability and an increase in screening efficacy: A study of 63 false-negative Smears,” *Modern Pathology*, vol. 7, no. 9, pp. 957-961, 1994.

[Bruga79] G.I. Brugal, “Double scanning microphotometer for image analysis, hardware, software, and biomedical applications,” *Journal Histochemistry and Cytochemistry*, vol. 27, pp. 212-217, 1979.

[CaSa42] T. Caspersson and L. Santesson, “Studies on protein metabolism in the cells of epithelial tumours,” *Acta Radiology*, vol. 46 (Suppl), pp. 1-105, 1942.

[DaAW91] A.E. Dawson, R.E. Austin and D.S. Weinberg, “Nuclear grading of breast carcinoma by image analysis;” *American Journal of Clinical Pathology*, vol. 95, pp. 29-37, 1991.

[Desi88] D. Desieno, “Adding a conscience to competitive learning,” *Proc. Int. Conf. On Neural Networks*, vol. I, IEEE Press, New York, pp. 117-124, 1988.

[DhAr93] A.P. Dhawan and L. Arata, “Segmentation of medical images through competitive learning,” *Computer Methods and Programs in Biomedicine*, vol. 40. pp. 203-215, 1993.

[DyWB88] H.E. Dytch, G. Wied and M. Bibbo, "Artificial neural acts as tools for expert systems in objective histopathology" in *Proceedings of 10th Annual International Conference, IEEE Engineering in Medicine & Biology*. vol 10, pp. 1375-1382, 1988.

[DyWi90a] H.E. Dytch and G. Wied, "Neurocomputing: Artificial neural nets acts as AI tools for quantitative pathology" in *Expert Systems in Cytopathology*, JPA Baak, ed., New York: Springer-Verlag, pp. 485-502, 1990.

[DyWi90b] H.E. Dytch and G.L Wied, "Artificial neural networks and their use in quantitative pathology," *Analytical and Quantitative Cytology and Histology,* vol. 12,no.6, pp. 379-393, 1990.

[Ehri36] W.E. Ehrich, "Nuclear sizes in growth disturbances with special reference to the tumor cell nuclei,." *Am J Med Sci.*, vol. 192, pp. 772-790, 1936.

[GaCB86] C. Garbay, J.M. Chassery and G. Brugal, "An iterative region growing process for image segmentation based on local color similarity and global shape criteria," *Analytical and Quantitative Cytology*, vol. 8, pp. 25-34, 1986.

[GoWo92] R.C. Gonzales and R.E. Woods, *Digital Image Processing*, Reading, MA: Addison-Wesley, 1992, 716 pp.

[Gros78] S. Grossberg, "A theory of human memory: Self-organization and performance of sensory motor codes, maps and plans," in *Progress in Theor. Biol.* R. Rosen and S. Snell, eds., vol. 5, pp. 233-374, 1978.

[Gros76] S. Grossberg, "Adaptive pattern classification and universal decoding: I Parallel development and coding of neural feature detectors," *Biol. Cyber*. vol. 23, pp. 121-134, 1976.

[HGBS93] H. Harms, U. Gunzer and I. Baumann, S. Serbouti, "Malignancy-associated changes in monocytes in acute leukemias measured by high-resolution image processing," *Analytical and Quantitative Cytology and Histology*, vol. 15, no.6, pp. 371-378, 1993.

[Hayk94] S. Haykin, *Neural Networks*. Maxwell Macmillan Canada, Inc., 1994.

[Hebb49] D. Hebb, *The Organization of Behavior*. Wiley, New York, 1949.

[Hech90] R. Hecht-Nielsen, *Neurocomputing*. Addison-Wesley Publishing, Reading, MA, 1990.

[Holm78] J. Holmquist, "Computer analysis of cervical cells: Automatic feature extraction and classification," *Journal of Histochemistry and Cytochemistry*, vol. 26, pp. 1000-1017, 1978.

[Hopf84] J.J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Natl. Acad. Sci*, vol. 81, pp. 3088-3092, 1984.

[Jaco25] W. Jacobj, "Uber das rhythmische Wachstum der Zellen durch Verdoppellung ihres Volumens," *Arch. Entwickl. Mech. Org.*, vol. 106, pp. 124-192, 1925.

[Kara99] P. Karakitsos, "Learning vector quantizer in the investigation of thyroid lesions," *Analytical and Quantitative Cytology and Histology*, vol. 21(3), pp. 201-208, 1999.

[KhKO93] M. Kharazi, S. Keyhani-Rofagha and R. O'Toole, "Automated screening of cervical cytology using PAPNET: Comparable cytological findings with biopsy," *Am. J. Clin. Path.*, vol. 99, pp. 337-338, 1993.

[KiSh92] J.K. Kish and D. Shelk, "Evaluation of the performance of the PAPNET automated cytological screening system on the conventionally prepared cervical Papanicolaou smears at Hinsdale Hospital, Hinsdale, Illinois," *Analytical and Quantitative Cytology and Histology*, vol. 14, pp. 267 - 282, 1992.

