İzmir Institute of Technology

The Graduate School of Engineering and Sciences

IMAGE CLASSIFICATION BY MEANS OF PATTERN RECOGNITION TECHNIQUES

A Thesis in

Computer Engineering

by Cumhur GÜZEL

Submitted in Partial Fulfillment of the Requirements
for the Degree of

Master of Science

June 1997

We approve the thesis of **Cumhur GÜZEL**

Date of Signature

09/07/1997

Prof. Dr. Halis Püskülcü
Thesis Advisor
Department of Computer Engineering

09/07/1997

Prof. Dr. Sıtkı Aytaç
Committee Member
Chairman of Department of Computer Engineering

09/07/1997

Prof. Dr. Tanju Aktuğ
Committee Member
D.E.Ü, Faculty of Medicine, Department of Pediatric Surgery

Acknowledgments

I wish to express my sincerely gratitude to

- my advisor, Prof. Dr. Halis Püskülcü, for his constant support, and his assistance in the course of the research and investigation,
- Prof. Dr. Sıtkı Aytaç who establishes the contact of Dokuz Eylül University Hospital, and his assistance in the course of the research and investigation,
- Prof. Dr. Tanju Aktuğ, for his assistance in obtaining VUR images and providing detailed information on VUR disease,
- my family, for their patience and love,
- H.Cüneyt Aka , for his cooperation in performing PROVISION Software and other tasks
- my friends in IIT for their helps and clues,
- Memet Uludağ and Mete Eminağaoğlu, for their helps in the edition of the thesis,
- Bahattin Tayanç who provides us necessary software and hardware tools and service,

In addition, I wish to acknowledge to the following organizations for their support.

- Computer Center of İzmir Institute of Technology, for providing necessary computing environments and tools.
- Dokuz Eylül University Faculty of Medicine, Department of Pediatric Surgery, for their valuable inputs utilized in the thesis.
- Biomed-3 Organization Committee, for motivation by awarding us for the presentation performed in Biomed-3 Conference.

Abstract

Image classification plays an important role in many computer vision tasks such as surface inspection, shape determination etc. Various 2-D image classification techniques are investigated, assessed and a computational method to classify the 2-D X-ray images is developed and evaluated in this study. Various pattern recognition techniques are devised for the solution of the image classification. Those techniques may be divided into mainly two groups. First one, is mathematical and statistical model based, second one, is the artificial neural network based techniques. We have concentrated on artificial neural network techniques. In the experiments, both techniques were applied for the classification of the VUR (vesico ureteral reflux) images, in this study. However, according to the experiments performed on VUR case study, neural network technique was more successful than others, in terms of classifier. A hybrid method is proposed in this study, rather than pure artificial neural network solution. Representation of images is performed via transformation invariant mathematical structure called Fourier Descriptors and these structures are used as input to train the neural network for the classification part.

The application is performed as follows: Feature extraction is performed first, then extracted features are used as pattern vectors for training the neural network. Representation of the shapes in X-ray images is performed by using Fourier Descriptors. Usage of Fourier descriptors as a method of representation of the shapes, provides the transformation invariant (translation, rotation, scaling invariant structure) representation of X-ray images. These new vector representation is fed to the neural network. Backpropagation is used as a training algorithm. After training is finished, system is ready for questioning. The minimum-mean-distance and nearest neighbor rules are also applied for the pattern vectors generated for the experiments. But the multilayer perceptron trained by backpropagation outperforms both of these statistical classifiers.

Öz

Görüntü sınıflandırma işlemi, yüzey tanımlama, yüzey inceleme, görüntü analizi, uzaktan algılama, şekil tanıma gibi birçok bilgisayarla görü uygulamalarında önemli rol oynamaktadır. Bu çalışmada, tıbbi görüntülerin sınıflandırılması için, değişik metodlar incelenmiş, bir 2-B görüntü tanıma yöntemi önerilmiş ve değerlendirilmiştir. Çeşitli örüntü tanıma teknikleri, görüntü sınıflandırma sorununu çözmek için geliştirilmiştir. Bu örüntü tanıma teknikleri ana olarak iki gruba ayrılabilir. Bunlardan ilki matematiksel ve istatistiksel tabanlı olanlar, ikincisi ise, yapay sinir ağları tabanlı olanlardır. Biz bu çalışmada sinir ağı tabanlı modellere ağırlık vermiş bulunuyoruz. Bu çalışmada, her iki gruba düşen teknikler, deneysel çalışma aşamasında, VUR görüntülerinin tanınması üzerinde uygulanmıştır. Sınıflandırıcı başarım düzeyini göz önüne aldığımızda, yapay sinir ağlarıyla elde ettiğimiz sınıfladırıcı, diğer istatistiksel sınıflandırılardan daha başarılı sonuçlar vermiştir. Bu çalışmada melez bir yöntem önerilmiştir. Görüntülerin temsili, dönüşümden bağımsız matematiksel bir yapı olan Fourier tanımlayıcılarıyla gerçekleştirilmiştir. Fourier tanımlayıcıları yapay sinir ağını eğitmede girdi olarak kullanılmıştır.

Uygulama aşağıda verildiği gibi yapılmıştır. Öncelikle, özellik çıkarımı gerçekleştirilmiş ve elde edilen özellikler tasarımı yapılan yapay sinir ağının eğitiminde kullanılmıştır. Şekillerin temsili Fourier tanımlayıcılarının vektörel forma dönüştürülmesiyle gerçekleştirilmiştir. Fourier tanımlayıcılarının kullanımı, dönüşümden bağımsız (Yer değiştirmeden, dönmeden, ölçeklemeden bağımsız) olarak şekillerin temsilini sağlamıştır. Bu yeni vektörel tanımlama yapay sinir ağına girdi olarak sunulmuştur. Geri yayınım algoritması, yapay sinir ağının eğitiminde kullanılmıştır. Eğitim tamamlandıktan sonra, sistem sorgulamaya hazır hale gelmektedir. Gerçekleştirilen deneysel çalışmalarda en yakın komşuluk kuralı ve en yakın ortalama uzaklık kuralı sınıflandırıcıları da ayrıca üretilen örüntü vektörlerinin tanınmasında uygulanmıştır. Ancak geri yayınım algoritmasıyla eğitilen çok katmanlı perceptrondan oluşturulan sınıflandırıcı ile , yukarıda bahsedilen istatistiksel sınıfladırıcılardan daha başarılı sonuç elde edilmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Motivation

Image classification plays an important role in computer vision tasks. Various 2-D image classification techniques are investigated and a computational method to classify the 2-D X-ray images is proposed and evaluated in this study. Image classification techniques are mainly classified in two groups: First one is mathematical-statistical model based techniques, second one is artificial neural network based techniques. We have concentrated on artificial neural network techniques in this project and developed a solution to the 2-D X-ray image classification problem, for VUR disease.

In this study, we have considered one of the sub-field of the machine vision which is known as image classification. The study covers detailed investigation and examination, on methods for 2-D image classification via pattern recognition techniques and as a case study, classification (grading)of vesico ureteral reflux X-ray images is performed.

VUR grading is performed in five levels as given in [Aktuğ 1995] Even physicians have difficulties in choosing the grade of the disease objectively. We have utilized high level computer vision and artificial neural network techniques in order to help the X-ray physician to classify the X-ray images. We have developed an original computational method for grading of the VUR disease X-ray images in this study. The design and implementation details of the developed method is given in the last chapter.

1.2 Introduction to Computer Vision

The goal of the computer vision system is to create a model of real world from images. "A machine vision system recovers useful information about a scene from its two-dimensional projections of three dimensional world" [Jain 1995]. Since images are two-dimensional projections of the three-dimensional world, the information is not directly available and must be obtained via processes (sampling, etc.). To recover the

information, knowledge about the objects in the scene and projection geometry is required.

Machine vision systems help physicians in recognition and reconstruction of the images. Quantitative measurements on regions of interest can also be made easily available. Such systems are being developed for all imaging modes useful in different aspects of health care. Similar applications are being developed for inspection of industrial, agricultural, and other products. Machine vision systems have been used for quality control of products ranging from pizza to turbine blades, from sub-micron structures on wafers to auto-body panels, and from apples to oranges. The examples on object recognition are given in [Shridhar and Badreldin 1984] and [Kim and Nam 1995].

Here we briefly summarize the closely related fields about computer vision. No effort is made to relate machine vision (also called computer vision) to other fields exhaustively.

Image processing is a well-developed field. Image processing techniques usually transform images into other images; the information recovery is left to a human user. This field includes topics such as image enhancement, image compression, and correcting blurred or out-of-focus images. On the other hand, machine vision algorithms take images as inputs but produce other types of outputs, such as representation for the object contours in an image. Thus, emphasis in machine vision is on recovering information automatically and recognition of objects, with minimal interaction with a human. Image processing algorithms are useful in early stages of a machine vision system. They are usually used to enhance particular information and suppress noise.

Computer graphics generates images from geometric primitives such as lines, circles, and free-form surfaces. Computer graphics techniques play a significant role in visualization and virtual reality. Machine vision is the inverse problem: estimating the geometric primitives and other features from the image. Thus, computer graphics is the synthesis of images, machine vision is the analysis of the images. In the early days of these two fields, there was not much relationship between them, but in the recent years these two fields have been getting closer to each other. Machine vision is using curve and surface representations and several other techniques from computer graphics, and computer graphics is using many techniques from machine vision to enter models into the

computer for creating realistic images. Visualization and virtual reality are bringing these two fields closer.

Pattern recognition classifies numerical and symbolic data. Many statistical and syntactical techniques have been developed for classification of patterns for classification of patterns. Techniques from pattern recognition play an important role in machine vision for recognizing objects. In fact, many industrial applications rely on heavily pattern recognition.

Artificial intelligence is concerned with designing systems that are intelligent and with studying computational aspects of intelligence. Artificial intelligence is used to analyze scenes by computing a symbolic representation of the scene contents after the images have been processed to obtain features. Artificial intelligence may be viewed as having three stages: perception, cognition, and action. Perception translates signals from the world into symbols, cognition manipulates symbols, and action translates symbols into signals that effect changes in the world. Many techniques from artificial intelligence play important roles in all aspects of computer vision. In fact, computer vision is often considered a sub-field of artificial intelligence.

Design and analysis of neural networks has become a very active field in the last decade. Neural Networks are being increasingly applied to solve machine vision problems.

1.3 Artificial Neural Networks and Computer Vision

This section focuses on the shape recognition problem of computer vision. Shape recognition, has many important industrial, biomedical [Ghorbel and Tocnaye 1990], and [Shen et. al. 1994]

Recently neural networks have also been applied to this problem. It is generally assumed that the underlying application need not be modeled very much and that an artificial neural network solution can be obtained instead, by training from empirical data with little or no a priori information about the application. However, a right network architecture is fundamental to a good solution. Also questions like what constitutes a sufficient training example or what is a suitable network architecture have not been

answered satisfactorily. For uses in critical applications, however, any learning system has to clearly demonstrate and answer questions like : what is being learned, how it is learned, what part of the data is used to learn, how to represent what has been learned, how well and how efficient the learning takes place and what are the evaluation criteria at hand.

The goal of Computer Vision research is to provide computers with human-like perception capabilities so that they can sense the environment, understand the sensed data, take appropriate actions and learn from this experience in order to enhance future performance. However, much of the current models and methodologies do not scale up well for real world problems. Therefore, the current state-of-the-art in Computer Vision needs significant improvements to deal with real world applications such as navigation, target recognition, manufacturing, photo-interpretation, remote sensing etc. Machine learning offers effective methods for computer vision for automating the model/concept acquisition and updating processes, adapting task parameters and representations and using experience for generating, verifying and modifying hypotheses. The five key machine learning paradigms currently available are : inductive learning, analytic learning, case-based learning, genetic algorithms and neural networks. Amongst the above, neural networks is currently the most popular learning paradigm. The multilayer feed-forward neural network is especially popular because it can learn any non-linear mapping which increases its usage in the pattern recognition area.