[Koho84] T. Kohonen, *Self-organization and Associative memory*. Berlin: Springer-Verlag, 1984.

[LKKK94] A. Langi, K. Ferens, W. Kinsner, T. Keckt, and G. Sawatsky. "Intelligent storm identification system using a hierarchical neural network,." Proc. IEEE *Can. Conf. Elec. & Comp. Eng.*, Halifax, Canada, pp. 501 - 504, 1994.

[MaPa89a] C. MacAuley and B. Palcic, "An edge relocation algorithm segmentation algorithm," *Analytical and Quantitative Cytology and Histology*, vol. 12, no.3, pp. 165-172, 1989.

[MaPa89b] C. MacAuley and B. Palcic, "A comparison or some quick and simple threshold selection methods for stained cells," *Analytical and Quantitative Cytology and Histology*, vol. 10, no.5, pp. 134-138, 1989.

[Mang68] O.L. Mangasarian, "Multisurface method of pattern separation, *IEEE Transactions on Information Theory*, vol. 14, no.6, pp. 801-807, 1968.

[Mang94] L. Mango, "Computer-assisted cervical screening using neural networks," *Cancer Letters*, vol. 77, pp. 155-162, 1994.

[Mang93] L. Mango, "Rescreening of 89 negative cervical smears preceding invasive carcinoma by the PAPNET cytologic screening system," *Acta. Cytol.*, vol. 37, pp. 780, 1993.

[McPi43] W.S.McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bulletin of Math. Bio.*, vol. 5, pp. 115-133, 1943.

[MeSH94] B. Mehdi, D. Stacey and G. Harauz, "A hierarchical neural network assembly for the classification of cervical cells in automated screening," *Analytical Cellular Pathology*, vol. 7, pp. 171-180, 1994.

[MeSi51] R.C. Mellors and R. Silve, "A microfluorometric scanner for the differential detection of cells: Application to exfoliative cytology," *Science*, vol. 114, pp. 356-360, 1951.

[MeGP54] R.C. Mellors, A. Glassman and G. A. Papanicolaou, "A microfluorometric scanning method for the detection of cancer cells in smears of exfoliated cells," *Cancer*, vol. 5, pp. 458-468, 1954.

[Mins54] M. Minsky, "Neural nets and the brain-model problem" *Doctoral Dissertation,* Princeton University, Princeton, NJ, 1954.

[MiPa69] M. Minsky and S. Papert, *Perceptrons*. MIT Press, Cambridge MA, 1969.

[Muka83] A. Mukawa, "Progress report on experimental use of CYBEST model 2 for practical gynecologic mass screening: Alterations of the specimen, rejection threshold and specimen preparation," *Analytical and Quantitative Cytology*, vol. 5, pp. 31-34, 1983.

[OlMB92] T.J. O'Leary, U.V. Mikel and R.L. Becker, "Computer-assisted image interpretation: Use of a neural network to differentiate tubular carcinoma from sclerosing adenosis," *Modern Pathology*, vol. 5(4), pp. 402-405, 1992.

[OlPT77] L.H. Oliver, R.S. Poulsen and G.T. Toussaint, "Estimating false-positive and false negative error rates in cervical cell classification," *Journal of Histochemistry and Cytochemistry*, vol. 25, pp. 696-701, 1977.

[PMST92] B. Palcic, C. MacAuley, S. Shlien, W. Treurniet, H. Tezcan and G. Anderson, "Comparison of three different methods for the automated classification of cervical cells," *Analytical Cellular Pathology*, vol. 4, pp. 429-441, 1992

[PSGO93] B. Palcic, B. Susnik, D. Garner and I. Olivotto, "Quantitative evaluation of malignant potential of early breast cancer using high resolution image cytometry," *Journal of Cellular Biochemistry*, vol. 17G, pp. 107-113, 1993.

[PrMe66] J.M.S. Prewitt and M.L. Mendelsohn, "The analysis of cell images," *Ann NY Acad Sci.*, vol. 128, pp. 1035-1043, 1966.

[PyTa80] D. Pycock, C.J. Taylor, "Use of the MAGISCAN image analyser in automated uterine cancer cytology," *Analytical and Quantitative Cytology*, vol. 2, pp. 303,1980.

[RESB79] E.R. Reinhardt, R. Erhardt, P. Schwarzmann, W.H. Bloss and R. Ott, "Structure analysis and classification of cervical cells using a processing system based on TV," *Analytical and Quantitative Cytology*, vol. 1, pp. 143-150, 1979.

[Rose58] F. Rosenblatt, "The Perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Review*, vol. 65, pp. 386-408, 1958.

[Rose93] D.L. Rosenthal, "Negative Pap smears preceding carcinoma of the cervix: Rescreening with the PAPNET system," *Am. J. Clin. Pathol.*, vol. 100, pp. 331, 1993.

[RuMc86] D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing in the Microstructure of Cognition, I & II.* MIT Press, Cambridge, MA, 1986.

[ScTP00] F. Schnorrenberg, N. Tsapatsoulis, C.S. Pattichis, C.N. Schizas, S. Kollias, M. Vassiliou, A. Adamou and K. Kyriacou, "Improved detection of breast cancer nuclei using modular neural networks," *IEEE Engineering in Medicine and Biology.* pp. 48-62 Jan/Feb 2000.

[Sher94] M.E. Sherman, "PAPNET analysis of reportedly negative smears preceding the diagnosis of a high-grade squamous intraepithelial lesion or carcinoma," *Mod. Path.*, vol. 7, pp. 578-581, 1994.

[StOW62] L.M. Stark and M. Okajima and G.H. Whipple, "Computer pattern recognition techniques," *Comm. of the ACM*, vol. 5, pp. 527-532, 1962.

[Toll55] W.E. Tolles, "The Cytoanalyzer: An example of physics in medical research," *Ann NY Acad Sci*, vol. 17, pp. 250-256, 1955.

[TuHu80] J.H. Tucker and O.A.N. Husain, "Trials with the CERVISCAN experimental prescreening device on polylysine prepared slides," *Analytical and Quantitative Cytology*, vol. 2, pp. 186-224, 1980.

[Vond73] C. Von der Malsberg, "Self-organization of orientation sensitive cells in the striate cortex," *Kybernetik*, vol. 14, pp. 85-100, 1973.

[Wata76] S. Watanabe, "A Pap smear prescreening system: CYBEST," in *Digital Processing of Biomedical Images.* K. Preston Jr., L. Onoe, eds., New York: Plenum Press, pp. 227-241, 1976.

[Werb74] P.J. Werbos, "Beyond regression: New tools for prediction and analysis in the behavioral sciences," *Doctoral Dissertation,* Appl. Math, Harvard University, November 1974.

[WBBO68] G.L. Wied, P.H. Bartels, G.F. Bahr, D.G. Oldfield, "Taxonomic intra-cellular analytic system for cell identification," *Acta Cytologica,* vol. 12, no.3, pp. 180-204, 1968.

[WoMa90] W.H. Wolberg and O.L. Magasarian, "Computer-aided diagnosis of breast aspirates via expert systems," *Analytical and Quantitative Cytology and Histology*, vol. 12, pp. 314-320, 1990.

[Zahn79] D.J. Zahniser, "BioPEPR: a system for the automatic prescreening of cervical smears," *J. Histochemistry and Cytochemistry*, vol. 27, pp. 635-641, 1979.