1.4 Contributions and Organizations

We present an integrated approach in recognition and classification of 2-D X-ray images in this study. We have utilized high level computer vision and artificial neural network techniques in order to solve the X-ray image classification problem. To help the physician in diagnosis of vesico ureteral reflux disease, classification of X-ray images has been chosen as a subject of this study. We have developed an original computational method for grading(classification) of the VUR disease phases.

The validity of our method has been examined by extensive experimentation on real applications. (VUR example images taken from Dokuz Eylül University Hospital). Our training set is consisted of 100 images which are obtained from Dokuz Eylül University and we have applied some transformations and added noise to these images. Recognition percentages, and detailed experimental results, are given in the last chapter.

Additionally, as an application software, we have developed a Motif based X-Windows software for classification of X-ray images. It may be available for scientific and educational use in this area.

We begin by brief introduction to subject covered in this study in first chapter. Second chapter is a close examination on the pattern recognition approach, then it continues with artificial neural network basics, and then 2-D shape recognition via artificial neural networks is summarized in chapter 4.

Shape representation methods is given in the fifth chapter. Then it continues with MLP (multilayer perceptron) and backpropagation algorithm, respectively which forms the corner stones of our subsequent works in image classification. Modular approach is considered in the solution of the image classification. The problem is considered in two phases. First phase is shape representation models (explained in detail in chapter 5), and the second one is neural network classifier (in chapter 6).

In chapter 7, we proceed to formally develop the system for the solution of recognition of 2-D VUR X-ray images. At the end, we summarized the results, conclusion and planned future works.

In appendix parts of the thesis, the interface of the program developed as a solution to the VUR is given and samples of reconstructed VUR X-ray images are given.

Chapter 2

INTRODUCTION TO PATTERN RECOGNITION

2.1 Introduction

Pattern recognition, artificial intelligence and neurocomputing have a common history that has been started in the 1950s. Contributions to pattern recognition are now being made by a variety of disciplines including artificial intelligence, cognitive science, computer engineering, neurobiology, philosophy, and psychology. These advances are coordinated to some extent under the neural networks research, but a wider coordination would be helpful. Knowledge of pattern recognition is important because the occurrences of human existence seem to take form in the form of patterns as given in [Pao 1989]. The formation of language, the utterance of speech, the drawing of pictures and understanding of images all involve patterns. No one aspects, nor any sequence of aspects, of a pattern determine the significance of that pattern. Instead, there is a great deal of compensation among all aspects of the pattern, so the significance of a pattern can be grasped only if all aspects are available simultaneously for consideration.

Pattern recognition is not only important in human perception, but also in cognition. Human being, assess situations in terms of the pattern of circumstances that constitutes the situation, and they act in accordance with what that pattern might represent or what it might portend. It is useful not only because it is intrinsically an essential part of human behavior, but also because computer-implemented pattern recognition capabilities can enable machines to carry out perception tasks. Knowledge of pattern recognition is often highly specialized. Contributions to the field have come from many different research areas, from various disciplines. This itself is a healthy indication of the breadth and depth of interest in the topic and the vigor of the associated research. An ancillary effect, however, is that investigators often have knowledge that pertains to only their own tightly knit research group [Pao 1989].

In this study we have utilized the neural networks approach in solving the image understanding problem of pattern recognition. Neurocomputing implementations of pattern recognition algorithms [Rumelhart et. al 1986] are interesting not only because they allow iterative procedures to be implemented rapidly, but also- and perhaps

primarily- because a neural networks perspective on information processing stimulates us to be creative in visualizing new approaches. The considerations presented next chapter will be the introduction to the artificial neural networks paradigm, and its applications.

2.2 Motivation for the Study of Adaptive Pattern Recognition

Situations in the pattern recognition world, generally can not be assessed in terms of isolated facts or even in terms of a body of isolated facts [Pao 1989]. Rather, we find that we need to describe situations in terms of patterns of interrelated facts. Sometimes, the interrelation is implicit, in the sense that we know that all those facts pertain to the same object situation. In other cases, a pattern may be meaningful only because of explicit relationships among the various features of the pattern [Pao 1989].

It is interesting to observe that our perceptive powers seem to be well adapted to such pattern-processing tasks. For example, we are as a human, able to recognize images such as handwriting in a robust manner, despite major variations, distortions, or omissions or in summary in any translated or distorted position. A related trait is the ability of humans to retrieve information on the basis of associated cues, consisting of only part of a pattern. A few whistled notes can evolve a memory of an entire tune; a glimpse of the back of a head in a crowd can remind us in detail of an old friend.

This same capability to cope easily of associated items is found not only in perception but also in tasks that clearly involve both perception and cognition. Thus, it seems that nature decreed that information be in pattern format, and humans seemingly adapted well to that circumstance [Pao 1989]. Part of the study of pattern recognition stems from our desire to understand the basis for these powers in humans. There is also a strong engineering-based pragmatic motivation [Pao 1989]. As we learn how to build computers to help us in performance of tasks, we strive to make them "intelligent" so that they might be more compatible with the manner in which we normally behave. We would like to have computer-based machines with perception and cognition capabilities that we ourselves possess. This would enable us to use the machines more easily and would undoubtedly make the machines more efficient in handling real world tasks [Pao 1989].

This second combination of circumstances provides great motivation for understanding how pattern-formatted information might be handled. Traditional artificial

intelligence, psychology, and neurobiology, also undertook activities directed toward that objective.

To understand more clearly the role of adaptive pattern recognition [Pao 1989], we shall look at both traditional artificial intelligence and traditional pattern recognition. These two research disciplines have several common objectives, the most important of which are to understand perceptual and cognitive processes in humans and to implement similar capabilities in machines. The two disciplines had common starting points, but then developed increasingly different interests, different styles, and different areas of emphasis.

The work of traditional artificial intelligence, for example, is based on the representation of hypothesis and resolution of facts. (the idea that the world can be represented symbolically) From this viewpoint, perception and cognitive processes consist of acquiring, manipulating, associating, and modifying symbolic representation [Pao 1989]. The approach seems to be natural for representing certain high level mental functions, but may be inappropriate and perhaps even unnatural for dealing with tasks involving combined perception and cognition. In addition, such high level symbolic representations seem to be far removed from anything we know about processes in biological neural systems.

In contrast to all this, traditional pattern recognition experienced some early disappointments with the perception approach, then proceeded to concentrate on the mathematical or computer-science aspects of pattern-formatted information processing [Pao 1989]. There is, for example emphasis on statistical pattern recognition [Fukunaga 1990], and there is also considerable interest in the use of mathematical linguistics in the classification of patterns with syntactic structures. Use of fuzzy logic provides one link between pattern recognition research and human-like behavior as given in Zadeh [Pao 1989].

Although the two disciplines shared the interesting objectives, both left major categories of problems untouched. It is a matter of fact that valuable progress has occurred, rather ironically, more has been attained in the area of useful practical engineering than in that of "understanding" human perception and cognition.

More recently groups of researchers are experimenting connectionist models or parallel distributed models of computation (PDP) by [Rumelhart et. al 1986]. These models are based on use of large numbers of elemental processors interconnected in

manners reminiscent of human neural networks, and they can exhibit powerful learning, memorization, and associative-recall capabilities for pattern-formatted information. Such subsymbolic-level processing seem to be appropriate for dealing perception tasks and even with tasks that call for combined perception and cognition. Moreover, because the structure and the processing modes of these models seem to be conceptually compatible with what we know of biological neural networks, it is often possible to derive inspiration from neuro-biological or psychological studies, even though the objective might be engineering. By the same token when engineering performance mirrors human performance, it might be that, in fact, similar systems considerations apply in the biological neural networks as in computer neural networks, and mutually useful hints can be obtained in this manner.

Neural networks research gives rise to two important related research goals. One of these is to build bridges of understanding between neural networks computing and symbolic processing. The other is to coordinate, interpret , organize and make coherent the various parts and pieces of pattern recognition expertise that reside in statistical pattern recognition, syntactic pattern recognition, fuzzy logic, and (now) neural networks computing. It is important to organize these pieces in to a related , coherent body of knowledge , so that research will not be wasteful. We need to organize and develop a coherent approach to adaptive pattern recognition, an approach that will recognize the contributions of the past and yet will be compatible with the rapid pace of future progress in neural networks research.

Situation is illustrated schematically in figure 2.1. The objects A and B may each be represented by many different patterns, but their pattern descriptions do not overlap in feature space. The patterns are mapped unambiguously in a many to one manner to class membership designations. For such circumstances, computer-based deterministic procedure for estimating similarities is of prime importance. Such a procedure allows us to distinguish between classes and to generalize within each class. The usual classification rule is that, if pattern X is most similar to patterns of class A then pattern X belongs to class A.

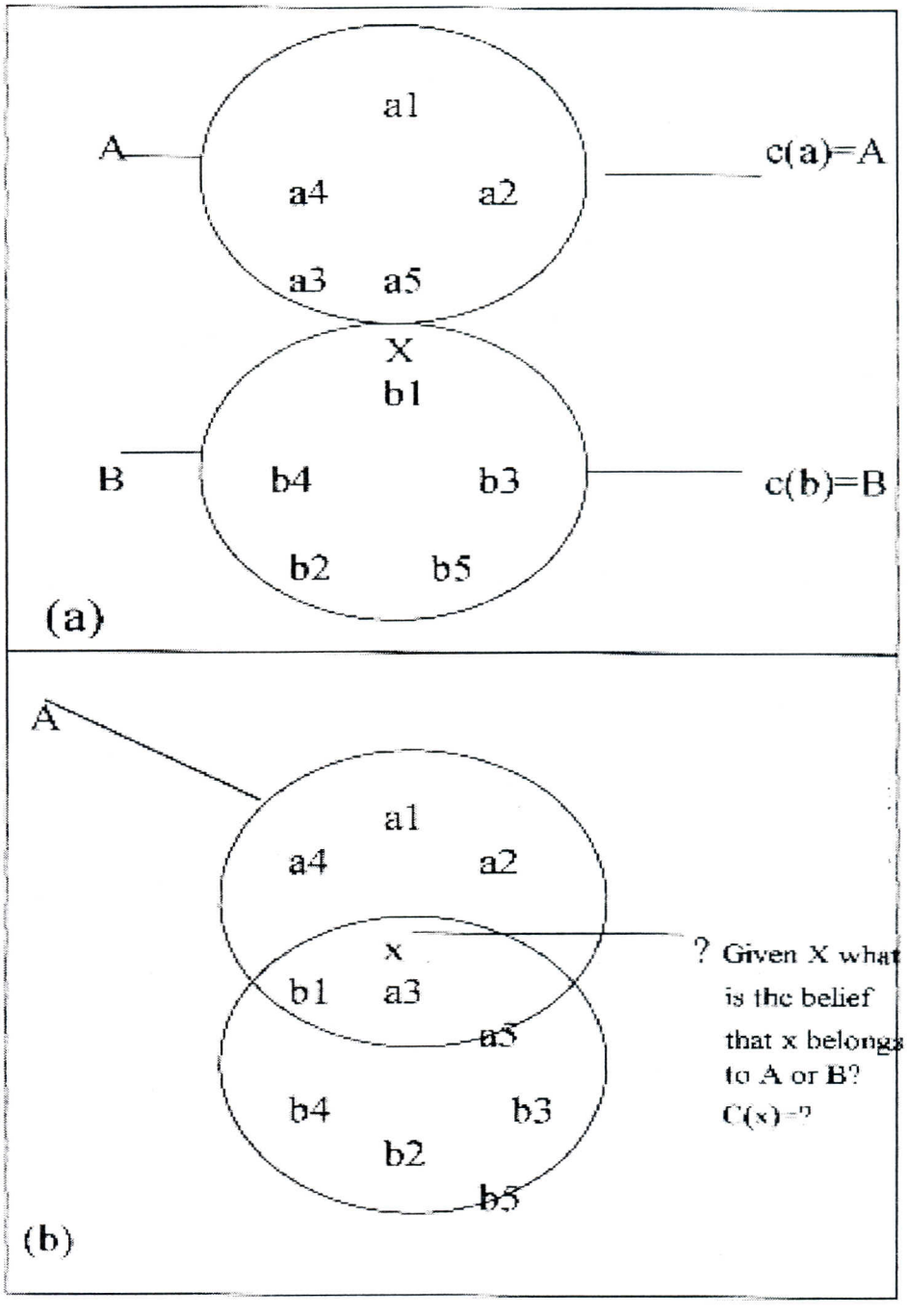


Figure 2.1 Estimating similarities and beliefs [Pao 1989]

a) Generalization through similarities: Is x more similar to a patterns, or b patterns?
b) Estimation of belief that pattern belongs to a certain class, given that is valid for pattern to be representing either of two classes.