APPENDICES

Appendix A: Matlab Code for this Thesis

```
% create training sets for ILC,IDC, FA and FC

a=49;  % GLOBAL VARIABLE the row length of the mask
b=49;  % GLOBAL VARIABLE the column length of the mask
n=25;  % the incrementation of the reading frame
nn=20; % GLOBAL VARIABLE the number of images to be read in

q=[];
for f=1:n:1260-a % generates the first indices of the
location of the square sampling
  for d=1:n:960-b
     p=[d f];
     q=[q;p];
  end;
end;

%1. The ILC SOFM training sets

for z=1:nn
   cd J:\Thesis_Images;
   zzz=['ilc' num2str(z) '.tif'];
   A=imread(zzz);
   cd D:\thesis\ilcdata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
```

```
    ilcs{z}=E;  % the square pixel blocks are saved for each
 image in a cell array
 end

  trilc=cat(2,ilcs{:});
  rsz=size(trilc);
  trilc=trilc(:,randperm(rsz(:,2)));
  save ilcs.mat ilcs;
  save trilc.mat trilc;

  %2. The IDC SOFM training sets

 for z=1:nn
     cd J:\Thesis_Images;
     zzz=['idc' num2str(z) '.tif'];
     A=imread(zzz);
     cd E:\thesis\idcdata;
     A=A(:,:,1);
     g=size(A);
     h=g(1)-a;
     j=g(2)-b;
     D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
 set in relation to the size of the image and the sampling
 rate
     m=0;
     for e=1:n:j   % reading along the row "a" pixels at a
 time(column-by-column)
        for d=1:n:h % reading along the column "b" pixels at
 a time (row-by-row);
          m=m+1;
          D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
       end;
   end;
   E=cat(2,D{:});
   idcs{z}=E;  % the square pixel blocks are saved for each
 image in a cell array
 end

  tridc=cat(2,idcs{:});
  rsz=size(tridc);
  tridc=tridc(:,randperm(rsz(:,2)));
  save idcs.mat idcs;
  save tridc.mat tridc;

  %3. The FA SOFM training sets

 for z=1:nn
```

```
   cd J:\Thesis_Images;
   zzz=['fa' num2str(z) '.tif'];
   A=imread(zzz);
   cd F:\thesis\fadata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
  fas{z}=E;  % the square pixel blocks are saved for each
image in a cell array
end

 trfa=cat(2,fas{:});
 rsz=size(trfa);
 trfa=trfa(:,randperm(rsz(:,2)));
 save fas.mat fas;
 save trfa.mat trfa;

 %4. The ILC SOFM training sets

for z=1:nn
   cd J:\Thesis_Images;
   zzz=['fc' num2str(z) '.tif'];
   A=imread(zzz);
   cd G:\thesis\fcdata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
  fcs{z}=E;  % the square pixel blocks are saved for each
image in a cell array
end

 trfc=cat(2,fcs{:});
 rsz=size(trfc);
 trfc=trfc(:,randperm(rsz(:,2)));
 save fcs.mat fcs;
 save trfc.mat trfc;


% the master program for studying the LVQ and Competitive
Neural Networks
addpath D:\Thesis_Images\fa85-20095
addpath D:\Thesis_Images\fa86-2107
addpath D:\Thesis_Images\fc85-10830
addpath D:\Thesis_Images\fc85-30689
addpath D:\Thesis_Images\idc86-22790
addpath D:\Thesis_Images\idc86-37485
addpath D:\Thesis_Images\ilc84-35370
addpath D:\Thesis_Images\ilc86-23410
addpath C:\MATLAB_SR11\work\ilcdata
addpath C:\MATLAB_SR11\work\idcdata
addpath C:\MATLAB_SR11\work\fadata
addpath C:\MATLAB_SR11\work\fcdata

a=9;  % GLOBAL VARIABLE the row length of the mask
b=9;  % GLOBAL VARIABLE the column length of the mask
n=10;  % the incrementation of the reading frame
nn=20; % GLOBAL VARIABLE the number of images to be read-in
for each diagnosis

q=[];
for f=1:n:1260-a % generates the first indices of the
location of the square sampling
  for d=1:n:960-b
     p=[d f];
     q=[q;p];
  end;
end;

load trilc.mat;
load tridc.mat;
load trfa.mat;
load trfc.mat;
rsz=size(trilc);
ilcx=ones(1,rsz(:,2));% the target vector for ILC for
training LVQ
idcx=ones(1,rsz(:,2))*2;
trmaster=[ilcx ilcx idcx idcx;trilc tridc trfa trfc];% the
master training set. note: the first row is the LVQ target
vector
clear trilc tridc trfa trfc ilcx idcx;
rsz=size(trmaster);
trmaster=trmaster(:,randperm(rsz(:,2)));

fe1=24; %number of features for the 1D-SOFM, LVQ and
Competitive neural networks

DD=(a+1)*(b+1); %the training vectors are 1-D vectors made
from square pixel blocks
prt=200000; % the size of the submatrix of "trmaster"
training set to be
% used, one submatrix at a time, for training the LVQ and
compettive networks

te1=double(trmaster(2,:));% "te1" will be the LVQ target
vector for distinguishiing benign and malignant. It is
already randomized
te1=ind2vec(te1);
te1=full(te1);% the true target vector, of zeros and ones.
te2=double(trmaster(1,:));% "te1" will be the LVQ target
vector for distinguishiing benign and malignant. It is
already randomized
te2=ind2vec(te2);
te2=full(te2);% the true target vector, of zeros and ones.
trmaster=trmaster(3:end,:);% the training set(randomized)
is separated.
rsz=size(trmaster);
iter=fix(rsz(:,2)/prt)-1; %the number of required samplings
of the
% whole training set

AA=ones(DD,1)*double(min(min(trmaster)))/255;% the lower
limit of all pixel values
```

```
BB=ones(DD,1)*double(max(max(trmaster)))/255;% the upper
limit of all pixel values
CC=[AA BB];
clear AA;
clear BB;

netcomp=newc(CC,[fe1]);% the competitive neural networks
are created
netlvq1=newlvq(CC,fe1,[0.25 0.25 0.25 0.25]);% the LVQ
networks are created, with each tissue diagnosis as a
target
netlvq2=newlvq(CC,fe1,[0.50 0.50]);% the LVQ networks are
created, with benign vs. malignant as a target