In contrast to the deterministic situation illustrated in figure 2.1 another extreme exists where we know that every pattern that we encounter could belong to any one of several previously identified classes. That is, it is just as valid that a pattern X represents object A as it is that X represents object B, and so on, as illustrated in Figure 2.1. But given that we encounter X, what are the weights of the relative beliefs that X represents object A, or object B or any of the other objects?. In this case, the focus is on estimating beliefs rather than on estimating similarities.

In reality, actual pattern recognition tasks lie between the two extremes. Even if the patterns were separable in some feature space, there would still be the troublesome problem of what happens in detail at regional boundaries, where some overlap might occur. Conversely, it is extremely unlikely that complete will occur in all of the feature space for two or more classes of patterns; thus measures of similarity might generally be as useful as are belief estimates.

2.3 General Bibliography on Pattern Recognition

Researchers generally recognize that the science of pattern recognition is concerned with three major issues, these being;

1. The appropriate description of objects, physical or conceptual, in terms of representation space,
2. The specification of an interpretation space,
3. The mapping from representation space to interpretation space.

This abstraction sets the scope and structure for the development of the science of pattern recognition.

The matter of representation was and still is an important issue. If we restrict ourselves to a geometric representations, then statistical methods such as that based on Karhunen-Loéve transformation can indeed determine what combination of feature values provides measures for discriminating between classes and what combinations are largely irrelevant and might be ignored. Such statistical analyses are largely irrelevant, however, if we depart geometric representations and other matters that need to be considered.

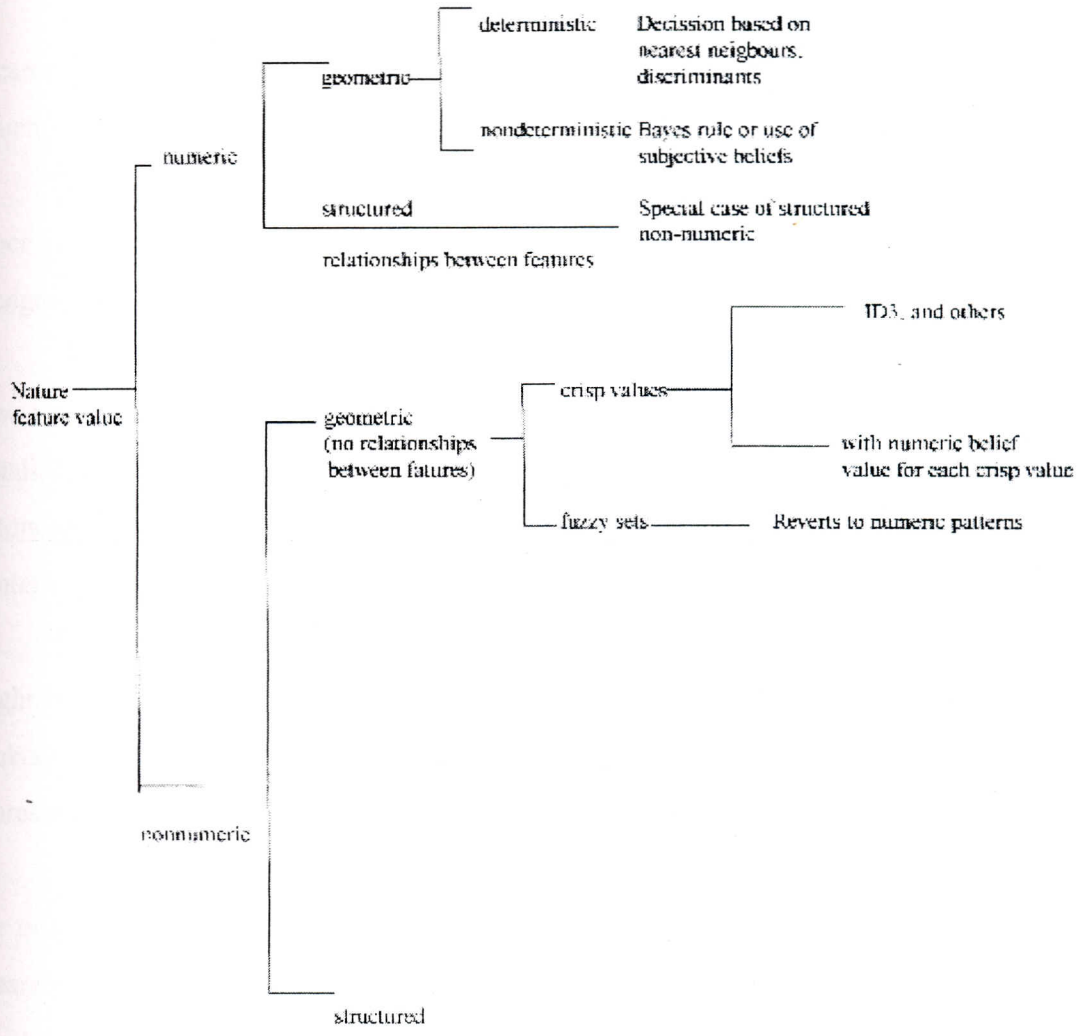


Figure 2.2 Categorization of pattern formats based on nature of feature value. [Pao 1989]

Sometimes, even though some aspects of objects are clearly important, features, it develops that those features are not important in themselves. Rather it is the structural relationships among the features that contain critically important information. That realization gave rise to a large body of literature on syntactic and structural pattern recognition.

Whereas statistical pattern recognition made use of the results of statistical communication and estimation theory as given in [Fukunaga 1990], syntactic pattern

recognition built on earlier work in mathematical linguistics and on the results of research in computer languages.

Syntactic pattern recognition as developed by its principal proponents, however, became limited to "before - after" relationships and could not cope with more general structures.

The purpose of pattern recognition is to carry out a mapping from representation space to interpretation space given in [Pao 1989] and an important aspect of pattern recognition is to determine the operator that will carry out this mapping.

Against this background of gains in research, there still remains the question of how the requisite operators are to be learnt. Certainly, they can be hand-crafted and installed in specific pattern recognition systems, but what are the general principles autonomous inference of such operators or discriminators, even if, for example, we are content to limit ourselves to geometric representations?

[Rumelhart et. al 1986] published their account of how multilayered perceptrons might be trained with use of the generalized delta rule, and new ground is now being broken in the exploration of means for autonomous learning of mappings from representation space to interpretation space.

The actions of the internal layers might be viewed as successive transformations of the original representation until we attain a final representation in which the desired separations can be achieved with use of a hyperplane. In a sense, Rosenblatt [Rumelhart et. al 1986] could have been correct if the initial representation had been chosen with care.

Although it is rarely acknowledged explicitly, estimating the values of attributes is often the principal objective of pattern recognition. That is, often the question is not whether or not x, belong to class A. Rather, the question is what is the value of attribute A of x. It is in this role that neurocomputing will excel.

Chapter 3

ARTIFICIAL NEURAL NETWORKS APPROACH

3.1 Global View

We can define the artificial neural networks as synthetic networks that emulate the biological neural networks found in living organisms. Simply a class of algorithms, produce solution to a number of specific problems. Artificial neural networks (ANNs), also called artificial neural systems (ANSs), neurocomputers, parallel distributed processors (PDPs) or connectionist models are an attempt to mimic (simulate) at least partially the structure and functions of brains and nervous systems of living creatures. Generally speaking, an artificial neural networks is an information or signal processing system composed of a large number of simple processing elements, called artificial neurons or simply nodes, which are interconnected by direct links called connections and which cooperate to perform parallel distributed processing (PDP) in order to solve desired computational task. One of the attractive features of ANNs is their capability to adopt themselves to special environmental conditions by changing their connection strengths or structure. Artificial neural networks are sometimes considered as grossly simplified models of the human brain as given in [Lippman 1992]. This view is somewhat exaggerated and misleading because the human brain is not well understood and indeed its behavior is very complex. It may be more reasonable to compare artificial neural network capabilities to simpler nervous systems of primitive animals such as insects which have the ability to adopt themselves to a complex environment.

3.2 Basic Features and Classifications of ANNs

A wide range of models of artificial neural networks has been developed for a variety of purposes; they differ in structure, implementation and principle of operation, but share common features. Generally speaking, artificial neural networks are computing systems made up of a number of simple highly interconnected signal or information

processing units (artificial neurons) with the following features as defined in [Cichocki and Unbehauen 1994]:

(i) Processing of information and memory are distributed among the whole structure, and therefore it is difficult to separate the hardware and software in the structure. (this feature is in contrast to the traditional computing systems with low-connectivity networks with few complex processors and local processing information). In fact, neural networks are trained, rather than programmed the given task.

(ii) Artificial neurons are highly interconnected in such way that the state of one neuron affects the potential of the large number of neurons to which it is connected according to the weights of connection.

(iii) The connection weights (synaptic strengths) are usually adaptive. Since adaptation can take place everywhere in the structure of the network, we speak of distributed memory. Connection weights are adjusted according to some adaptive (optimization) algorithms. The algorithms employed depend on specific applications and more or less close to neuronal information processing principles. In some applications the connection weights are adapted or optimized only once during the training phase (process) and after that no longer imitate real biological neural networks in which learning is a permanent process.

(iv) The processing units (neurons) contain typically nonlinear activation functions, i.e. the new state of a neuron is a nonlinear function of the signals created by the firing activity of other neurons.

(v) Although the networks often use imprecise and unreliable elements they are characterized by high robustness to noisy input data and element failure through the use of a highly redundant distributed structure. In other words, neural networks exhibit notable robustness since their functionality is not affected by parameter variations over a wide range.

There are many different ways to connect artificial neurons (processing units) to large networks. These different patterns of interconnections between the neurons are called architectures or circuit structures. Such large networks may be able to perform complex tasks which would be impossible for individual neurons. The architectures of artificial neural networks can roughly be divided into three large categories:

a) feed-forward (multilayer) neural networks

b) feedback (recurrent) neural networks

c) cellular neural networks

In feedforward neural networks artificial neurons are arranged in a feedforward manner (usually in the form of layers), i.e. each neuron may receive an input from the external environment and/or from other neurons, but no feedback is formed. A standard feedforward neural network consists of simple processing units (without dynamic elements). A feedforward neural network computes an output pattern in response to some input pattern. Once trained (with fixed connection weights) the output response to given input pattern will be same regardless of any previous network activity. This means that the feedforward neural network does not exhibit any real dynamics, and there are no stability problems in such networks. For feedforward networks the dynamics are often simplified to a single instantaneous nonlinear mapping, [Cichocki and Unbehauen 1994].

On the other hand, feedback neural networks the dynamics are no longer trivial since they consist of processing units with dynamic building blocks and they operate in feedback mode. The dynamic properties of such networks are described by a system of nonlinear ordinary differential or difference equations. A feedforward neural network is represented by static nonlinear maps while feedback neural networks are represented by nonlinear dynamic systems.

Cellular neural networks (CNN), similar to cellular automata consists of regularly spaced special artificial neurons called cells which communicate directly with other neurons only in their nearest neighborhood. Adjacent cells can interact with each other by means of mutual lateral interconnections. Cells not connected can affect each other indirectly because of the propagation of signals during the transient regime. The cells are usually organized in a two-dimensional array of a rectangular, triangular, hexagonal or other regular grid. One of the simplest possible CNN architectures is the rectangular grid for which any cell is connected with its 8 neighbors. Due to a local connectivity every cell is excited by its own signals and by signals flowing from its adjacent cells. Due to mutual interactions the processed signals propagate in time within the whole array of CNN. [Cichocki and Unbehauen 1994]

Generally, an artificial neural network is characterized not only by its architecture, but also by the type of neurons used, by the leaning procedure and by the form (principle) of operation. Artificial neural networks can operate, in general, as deterministic or

stochastic systems. In deterministic ANNs all parameters and signals have deterministic nature. In stochastic ANNs signals and parameters (connection weights) are changed randomly (from time to time with same probability) by some random amount [Cichocki and Unbehauen 1994]

The artificial models(paradigms) of real biological neural networks developed so far are only very simple and rather crude approximations of real biological structures. The problem of artificial neural network architectures is still quite open and the subject of current intensive investigations by many researchers. It is not clear whether it is necessary to model possibly exact biological structures or whether one can reach desired properties by using models which do not fully correspond to real biological nervous systems.

3.3 Artificial Modeling for Neural Systems

As given in figure 3.1.a $X_1....X_n$ shows the input vector to the neuron $W_1....W_n$ weight vector shows the weights of the neuron, F refers to the activation function.

Output of a neuron = $F(\sum X[i]W[i])$

Input vector = $X[X1X2.....Xn-1Xn]$

Weight vector = $W[W1W2.....Wn-1Wn]$

Activation Function = F may be sigmoid or hard limiter or other type.

Continuos and discrete models are given in figure 3.1.a and 3.1.b, [Cichocki and Unbehauen 1994].

3.4 Fields of Neural Computations

- CLASSIFICATION
- FUNCTION APPROXIMATION
- MEMORY
- PATTERN COMPLETION
- OPTIMIZATION
- JOB-SHOP SCHEDULING
- RESOURCE ALLOCATION

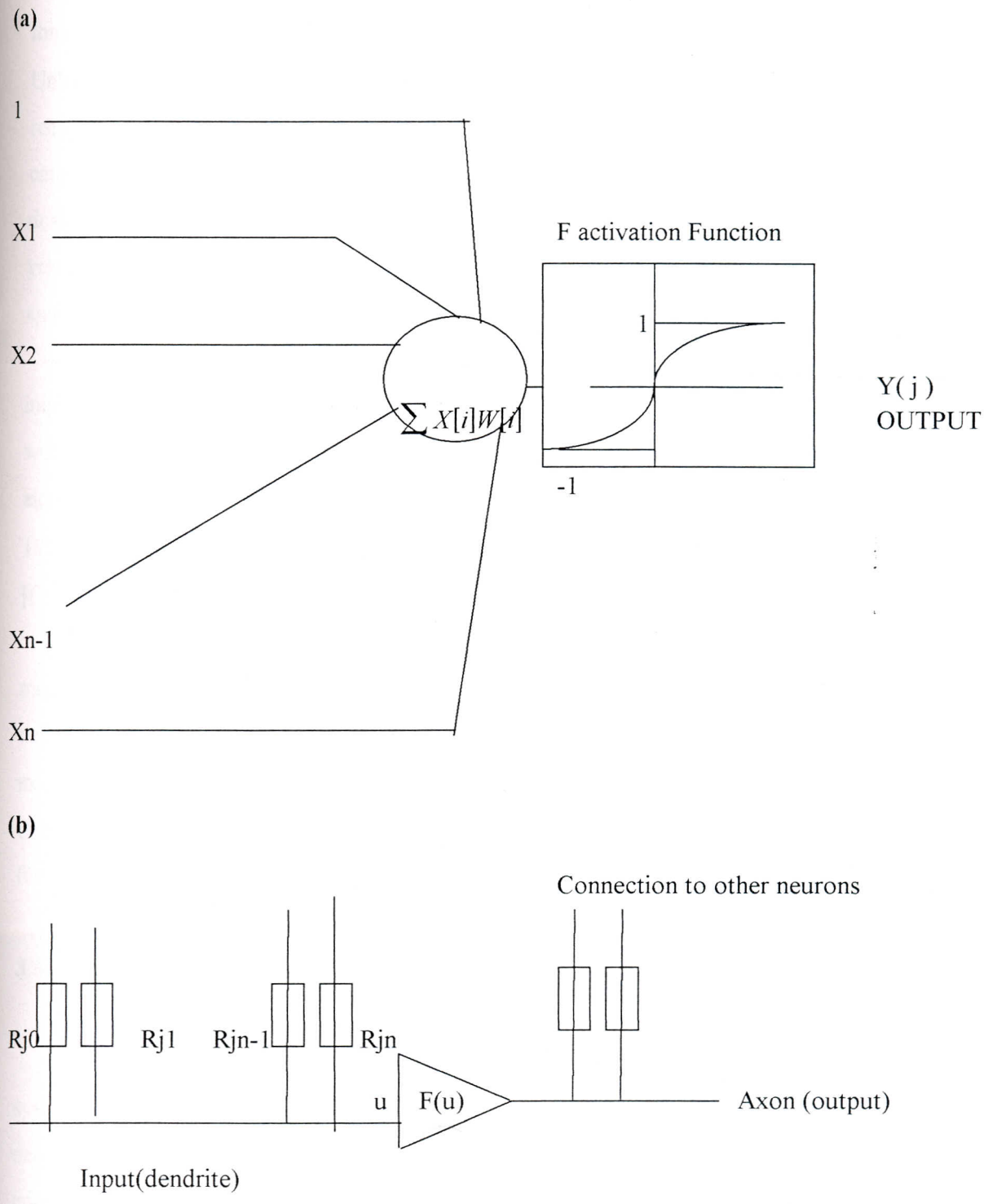


Figure 3.1 a) Simplified functional model of an basic neuron cell and its symbol
b) Electronic analog model of the basic neuron cell(input and output signals are voltages)

performed by the information transfer through these connections[Cichocki and Unbehauen 1994].

Figure 3.1 shows an artificial neural network, where there are sets of inputs and outputs and a number of processing elements which connect inputs through dense connections. Each processing element has inputs/ and/or outputs of other processing elements as its inputs. The relation (as it is most widely accepted) between the inputs and outputs of each processing element for a first order neuron is shown in figure 3.1 a.

Here,

$$y = f\left(\sum_{i=1}^{N} (X_i W_i + W_o)\right) = f(x) \qquad (3.1)$$

where W_i is a weight associated with i'th input X_i and W_o is an offset (bias) value. The function f is a nonlinearity which usually takes one of the two forms, as in Eq. 3.2. Hard-limiter function can take two values, or, the output fires (+1 output) or it does not fire (-1 output) as a function of its inputs and input weights. This function is in general used with binary inputs and outputs. Sigmoid function is a differentiable form of hard limiter, which is mostly used with continuous inputs. These functions will later be mentioned with related algorithms in chapter 6. The general form of the sigmoid function is

$$f(x) = 1/(1 + \exp.(-g\,x)) \qquad (3.2)$$

which is called binary sigmoid [Cichocki and Unbehauen 1994].

(g is parametric value depending your input characteristics)

McCullogh-Pitts [Cichocki and Unbehauen 1994] neuron was one of the earliest and simplest of the neurons that fits above structure. In Mc Cullough-Pitts [Cichocki and Unbehauen 1994] neuron, all the excitatory (positive)connections have the same weights and the hard-limiter transfer function is used. Inputs and outputs are binary. This model has historical importance but finds no practical use today.

A major facility of a neural network is its self-learning property. This is provided by-adjusting the weights w_i to the inputs corresponding to the samples from a given problem to obtain a desired output. Weight adjustment procedures are known as training algorithms.

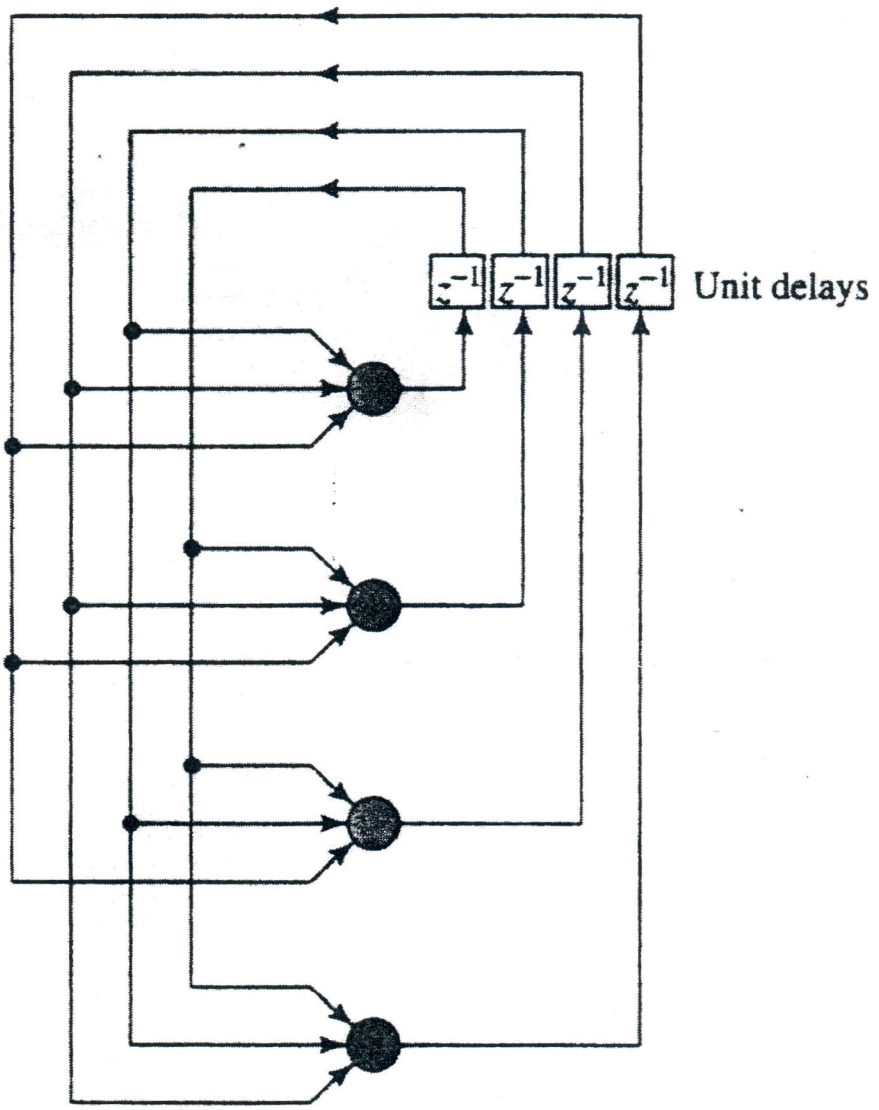


Figure 3.2 Simplified Hopfield Neural Network Architecture

The neural networks can be differentiated by the network topology, learning algorithm, and/or node characteristics as discussed above. In the following section, we will discuss different examples.