for aaa=0:iter % the training set is broken into segments
      trmast=double(trmaster(:,aaa*prt+1:(aaa+1)*prt))/255;
      netcomp=train(netcomp,trmast);
      tee=te1(:,aaa*prt+1:(aaa+1)*prt);
      netlvq1=train(netlvq1,trmast,tee);
      clear tee;
      tee=te2(:,aaa*prt+1:(aaa+1)*prt);
      netlvq2=train(netlvq2,trmast,tee);
      clear trmast;
      clear tee;
end
trmast=double(trmaster(:,(iter+1)*prt+1:end))/255;
netcomp=train(netcomp,trmast);% the tailend of the training
set is processed
tee=te1(:,(iter+1)*prt+1:end);
netlvq1=train(netlvq1,trmast,tee);% the tailend of the
training set is processed
clear tee;
tee=te2(:,(iter+1)*prt+1:end);
netlvq2=train(netlvq2,trmast,tee);% the tailend of the
training set is processed
clear trmast;
clear tee;

wtscomp=netcomp.IW{1,1}; % the network weights are saved
after every 200 epochs
save wtscomp.mat wtscomp;

wts1lvq1=netlvq1.IW{1,1};
save wts1lvq1.mat wts1lvq1;

wts2lvq1=netlvq1.LW{2,1}; % the weights of the SECOND layer
of LVQ
```

```
save wts2lvq1.mat wts2lvq1; % line 200

wts1lvq2=netlvq2.IW{1,1};
save wts1lvq2.mat wts1lvq2;

wts2lvq2=netlvq2.LW{2,1}; % the weights of the SECOND layer
of LVQ
save wts2lvq2.mat wts2lvq2; % line 200

% The Images with their Competitive features
load ilcs.mat;

for zz=1:nn % each image is read-in in order to find the
features extracted by the competitive neural network
      cd D:\Thesis_Images
     zzz=['ilc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(ilcs{zz})/255;
     xx=sim(netcomp,db);% the vectors from the training set
created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
       end
       bb=['fe',int2str(w),'ilccomp',int2str(zz),'.tif'];
       cd C:\MATLAB_SR11\work\ilcsfecomp;
       imwrite(B,bb);
       clear B;
    end
  end

clear ilcs

load idcs.mat;
```

```
  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['idc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(idcs{zz})/255;
     xx=sim(netcomp,db);% the vectors from the training set
created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w  % line 250
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end
        bb=['fe',int2str(w),'idccomp',int2str(zz),'.tif'];%
the feature # 1st, then the netrk number (by epoch), then
the image #
        cd C:\MATLAB_SR11\work\idcsfecomp;
        imwrite(B,bb);
        clear B;
      end
   end

clear idcs

load fas.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fa' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fas{zz})/255;
     xx=sim(netcomp,db);% the vectors from the training set
created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
         if yy(x)==w
            for z=0:a
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
               end
         end
        end
        bb=['fe',int2str(w),'facomp',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\fasfecomp;
     imwrite(B,bb);
     clear B;
    end
  end

clear fas

load fcs.mat;
% line 300
for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fcs{zz})/255;
     xx=sim(netcomp,db);% the vectors from the training set
created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
         if yy(x)==w
```

```
            for z=0:a
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
               end
        end
        end
        bb=['fe',int2str(w),'fccomp',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\fcsfecomp;
     imwrite(B,bb);
     clear B;
    end
   end
   clear netcomp;
   clear fcs;

% The Images with their LVQ1 features( benign vs malignant)

netlvq1sofm=newsom(CC,[fe1]);
load wts1lvq1.mat;
netlvq1sofm.IW{1,1}=wts1lvq1;

load ilcs.mat;% start with the ILC image data

for zz=1:nn % each image is read-in in order
      cd D:\Thesis_Images
     zzz=['ilc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(ilcs{zz})/255;
     xx=sim(netlvq1sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1         % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255; % line 350
        for x=1:yy2(:,2)
        if yy(x)==w
            for z=0:a
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
               end
        end
        end

bb=['fe',int2str(w),'ilcnetlvq1sofm',int2str(zz),'.tif'];%
the feature # 1st, then the netrk number (by epoch), then
the image #
        cd C:\MATLAB_SR11\work\ilcsfelvq1sofm;
        imwrite(B,bb);
        clear B;
     end
  end

clear ilcs

load idcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['idc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(idcs{zz})/255;
     xx=sim(netlvq1sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end

bb=['fe',int2str(w),'idcnetlvq1sofm',int2str(zz),'.tif'];%
the feature # 1st, then the netrk number (by epoch), then
the image #
        cd C:\MATLAB_SR11\work\idcsfelvq1sofm;
        imwrite(B,bb);
        clear B;
```

```
            end
         end
      clear idcs
      % line 400
      load fas.mat;

        for zz=1:nn % each image is read-in in order

            cd D:\Thesis_Images
           zzz=['fa' num2str(zz) '.tif'];
            A=imread(zzz);
            cd C:\MATLAB_SR11\work;
            db=double(fas{zz})/255;
           xx=sim(netlvq1sofm,db);% the vectors from the training
      set created in "brtr" are classified