3.6.1 Hopfield Neural Network Model

The topology of the Hopfield network is given in Figure 3.2 This model functions as an associative memory with the property that when part of a whole is presented as input, it will be completed by the network and presented at the output. As can be seen in the figure, there are as many outputs as inputs. In the Hopfield network inputs and outputs are binary and hard limiter function is used. The training algorithm consists of assigning weights so that when desired outputs are presented at the input, same is obtained at the output. The weights W_{ij} between nodes i and j are found by the following:

$$W_{ij} = \sum_{i=1}^{n} \sum_{j=1, i \neq j}^{n} X_i X_j^S \qquad (3.3)$$

here, x_i^s is the $i^{th.}$ element of $s^{th.}$ sample (one of the desired outputs).

When an unknown pattern is to be assigned into one of the prescribed outputs, the following steps are followed:

1. Present the unknown sample at the input, initialize the outputs to this value.
2. Remove the input and iterate until output converges to one of the desired outputs.

Hopfield network responds well when a pattern with missing parts is presented at the input and converges to one of the inputs presented during learning. That's why it can be used as an associative memory.

3.6.2 Kohonen's Self Organizing Feature Maps

Kohonen's Network was first introduced by [Kohonen 1992]. Here, we will briefly discuss the topology of Kohonen's Neural network.The self-organizing feature map works essentially as a coder in an unsupervised manner. It is a "single layer" network with a 2-dimensional output topology, as shown in figure 3.3. Each one of the N inputs are connected to each of the outputs. Kohonen demonstrated that by a self organizing learning process, topologically similar inputs produce outputs topologically close to each other. The feature map has the special property of effectively creating spatially organized

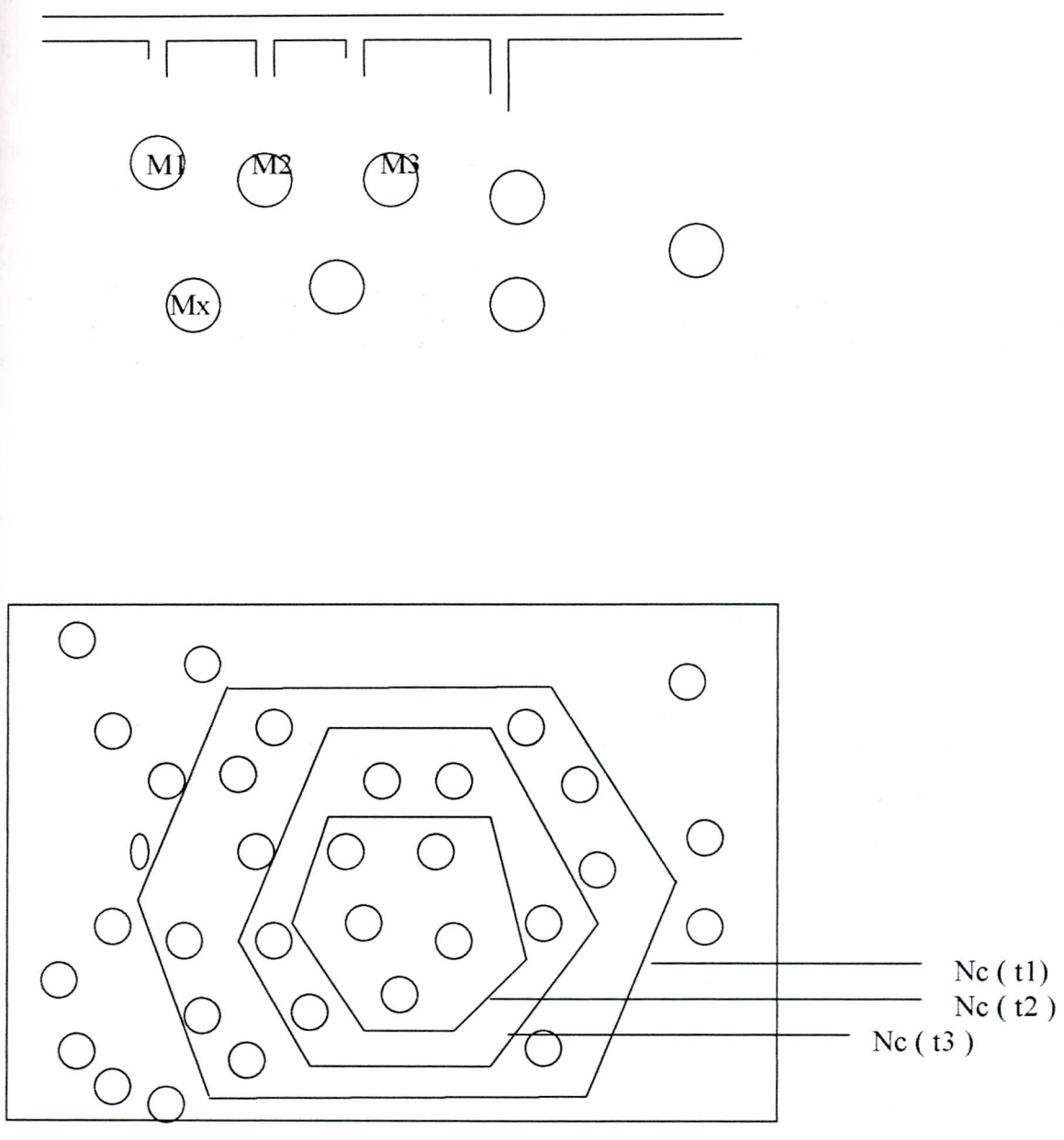


Figure 3.3 Kohonen Neural Network Model

a) Cell arrangement for the map and definition of variables [13]

b) Examples of topological neigbbourhood Nc (t), where t1 , t2 , t3 respectively. [13]

"internal representations" of various features of input signals. A topological neighborhood of each output neuron is effected by the changes in the weights during learning figure 3.3. The neighborhood size starts large and shrunk in time. Define M_i as a weight vector $M_i = (m_{i1}\ m_{i2},...m_{iN})$ that represents the weights between the inputs and the output node i. It can be shown that the collection of M_i's tend to approximate the probability density function P(x) of the input signal $X= (x_1,...x_N)*R_n$ after learning.

Basic self-organizing Feature Map Learning Algorithm can be summarized as follows [Kohonen 1992]:

Step 1. Initialize weight vectors M_i to small random values. Set an initial radius for the neighbors of each output node.

Step 2. Present a new input X_t

Step 3. Compute the distanced d_j between the input X and each output node j using

$$d_j = \sum (X_k[t] - M_k[t]) **2 \qquad (3.4)$$

Step 4. Select node (j*) = min d_j

Step 5. Update weights of node J* and all its neighbors using

$$M_k\ (t+1) = M_k\ (t) + h\ (t)\ (\ X_k\ (t) - M_k\ (t)) \qquad (3.5)$$

for k * N_c, N_c being all nodes in the neighborhood. $M_k\ (t+1) = M_k\ (t)$ for all other nodes. h (t) is a gain term (0.001) which decreases monotonically during training.

After learning, M_j represents the specific code given to the input vector.

There's been a number of studies on the neighborhood topologies. Minimal Spanning Tree topologies (rather than rectangular) were shown to produce effective results for non-uniformly distributed inputs. Here, the neighbors are not selected as the topologically closest ones but by creating an MST of the euclidian distances between the corresponding weight vectors.

3.6.3 Multilayer Perceptron

Multilayer perceptron is a "feed-forward" network with no feedback connections as in figure 3.4. There are a number of "hidden" layers between input and output layers. When there are no hidden layers, the network is called "single layer", which was the initial form of perceptron .

Refers to node or Processing elements(PE)

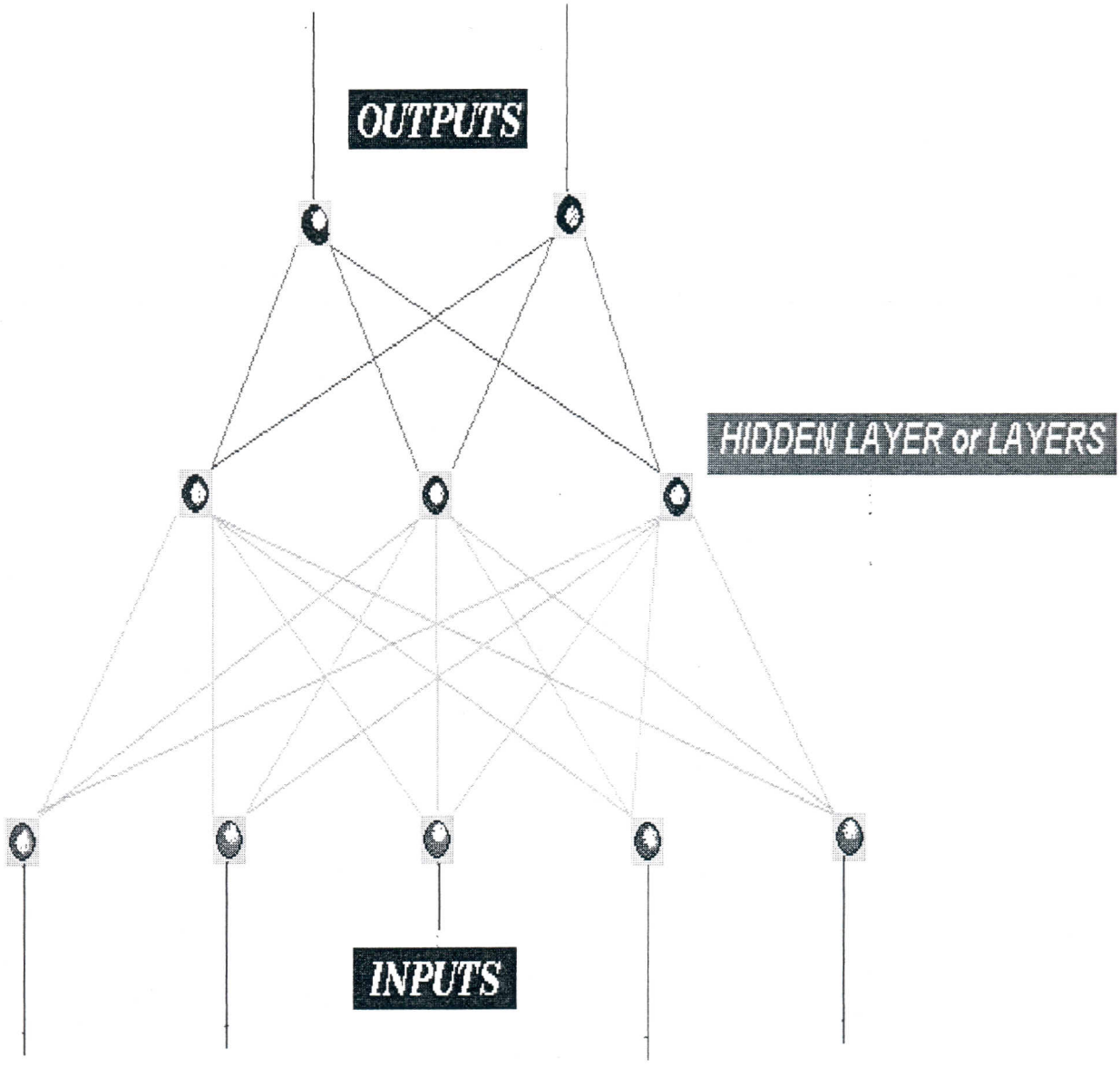


Figure 3.4 Example Multilayer Percepton Structure

This network is used in classification problems where learning samples are in mostly form of "feature vectors". It is assumed that there are M predefined categories and a number of learning samples from each category are available. The network will classify an unknown sample into one of M categories.

It was discussed in [Fausset 1994] that it is sufficient to have N * (2N+1) nodes for a three layer perceptron to result with any arbitrary boundary. Multilayer Perceptron with back propagation algorithm was compared with nearest neighbor cClassification for same sample problems. It was experimentally shown that it performed no worse than k-means NNR Classification. NET talk [Sejnowski and Rosenberg 1986] is an example application for multilayer perceptron that is used in producing text-to-speech sounds.

3.6.4 Recurrent Neural Networks and Boltzman Machines

In addition to the neural networks discussed above, many variations of those and other topologies exist. Here, we will discuss a general "recurrent" neural network, and "Boltzman Machine", a special architecture of a recurrent neural network.

A "recurrent" network is defined as one where feedback connections (loops) are allowed between neurons. Figure 3.5 shows a general recurrent neural network. Actually, the Hopfield Neural network discussed in 3.6.1 is recurrent. Other examples are Boltzman Machines Gaussian Machines, Cauchy Machines [Cichocki and Unbehauen 1994]. Recurrent neural networks are generally used to model sequential or time varying problems such as speech recognition.

A Boltzman Machine is a non-deterministic NN in the sense that the states (outputs) of neurons are found in a probabilistic manner. The neurons have binary states, 0 or 1. The overall network learns towards minimizing a consensus function of the weights. However, at each learning iteration, it is decided to set the state of a neuron using a probability function. The details of the algorithm is given in [Cichocki and Unbehauen 1994]. The procedure is analogous to a process called "simulated annealing", used in statistical physics problems.

The Boltzman Machines are mostly used in constrained optimization problems that qualify as a Traveling Salesman Problem, where an optimum path is to be found between two given nodes.

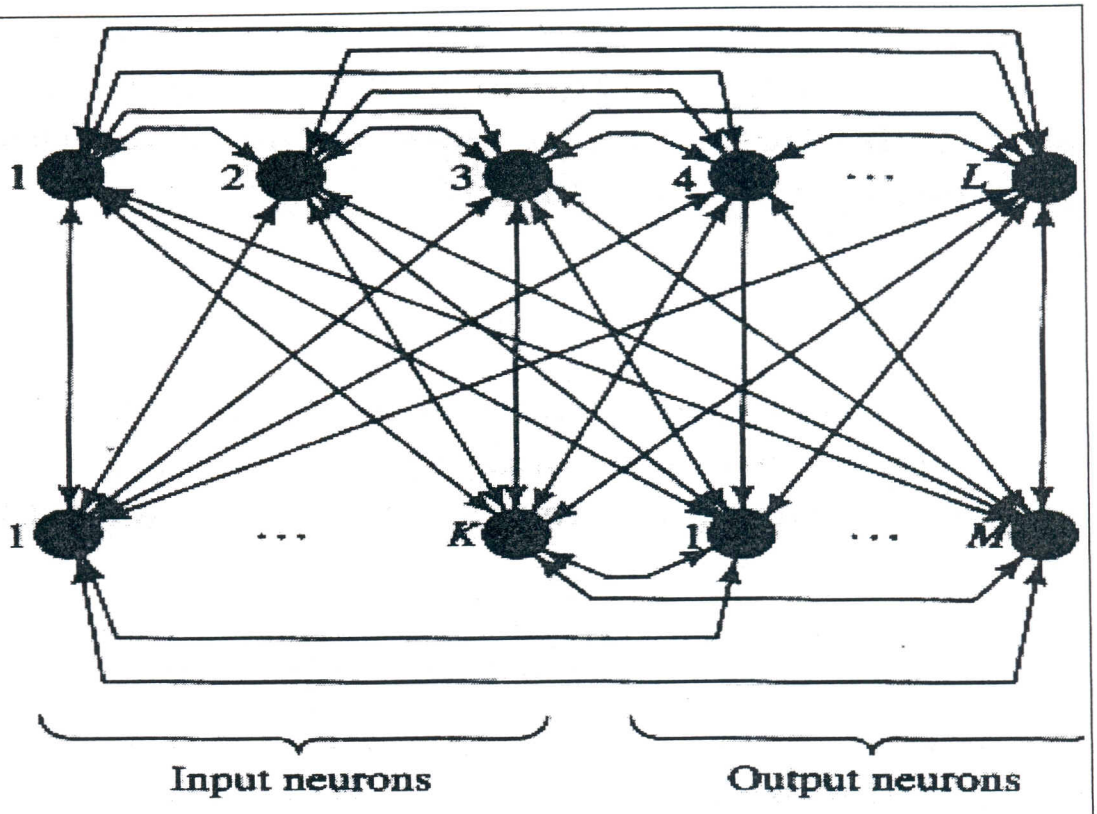


Figure 3.5 Generalized Multilayer Recurrent Neural Network Model

Chapter 4

SHAPE RECOGNITION VIA ARTICIFIAL NEURAL NETWORKS

4.1 Introduction

Shape recognition is an important problem in computer vision. It also has many important industrial, biomedical and military applications. Examples are identification of industrial parts by robots, identification of cancerous blood cells and of classifiers for the above applications is position, scale and rotation(PSRI) invariance. This means that the object be classified in varying positions, orientations and dimensions in the image plane. Another requirement from the classifiers is robustness i.e. they should be tolerant to random variations in shape. Also the classifiers should be able to handle partial occlusion which occurs when there is an overlap of objects or when parts of the object are beyond the field view of the camera.

4.2 Neural Network Approach

Human intelligence, pattern recognition capabilities etc. Are attributed to the massively connected network of biological neurons in the human brain. Recently several attempts have been made to model these biological neurons by artificial analog neurons. These have led to artificial neural networks which have shown lot of promise in solving pattern recognition, discrete optimization and functional approximation problems.

Neural networks offer a number of advantages as compared to traditional classifiers. They provide high computational rates by their ability to test a number of hypotheses in parallel, as compared to traditional classifiers which test hypotheses sequentially. Other advantages are their robustness to noise and the ability to adapt and learn. Recently several researchers have used neural networks in the shape matching problem.

4.3 The Shape Recognition Problem

Shape classification is difficult mainly because the shapes in a class are influenced by sources of shape variation such as :

a) Shapes of a class can be distorted by noise of sensors or digitization.

b) An object can be viewed differently depending on the viewpoint; shapes of a class can differ by perspective transformations.

c) Shapes of a class may be deformed arbitrarily, especially if the object is non-rigid.

d) Parts of a shape may be occluded. The occluded shapes have to be classified correctly.

Most of the shape classifiers are oriented to specific applications and are generally unusable for other tasks for which redesigning the classifier would be necessary.

There are also problems specific to neural networks.

1) The initial parameters of the network can greatly affect how well concepts are learnt.

2) Choice of the network topology is still empirical.

3) After training , the networks are difficult to interpret.

4) Before the learning takes place, it is difficult to quantify the number of samples needed to solve the shape recognition problem sufficiently well. After the learning is over it is difficult to quantify how well the pattern concept has been learnt.

In spite of all the above problems, a number of successful applications have been demonstrated with neural networks, especially with the feedforward neural network trained using the backpropagation algorithm.

4.4 Representation of Shapes

Many representations have been proposed for shape matching. They can be broadly classified as contour based or region based. Contour based approaches are more popular. Most classifier designs are based on a polygonal approximation of the boundary of the shape. A localized representation of the contour is obtained by segmenting the boundary of the points that correspond to the vertices of the approximating polygon. Generally the vertices used are the points of maximum curvature along the contour of the object. Typical local features used are segment lengths, slopes of segments, angles between adjacent segments and a combination of these features. Typically neural networks work with inputs as vectors and thus the representations are also usually in

vector form. Clearly this vector representation should carry the essential shape information and must also be normalized with respect to basic shape transformations such as translation, rotation and scaling.

4.5 PSRI Invariance and Neural Networks

One of the primary tasks of pattern recognition theory is to design classifiers which are invariant to transformations (affine or otherwise). This means that at least part of the output of the classifier must not be affected by the relevant transformations.

Artificial Neural Networks have proven to be extremely powerful classifiers. Whereas the boundaries of decision regions created by conventional classifiers usually belong to restricted families such as a set of hyperplanes, neural networks are able to employ much more general decision boundaries. There are three classes of techniques for invariant neural network recognition as explained [Barnard and Casasent 1991].

4.5.1 Invariance by Structure

The structure of the network is designed such that its output is always invariant to certain transformations. To do this one creates connections between the neurons which force transformed versions of the same input to have the same output. This approach although intuitively appealing is practically inconvenient because the number of connections required is prohibitively large for images of realistic size. For example, 4*N connections are necessary to implement complete translational invariance for an input image consisting of only N pixels with only one object present.

4.5.2 Invariance by Training

Since Neural networks are such powerful classifiers we can use their classification ability directly to obtain transformation invariance. To do this a number of different examples of the same object are input to the neural network for training; these examples correspond to different transformations of the object.

There are two disadvantages to the above method :

1) It is not necessary that a neural network trained to recognize one object invariantly can use this training to also recognize new objects invariantly.
2) The demands placed on the classifier system may be very severe.

4.5.3 Invariant Feature Spaces

It is possible to extract features which are invariant to transformations of an input. These features can be used to train the classifier. Invariant feature spaces that have been used in conjunction with neural networks are wedge-ring samples of the magnitude of the Fourier transform, the magnitude of the Fourier transform in log-polar coordinates and moments.

The advantages of this method are that features can be reduced to realistic levels. The main disadvantage is the need to calculate the features before the classifier can be employed.

[Yüceer and Oflazer 1993] have proposed a hybrid classification system based on a pattern preprocessor and an artificial neural network classifier that can recognize patterns deformed by the Japanese alphabet recognition and classification of geometric figures.

[Khotanzad and Lu 1990] have used the multilayer perceptron trained with backpropagation for distortion invariant character recognition. Features derived from geometrical moments of the image have been used as the input to the neural network.

4.6 Associative Memories for Shape Recognition

The first effort in this direction was by [Li and Nasrabadi 1989]. They have used a Hopfield neural network for the graph matching problem. In this approach, each object is represented by a set of landmarks (which are points of interest relative to the object that have important shape attributes). There are various algorithms in the literature for locating these dominant points. Given a scene consisting of a partially visible object, a model object in the scene is determined by how well the model landmarks are matched by those in the scene. The landmark matching task is done by a Hopfield neural network.

[Tsao et. al. 1993] have used constraint satisfaction neural networks for image recognition. They tackled the problem of identifying the bone, ventricle and gyros from

the X-ray CT images of a human head. They use a three layered neural network where each layer identifies one physical entity from the input regions. They have used a combination of Kohonen's self organizing maps and the optimal linear associative memory.

4.7 The Multilayer Perceptron in Shape Recognition

The first efforts in this respect were by [Gupta et. al. 1990]. A three layer perceptron using backpropagation is used for classification of closed planar shapes. The network is initially trained with contour sequences of noise-free reference shapes and generalization capability is demonstrated. Then the network is gradually retrained with increasingly noisy data to improve the robustness of the classifier.

[Pal et. al.1993] have described a method whereby a shape is approximated by a constant point polygon (between any two adjacent vertices of the polygon the number of points on the contour of the shape is constant). This representation is applicable for both concave and convex shapes. The angle of variation between two consecutive line segments is taken as the primary feature and the sequence is modeled as a circular autoregressive process. The least square error estimate of the AR coefficient vector is used as the input to a multilayer perceptron network for learning and classification.

[Mitzias and Mertzios 1994] have described a shape recognition technique which uses a neural classifier and which uses a fast polygon approximation technique. This approximation technique extracts feature vectors of fixed dimensions that characterize a given shape. These feature vectors are used as inputs to a neural classifier. The authors claim that this technique is fast enough for real time applications.

4.8 The Hybrid Approach

There have traditionally been two approaches to the pattern recognition problem, one using hand-built classifiers and the other using empirical learning. Hand-built classifiers use a domain theory without an extensive set of examples. There are some drawbacks to this approach.

1) The domain theory might be incomplete i.e. all the possible rules might not be there. Also verifying the correctness of all the rules is difficult.

2) Domain theories may be intractable to use i.e. it might require thousands of rules.