           clear db;
           yy=vec2ind(xx);   % each vector is assigned to a
      feature in this row vector
           yy2=size(yy);
            for w=1:fe1        % if the vector matches a feature,
      it is set to zero(black)
              B=double(A(:,:,1))/255;
              for x=1:yy2(:,2)
               if yy(x)==w
                  for z=0:a
                   for aa=0:b
                         B(q(x,1)+z,q(x,2)+aa)=0;
                   end
                     end
               end
              end

      bb=['fe',int2str(w),'fanetlvq1sofm',int2str(zz),'.tif'];%
      the feature # 1st, then the netrk number (by epoch), then
      the image #
              cd C:\MATLAB_SR11\work\fasfelvq1sofm;
           imwrite(B,bb);
           clear B;
          end
        end

      clear fas
      load fcs.mat;

        for zz=1:nn % each image is read-in in order

            cd D:\Thesis_Images
```

```
     zzz=['fc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fcs{zz})/255;
     xx=sim(netlvq1sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a % line 450
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end

bb=['fe',int2str(w),'fcnetlvq1sofm',int2str(zz),'.tif'];%
the feature # 1st then the image #
        cd C:\MATLAB_SR11\work\fcsfelvq1sofm;
     imwrite(B,bb);
     clear B;
    end
 end
 clear fcs.mat;

 % The Images with their LVQ2 features( specific diagnosis)

netlvq2sofm=newsom(CC,[fe1]);
load wts1lvq2.mat;
netlvq2sofm.IW{1,1}=wts1lvq2;

load ilcs.mat;% start with the ILC image data

for zz=1:nn % each image is read-in in order
      cd D:\Thesis_Images
     zzz=['ilc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(ilcs{zz})/255;
     xx=sim(netlvq2sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255; % line 350
        for x=1:yy2(:,2)
         if yy(x)==w
            for z=0:a
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
               end
         end
        end

bb=['fe',int2str(w),'ilcnetlvq2sofm',int2str(zz),'.tif'];%
the feature # 1st, then the netrk number (by epoch), then
the image #
        cd C:\MATLAB_SR11\work\ilcsfelvq2sofm;
        imwrite(B,bb);
        clear B;
     end
  end

clear ilcs

load idcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['idc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(idcs{zz})/255;
     xx=sim(netlvq2sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
```

```
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
         if yy(x)==w
            for z=0:a
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
               end
         end
        end

bb=['fe',int2str(w),'idcnetlvq2sofm',int2str(zz),'.tif'];%
the feature # 1st, then the netrk number (by epoch), then
the image #
        cd C:\MATLAB_SR11\work\idcsfelvq2sofm;
        imwrite(B,bb);
        clear B;
      end
   end
clear idcs

load fas.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fa' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fas{zz})/255;
     xx=sim(netlvq2sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end

bb=['fe',int2str(w),'fanetlvq2sofm',int2str(zz),'.tif'];%
the feature # 1st, then the netrk number (by epoch), then
the image #
        cd C:\MATLAB_SR11\work\fasfelvq2sofm;
     imwrite(B,bb);
     clear B;
    end
  end

clear fas
load fcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fcs{zz})/255;
     xx=sim(netlvq2sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
         if yy(x)==w
            for z=0:a % line 450
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
               end
         end
        end

bb=['fe',int2str(w),'fcnetlvq2sofm',int2str(zz),'.tif'];%
the feature # 1st then the image #
        cd C:\MATLAB_SR11\work\fcsfelvq2sofm;
     imwrite(B,bb);
     clear B;
    end
 end
```

```
% This program is intended to be able to use competitive
% neural networks of several different types to flexibly
% study different numbers of features of different sizes
% in images of breast disease

addpath D:\Thesis_Images\fa85-20095
addpath D:\Thesis_Images\fa86-2107
addpath D:\Thesis_Images\fc85-10830
addpath D:\Thesis_Images\fc85-30689
addpath D:\Thesis_Images\idc86-22790
addpath D:\Thesis_Images\idc86-37485
addpath D:\Thesis_Images\ilc84-35370
addpath D:\Thesis_Images\ilc86-23410
addpath C:\MATLAB_SR11\work\ilcdata
addpath C:\MATLAB_SR11\work\idcdata
addpath C:\MATLAB_SR11\work\fadata
addpath C:\MATLAB_SR11\work\fcdata

% create training sets for ILC,IDC, FA and FC

a=9;  % GLOBAL VARIABLE the row length of the mask
b=9;  % GLOBAL VARIABLE the column length of the mask
n=10;  % the incrementation of the reading frame
nn=20; % GLOBAL VARIABLE the number of images to be read-in
for each diagnosis

q=[];
for f=1:n:1260-a % generates the first indices of the
location of the square sampling
  for d=1:n:960-b
     p=[d f];
     q=[q;p];
  end;
end;

%1. The ILC training sets

for z=1:nn
   cd D:\Thesis_Images;
   zzz=['ilc' num2str(z) '.tif'];
   A=imread(zzz);
   cd C:\MATLAB_SR11\work\ilcdata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
  ilcs{z}=E;  % the square pixel blocks are saved for each
image in a cell array
end

 trilc=cat(2,ilcs{:});
 rsz=size(trilc);
 ilcx=ones(1,rsz(:,2));% the target vector for ILC for
training LVQ
 trilc=[ilcx;trilc];
 clear ilcx;
 save ilcs.mat ilcs;
 clear ilcs;
 save trilc.mat trilc;

 %2. The IDC training sets

for z=1:nn   % line 50
   cd D:\Thesis_Images;
   zzz=['idc' num2str(z) '.tif'];
   A=imread(zzz);
   cd C:\MATLAB_SR11\work\idcdata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
```

```
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
  idcs{z}=E;  % the square pixel blocks are saved for each
image in a cell array
end

 tridc=cat(2,idcs{:});
 rsz=size(tridc);
 idcx=ones(1,rsz(:,2))*2;% the target vector for ILC for
training LVQ
 tridc=[idcx;tridc];
 clear ilcx;
 save idcs.mat idcs;
 clear idcs;
 save tridc.mat tridc;

 %3. The FA training sets

for z=1:nn
   cd D:\Thesis_Images;
   zzz=['fa' num2str(z) '.tif'];
   A=imread(zzz);
   cd C:\MATLAB_SR11\work\fadata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
  fas{z}=E;  %line 100. the square pixel blocks are saved
for each image in a cell array
end
```

```
 trfa=cat(2,fas{:});
 rsz=size(trfa);
 fax=ones(1,rsz(:,2))*3;% the target vector for ILC for
training LVQ
 trfa=[fax;trfa];
 clear fax;
 save fas.mat fas;
 clear fas;
 save trfa.mat trfa;

 %4. The FC training sets

for z=1:nn
   cd D:\Thesis_Images;
   zzz=['fc' num2str(z) '.tif'];
   A=imread(zzz);
   cd C:\MATLAB_SR11\work\fcdata;
   A=A(:,:,1);
   g=size(A);
   h=g(1)-a;
   j=g(2)-b;
   D=cell(fix(g(1)/n-1)*fix(g(2)/n-1),1); %the size of D is
set in relation to the size of the image and the sampling
rate
   m=0;
   for e=1:n:j   % reading along the row "a" pixels at a
time(column-by-column)
      for d=1:n:h % reading along the column "b" pixels at
a time (row-by-row);
        m=m+1;
        D{m}=reshape(A(d:d+a,e:e+b),(a+1)*(b+1),1);
     end;
  end;
  E=cat(2,D{:});
  fcs{z}=E;  % the square pixel blocks are saved for each
image in a cell array
end

 trfc=cat(2,fcs{:});
 rsz=size(trfc);
 fcx=ones(1,rsz(:,2))*4;% the target vector for ILC for
training LVQ
 trfc=[fcx;trfc];
 clear fcx;
 save fcs.mat fcs;
 clear fcs;
 save trfc.mat trfc;
 % The Master Training Set

 trmaster=[trilc tridc trfa trfc];% the master training
set. note: the first row is the LVQ target vector
 clear trilc tridc trfa trfc;
 rsz=size(trmaster);
 trmaster=trmaster(:,randperm(rsz(:,2)));
 %save trmaster.mat trmaster;
% line 150
% using square pixel blocks sampled from ALL images, a
competitive neural network
% and a LVQ neural network will be trained simultaineously

fe1=24; %number of features for the 1D-SOFM, LVQ and
Competitive neural networks

DD=(a+1)*(b+1); %the training vectors are 1-D vectors made
from square pixel blocks

trmaster=trmaster(2:end,:);% the training set(randomized)
is separated.
AA=ones(DD,1)*double(min(min(trmaster)))/255;% the lower
limit of all pixel values
BB=ones(DD,1)*double(max(max(trmaster)))/255;% the upper
limit of all pixel values
CC=[AA BB];
clear AA;
clear BB;

% The SOFMs are now created and trained
fe2=5; % the number of feaures for each dimension of
% the 2-D SOFM network
fe3=3; % the number of features for each dimension of
% the 3-D SOFM network
prt=300000; % the size of the submatrix of "trmaster"
training set to be
% used. Just the about 1/6 of trmaster is used to train,
due to RAM limitations

net1sofm=newsom(CC,[fe1]);
net2sofm=newsom(CC,[fe2 fe2]);
net3sofm=newsom(CC,[fe3 fe3 fe3]);

trmast=double(trmaster(:,1:prt))/255;
clear trmaster;
net1sofm=train(net1sofm,trmast);
```

```
net2sofm=train(net2sofm,trmast);
net3sofm=train(net3sofm,trmast);

wts1sofm=net1sofm.IW{1,1};
save wts1sofm.mat wts1sofm;

wts2sofm=net2sofm.IW{1,1};
save wts2sofm.mat wts2sofm;

wts3sofm=net3sofm.IW{1,1};
save wts3sofm.mat wts3sofm;

save net1sofm.mat net1sofm;

save net2sofm.mat net2sofm;

save net3sofm.mat net3sofm;

load ilcs.mat;% start with the ILC image data

for zz=1:nn %each image is read-in in order
      cd D:\Thesis_Images
     zzz=['ilc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(ilcs{zz})/255;
     xx=sim(net1sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1       % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end

bb=['fe',int2str(w),'ilcnet1sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\ilcsfe1sofm;
        imwrite(B,bb);
        clear B;
     end
  end

clear ilcs

load idcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['idc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(idcs{zz})/255;
     xx=sim(net1sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end % line 550
              end
        end
        end

bb=['fe',int2str(w),'idcnet1sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\idcsfe1sofm;
        imwrite(B,bb);
        clear B;
      end
   end
clear idcs

load fas.mat;

  for zz=1:nn % each image is read-in in order
```

```
      cd D:\Thesis_Images
     zzz=['fa' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fas{zz})/255;
     xx=sim(net1sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
       end

bb=['fe',int2str(w),'fanet1sofm',int2str(zz),'.tif'];% the
feature # 1st, then the netrk number (by epoch), then the
image #
        cd C:\MATLAB_SR11\work\fasfe1sofm;
     imwrite(B,bb);
     clear B;
    end
  end

clear fas
load fcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fcs{zz})/255;