3) Domain theories may be difficult to modify. Empirical learning systems on the other hand generalize from specific examples. However they ignore problem specific theory. The drawbacks to this approach are :

1) Many features could be used to describe the same object. Also different objects could possess similar features. Feature selection is thus a problem.

2) Complex features constructed from the initial features may simplify learning. However constructing such features is difficult.

3) Uncommon cases may be difficult to handle as small sets of exceptions may be poorly represented or altogether unrepresented.

Artificial Neural Networks are a particular method of empirical learning. The feedforward neural network trained using the backpropagation algorithm has been a popular choice for pattern recognition. The main reason for this is that the above network can learn any non-linear mapping since it is an universal approximator. The neural network can also acquire the knowledge of the problem directly from the data. People rarely learn purely from theory or examples, often the learning is from a combination of both. In cases where certain expert knowledge is available about the pattern recognition domain, it may be helpful if a priori knowledge could be incorporated in the structuring and learning of the neural networks. Various models have been proposed for combining artificial neural networks and expert systems by [Gallant 1992].

[Towell and Shavlik 1994] have proposed a hybrid learning system built on top of connectionist learning techniques. It maps problem specific domain theories represented in propositional logic into neural networks and then refines this reformulated knowledge using backpropagation. This system has been found to generalize better than a wide variety of learning systems.

Most of the above methods presuppose that a set of rules can be found from which the initial network architecture can be decided. However formulating these rules itself is not an easy task in the general 2-D shape recognition problem, for example, in the classification of aircraft from their contours. Rule extraction from the trained network will not only be helpful in validating the network, but also in deciding the initial parameters of the network.

[Kim and Yang 1994] have recently proposed a neural network model for handwritten character recognition. They use a modification of the backpropagation

algorithm as their links in their architecture are either positive or negative. Their network uses three kinds of knowledge representation: feature variables, relation among features and model patterns. The relation between the features is incorporated into the network using fuzzy rules. This information is used in the initialization of the network. The learning occurs in two phases: in the primary learning some weights are kept fixed whereas in the secondary learning all weights are updated. There are many applications in which adaptive, incremental learning is required, for example an autonomous robot in a hostile environment.

The requirements from such a learning are:

a) It should be fast

b) It should not invalidate important older knowledge i.e. Truth Maintenance is necessary. This is called the stability-plasticity problem in neural networks.

c) It may forget redundant/unimportant older knowledge. The weight matrices of the neural network are obtained by solving a set of linear equations coupled by sigmoidal nonlinearities in least square sense. As new patterns are added, the least squares solution is adapted to incorporate information about the new patterns. The neural network architecture is kept fixed here.

4.8.1 Choice of the Network Topology

Finding the right number of neurons for the hidden layer is still a big problem and very application specific. One of the first attempts at a systematic procedure for deciding the network topology was by [Murphy 1992].

Research on fault tolerance of neural networks may prove to be useful in designing the topology of the neural network as the fault tolerance of the multi-layer perceptron is strongly related to the number of redundant hidden nodes it possesses.

[Neti et. al. 1992] have proposed a feedforward neural network with a guaranteed level of fault tolerance. The problem of estimating weights for such a network is formulated as a large-scale non-linear optimization problem. Also they have found out that such networks derived by introducing fault tolerance constraints have better generalization properties then solutions obtained via unconstrained back-propagation.

[Jou et. al. 1994] have given an analysis of the relationship between the hidden nodes and the fault tolerance of the multi-layer perceptron. They have also proposed a

method for the reduction of redundant nodes which can also give the maximum number of crashed nodes allowed within a given error for the neural network.

[Khunasaraphan et. al. 1994] have proposed a self-recovery technique of feedforward networks called weight shifting. If some input links of a specific neuron are detected to be faulty, their weights will be shifted to healthy links of the same neuron. When the neuron itself goes faulty this is treated as a special case of the above instance where all the links are faulty.

4.8.2 Interpreting the Weights of the Network

If classification is done by discriminant functions or rule-based networks, we can follow the exact line of reasoning which leads from input data to an output conclusion. In a multilayer feedforward network, all we have is a huge collection of weights connecting the neurons. The complexity inherent in such a model makes its understanding quite difficult as the networks can implicitly encode exponentially many If-Then rules as compared to the weights in the network. However validation of the network is quite important, especially when the network is going to be used for control of a plant, guidance of a missile or financial applications for insurance companies and banks. Some features of the data may be irrelevant while other features may be quite critical to the network's decisions.

[Kim and Yang 1994] have implemented a result explanation capability in their neural network for character recognition by tracing the value of nodes and connection weights. However the network is itself structured from fuzzy rules and this makes the result explanation easier.

However in the general shape recognition problem (for example aircraft classification from 2-D contours) it is difficult to formulate rules beforehand.

[Gallant 1992] has proposed some algorithms for extracting rules from neural networks, wholly based on examination of weight vectors. However highly correlated input variables could make rule extraction difficult, although the methods proposed in [Gallant 1992] in conjunction with those in [Kim and Yang 1994] may work well.

[Masters 1993] has also proposed some methods for interpreting the weights.

a) By clustering the training set based on activation levels of all hidden neurons.

b) By sensitivity analysis. The network error for the entire training set is evaluated with one of the inputs clamped to a fixed value for all training samples. The importance of that input can be gauged by the effect on the output error.

c) By producing stereotypical inputs. Here we try to find optimal input patterns for producing any desired output pattern. This will enable us to get an idea about those input features the network does not care about.

These results could also prove useful in the selection of features. It would be quite interesting to investigate these methods in the context of shape recognition.

Chapter 5

SHAPE REPRESENTATION MODELS

5.1 Introduction

In order to create base information for the introduction of the method explained in detail in chapter 7, some existing techniques for the representation and description of 2-D objects are reviewed in this chapter. The list of procedures listed here is by far not a complete summary, but serves as an orientation within this specific topic of artificial vision. Each of the methods is shortly described with an emphasis on the respective advantages and/or drawbacks. Additionally, following chapter (chapter 6) also extends our methodology, in terms of classifier when we consider the X-ray image classification problem in modular manner.

[Rauber 1986], is an compact survey over existing shape description techniques. He describes two different families of techniques, those describing the interior region of the object and those describing the boundary. This distinction is already a constraint by itself for the descriptor. For the accessibility to both domains, it assumes that the object must posses both an interior area and a boundary that delimits the object. Many methods assume furthermore that the boundary of the object is closed. This is normally not the case for the patterns like handwritten characters if they are composed only of strokes which do not have a measurable thickness. A hole inside a closed boundary of an object cannot be described by the same descriptor as the boundary itself. The only remedy to overcome this problem is to use hierarchical organization of descriptors to characterize nested type [Rauber 1986]. Generally speaking, there is no method suitable for the description of any kind of pattern.

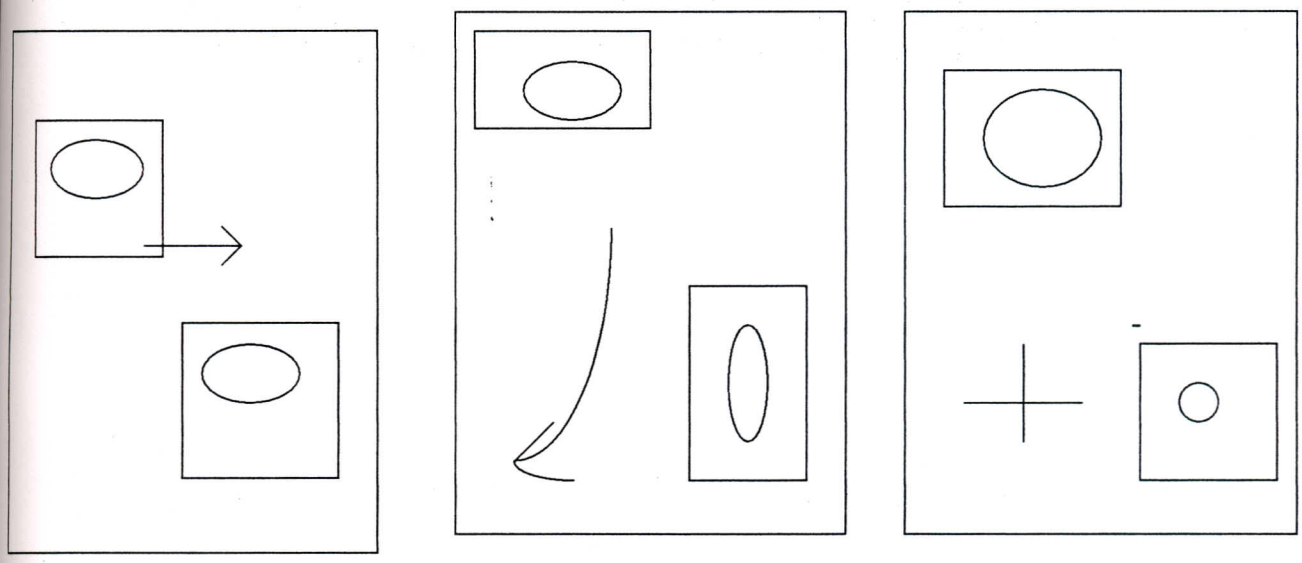
[He and Kundu 1991] gives also a short overview for 2-D shape classification techniques in the introduction of a paper which uses hidden Markov models for the description of partial sequences of the boundary o fan object.

A common quality requirement can be defined for all shape descriptors. The shape or form of the object is an attribute that must not change when the original object is submitted to a certain set of affine geometric transformations (Figure 5.1), or an arbitrary combination of the same. A 2-D shape descriptor should be insensitive to:

- Translations
- Scale changes(uniform in both the x-coordinate and y-coordinate)
- Rotations

This axiom for the shape descriptors implies that the descriptor is able to perform a normalization for the different appearances in which the object may occur. The actual instantiation of the object must be mappable to an isomorhous prototype of the pattern.

Affine geometric transformations



Translation Rotation Scaling

Figure 5.1 The form of an object is insensitive to the three affine geometric transformations: translation, rotation and uniform change of scale.

Description is the basis for classification. Ideally, the descriptors should be quantitative values which can be processed by a digital computer. It is then possible to build a pattern recognition system on top of the describing shape features.

5.2 Boundary Descriptors

Edge detection in image processing is an important preprocessing step. Its objective is the extraction of boundary points of the objects in the scene. The transitional zones of the image where a change between regions of interest and background takes place, contain most of the information of the scene. A human being is able to recognize a form often by merely regarding its silhouette. A shadowplay is an example of a pattern recognition process where the viewing public extracts all visual information from only boundaries of the participating objects and people. Some of the techniques which are based exclusively on the boundary of the object, are described in the following.

5.2.1 Chain Codes

Chain codes [Rauber 1986] are a compression and description tool for the boundary of the shapes. Starting from one boundary point, the next point is searched in one of a small set of possible directions, based on the neighborhood model of the image, 4-connectivity or 8-connectivity. Three bits are enough for the recording of the sequence of directional changes when following the curve. Chain codes are usually employed for the encoding of the boundary of binary objects in image processing systems.

The advantages of chain codes as shape representation tool are as follows:

- Direct obtainment of numerical encoding of the boundaries possible. A quantitative description of the boundary of the object can be derived from the binary image by the chain code technique.
- A compact representation of the boundary. Instead of storing the absolute coordinates of the boundary points, a sequence of directional changes is used. This saves a lot of storage memory.
- Normalization is possible. The chain code is invariant to translations of the object. A normalization relative to rotations and scale changes is feasible.

The disadvantages of chain codes as shape representation tool are as follows:

- The method is normally conceived only for closed boundaries.