     xx=sim(net1sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe1        % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
         if yy(x)==w
             for z=0:a
              for aa=0:b
                    B(q(x,1)+z,q(x,2)+aa)=0;
              end
                end
          end
         end

bb=['fe',int2str(w),'fcnet1sofm',int2str(zz),'.tif'];% the
feature # 1st then the image #
        cd C:\MATLAB_SR11\work\fcsfe1sofm;
     imwrite(B,bb);
     clear B;
    end
 end

fe=fe2^2;

load net2sofm.mat;% test the 2-D SOFM
load ilcs.mat;% start with the ILC image data

for zz=1:nn % each image is read-in in order
      cd D:\Thesis_Images
     zzz=['ilc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(ilcs{zz})/255;
     xx=sim(net2sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe         % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
         if yy(x)==w
            for z=0:a
```

```
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
              end
         end
        end

bb=['fe',int2str(w),'ilcnet2sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\ilcsfe2sofm;
        imwrite(B,bb);
        clear B;
     end
  end

clear ilcs

load idcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['idc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:
      db=double(idcs{zz})/255;
     xx=sim(net2sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
       end

bb=['fe',int2str(w),'idcnet2sofm',int2str(zz),'.tif'];% the
feature # 1st, then the netrk number (by epoch), then the
image #
```

```
        cd C:\MATLAB_SR11\work\idcsfe2sofm;
        imwrite(B,bb);
        clear B;
      end
   end
clear idcs

load fas.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fa' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fas{zz})/255;
     xx=sim(net2sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end

bb=['fe',int2str(w),'fanet2sofm',int2str(zz),'.tif'];% the
feature # 1st, then the netrk number (by epoch), then the
image #
        cd C:\MATLAB_SR11\work\fasfe2sofm;
     imwrite(B,bb);
     clear B;
    end
  end

clear fas
load fcs.mat;
```

```
  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:
      db=double(fcs{zz})/255;
     xx=sim(net2sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
      for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
        B=double(A(:,:,1))/255;
        for x=1:yy2(:,2)
        if yy(x)==w
            for z=0:a
             for aa=0:b
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
              end
         end
        end

bb=['fe',int2str(w),'fcnet2sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\fcsfe2sofm;
     imwrite(B,bb);
     clear B;
    end
 end

fe=fe3^3;

load net3sofm.mat;% test the 3-D SOFM
load ilcs.mat;% start with the ILC image data

for zz=1:nn % each image is read-in in order
      cd D:\Thesis_Images
     zzz=['ilc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(ilcs{zz})/255;
     xx=sim(net3sofm,db);% the vectors from the training
set created in earlier are classified
     clear db;
     yy=vec2ind(xx);    % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
         end
        end

bb=['fe',int2str(w),'ilcnet3sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\ilcsfe3sofm;
        imwrite(B,bb);
        clear B;
     end
  end

clear ilcs

load idcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['idc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(idcs{zz})/255;
     xx=sim(net3sofm,db);% the vectors from the training
set created in "brtr" are classified
     clear db;
     yy=vec2ind(xx);    % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
```

```
                   B(q(x,1)+z,q(x,2)+aa)=0;
             end
              end
         end
        end

bb=['fe',int2str(w),'idcnet3sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\idcsfe3sofm;
        imwrite(B,bb);
        clear B;
      end
   end
clear idcs

load fas.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fa' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fas{zz})/255;
     xx=sim(net3sofm,db);% the vectors from the training
set created in earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
        end

bb=['fe',int2str(w),'fanet3sofm',int2str(zz),'.tif'];
        cd C:\MATLAB_SR11\work\fasfe3sofm;
     imwrite(B,bb);
     clear B;
    end
  end

clear fas
load fcs.mat;

  for zz=1:nn % each image is read-in in order

      cd D:\Thesis_Images
     zzz=['fc' num2str(zz) '.tif'];
      A=imread(zzz);
      cd C:\MATLAB_SR11\work;
      db=double(fcs{zz})/255;
     xx=sim(net3sofm,db);% the vectors from the training
set created earlier are classified
     clear db;
     yy=vec2ind(xx);   % each vector is assigned to a
feature in this row vector
     yy2=size(yy);
     for w=1:fe        % if the vector matches a feature,
it is set to zero(black)
       B=double(A(:,:,1))/255;
       for x=1:yy2(:,2)
        if yy(x)==w
           for z=0:a
            for aa=0:b
                  B(q(x,1)+z,q(x,2)+aa)=0;
            end
              end
        end
       end

bb=['fe',int2str(w),'fcnet3sofm',int2str(zz),'.tif'];% the
feature # 1st then the image #
       cd C:\MATLAB_SR11\work\fcsfe3sofm;
    imwrite(B,bb);
    clear B;
   end
 end
```