- Transformations of the pattern (rotation, scale) change the pattern itself. This signifies that normalized chain code obtained from the transformed pattern is not equal to the chain code of the original pattern.
- Noise sensitivity. For the purpose of the direct matching of the chain code in a classification task, noise must be reduced to a minimum. If for instance a convexity of the object is lost because of an illumination disturbance, the chain code will be completely different in length and morphology. Therefore a matching will surely be possible.
- No nested processing Objects inside other objects must be described by a separate chain code. The global structure of a nested form cannot be characterized by a single chain code.

5.2.2 Polygonal Approximations

The idea behind this technique is to describe an eventually complex contour by a sequence of straight lines. By varying the number of the approximating lines, the coarseness of the resulting polygonal model of the original pattern can be controlled. In [Rauber 1986], a series of different approximation techniques is presented. [Li and Nasrabadi 1989] uses the split and merge method proposed by Pavlidis [Rauber 1986], as the feature model for a classifier based on a Hopfield neural network.

The advantages of polygonal approximations as shape representation tool are as follows:

- achieve a high compression rate for regular shapes
- can be used for the incomplete matching of shapes. This permits the recognition of objects which are partially occluded.

The disadvantages of polygonal approximations as shape representation tool are as follows:

- allow no nested processing of curves inside other curves.
- are sensitive to disturbances of the contour.
- are not invariant to rotations and scale changes.

5.2.3 Signatures

The commonly used signature is distance from the centroid of a closed boundary points as given in [Rauber 1986]. Generally speaking, signatures are one-dimensional representations of two-dimensional curve shapes. One-dimensional moments and Fourier descriptors [Rauber 1986], often use signatures as the intermediate representation, before calculating the final features.

The advantages of signatures as shape representation tool are as follows:

- Simple shapes are concisely represented by signatures
- Normalization for scale changes is relatively easy, translation insensitiveness, rotation results in cyclic shift of the function values.

The disadvantages of signatures as shape representation tool are as follows:

- No nested boundaries allowed.
- Concave objects generate ambiguous distance measures if the distance is sampled along a constant increment of the angle. If the distance is sampled along a constant curve length, concave objects can be represented, but the distance function is not 2Π - periodic. In this case, the signature is rotation sensitive.

5.2.4 Fourier Descriptors

Since the position along a closed contour is a periodic function, Fourier series may be used to approximate the contour. The resolution of the contour approximation is determined by the number of terms in the Fourier series.

Suppose that the boundary of an object is expressed as a sequence of coordinates $u(n) = [x(n), y(n)]$, for $n=0,1,2,...,N-1$. We can represent each coordinate pair as a complex number so that $u(n) = x(n) + \mathbf{j}\, y(n)$, for $n=0,1,2,.....,N-1$. In other words, the x axis is treated as the real axis, and y axis is treated as the imaginary axis of a series of complex numbers. Note that for a closed boundary, this sequence is periodic with Period N and that now the boundary is represented in one dimension.

5.2.4.1 Formulation

The discrete Fourier transform (DFT) representation of a one-dimensional sequence u(n) is defined as

$$u(n) = \sum_{k=0}^{N-1} a(k) e^{\frac{j2\Pi kn}{N}}, 0 \le n \le N-1 \qquad (5.1)$$

$$a(k) = \frac{1}{N} \sum_{n=0}^{N-1} u(n) e^{\frac{-j2\Pi kn}{N}}, 0 \le n \le N-1 \qquad (5.2)$$

The complex coefficients a(k) are called Fourier descriptors of the boundary. Fourier descriptors are compact representations for closed contours. However, low-resolution approximations, using only the low-order terms in the series, can be used as an even more compact representation. If only the first M coefficients are used, which is equivalent to setting a(k)=0 for k>M-1 the following approximation to u(n) is obtained:

$$\bar{u}(n) = \sum_{K=0}^{M-1} a(k) e^{\frac{j2\Pi kn}{N}}, 0 \le n \le N-1 \qquad (5.3)$$

Although only M terms are used to obtain each component of the boundary u(n), n still ranges from 0 to N-1. In other words, the same number of points are in the approximated boundary, but not as many terms are used in reconstructing each point. Simple geometric transformations of a boundary, such as translation, rotation, and scale, are related to simple operations of the boundary's Fourier descriptors. This makes the use of Fourier descriptors attractive for boundary matching. However, Fourier descriptors do have problems with occluded shapes. There other methods for obtaining, similar descriptors, using of the boundary representations.

5.2.4.2 Different View

The first elaborated definition of the Fourier Descriptor was given by [Persoon and Fu 1986]. The technique of Fourier descriptors can claim to be real shape feature of a plane closed curve. For many boundary patterns the method of Fourier descriptors can satisfactorily characterize the form of the pattern. Encoding, reconstruction and classification of 2-D curves is possible with Fourier Descriptors. The original definition of [Persoon and Fu 1986] is given below.

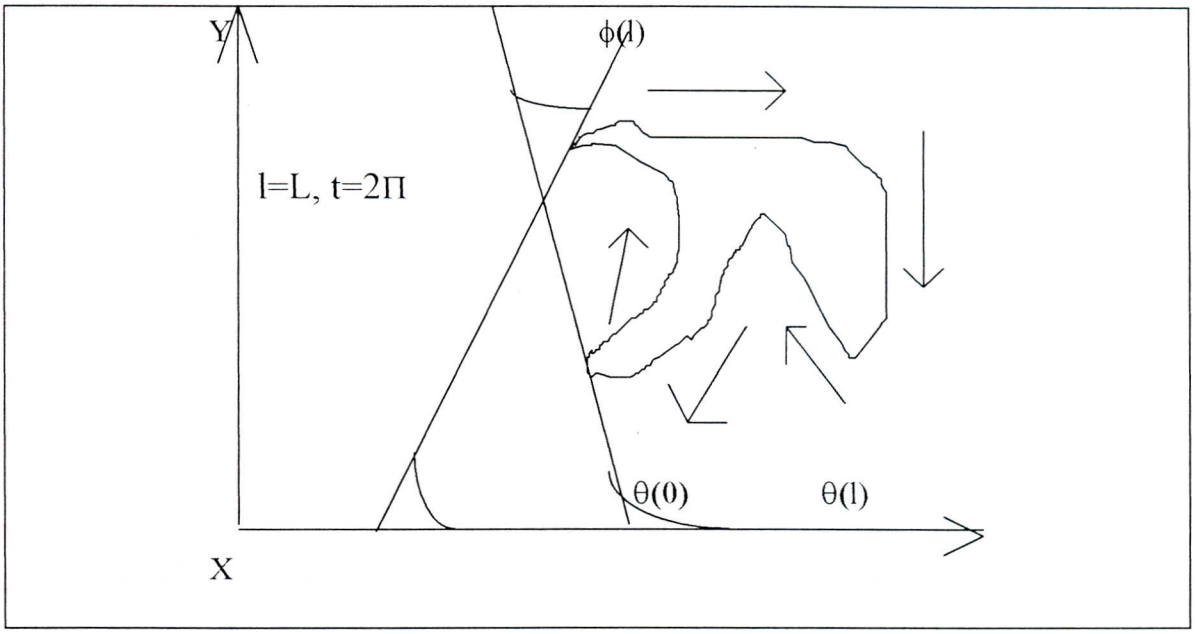


Figure 5.2 Angular direction θ(l) and cumulative angular function φ(l) as the basis for Fourier Descriptors.

The arc length l along the curve γ can be calculated from a parametric representation of the curve coordinates z(t) = (x(t),y(t)), where the parameter t varies between 0 and a maximum value T and the curve is assumed to be smooth, i.e. the derivative dz(t)/dt exists for all t ∈ [0,T].

$$l(t) = \int_{o}^{\|t\|} \|dz(t)/dt\| dt \quad \text{where} \quad \| \ \| \quad \text{is the Euclidian norm.} \qquad (5.4)$$

The parametric representation of the curve can unequivalently be expressed by the parameter l: z(l) = (x(l),y(l)). The angular direction of γ at arc length l is θ (l), see Figure 5.2. The cumulative angular function φ(l) sums the absolute angle θ (l) along the whole curve. For a smooth closed curve the relation φ(l) = θ (l) - θ (0) holds.

The arc length l is the period for the periodic function φ (l) . Since the ideal period for a Fourier Transforms. is 2π, φ (l) is normalized, using the parameter t, that varies between 0 and 2π. In order to compensate for the cumulative angle -2π that is accumulated with φ (l) when the curve is scanned once, an additive term is joined to the normalized form to obtain the periodicity condition φ (0) = φ* (2 π) =0

Normalized cumulative angular function: $\Phi^*(t) = \Phi(\left[\frac{Lt}{2\Pi}\right] + t$ Finally the Fourier

expansion of the function $\Phi^*(t) = \mu_0 + \sum_{k=1}^{\infty} A_k \cos(kt - \alpha_k)$ and (5.5)

Fourier descriptors $(A_k, \alpha_k), k = 1.....\infty$

The A_k are the harmonic amplitudes of the Fourier expansion and the α_k are the phases angles. It can be verified that the Fourier descriptors are invariant to the translation of the shape. The A_k are not affected by rotations to a different starting point. The normalization against changes in the scale of the shape is achieved by dividing each A_k by A_1 which is normally the largest of the harmonic amplitudes.

In practice it turns out that only the harmonic amplitudes A_k are appropriate for the classification of the shape [Rauber 1986]. The noise introduced by the discrete nature of the image grid reduces the discriminative properties of the phase angles α_k to a minimum.

Particular problems with the behavior of the Fourier descriptors are not elaborated here. Only the ideal case where shape is closed, smooth curve is considered. Fourier descriptors can as well as be obtained from polygons and line patterns that do not possess as interior region [Persoon and Fu 1986].

The advantages of Fourier descriptors as shape representation tool are as follows:

- Fourier descriptors capture the shape of the object in a feature vector ideal for machine supported classification.
- Fourier descriptors are easy to normalize.
- Fourier descriptors are information preserving. The original curve pattern can be reconstructed from the Fourier expansion. This is true due to the very nature of the Fourier transform.

The disadvantages of Fourier descriptors as shape representation tool are as follows:

- Curves must be closed. Problems arise when the curve has any discontinuity. Heuristic preprocessing must be applied to the curve in order to close it.
- No nested processing of curves is possible.

5.3 Region Descriptors

We pass now to a series of methods that represent the shape not by its boundary but its interior region. The first commonsense conclusion is that objects without an interior region are not representable by this group of shape descriptors. In fact it can be observed in the reviewed literature of [Rauber 1986] that all objects supposed to be characterized by region descriptors had a non-zero thickness. This is true for the method of moments and the Log - Polar transform which will be presented below.

5.3.1 Image Moments

The concept of moment as invariant image features was investigated by [Li 1992]. See [Khotanzad and Lu 1990] for an excellent treatment of the subject. Principally, in this section, the fundamental presentation of moments that is given in [Rauber 1986] is summarized for giving brief information about the subject.

The analytic definition for a moment of order p+q and a continuos 2-D function f(x,y) is given as

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)dxdy \quad p,q=0,1,2... \qquad (5.6\ a)$$

For the case of the discrete image matrix f(x,y), the moments of order p+q become

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \qquad (5.6\ b)$$

The central moments of a digital image are defined as

$$\mu_{pq} = \sum_x \sum_y (x-\bar{x})(y-\bar{y}) f(x,y), \qquad \bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \qquad (5.7)$$

Then normalized central moments are derived from the central moments as

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}}, \qquad \gamma = \frac{p+q}{2} + 1 \qquad p+q=2,3,.... \qquad (5.8)$$

Finally for order 2, and 3, seven invariant moments ϕ_I , ý=1....7 can be derived that insensitive to the affine transformations, translation, rotation and uniform changes in scale. The explicit definition ϕ_I , is given in [Rauber 1986]

Like the 2-D Fourier transform and the Complex-Log transform which will be characterized below, moments are image oriented features. Basically one can state that

the input for moment invariants are pixel values f(x,y), sampled at a certain image point(x,y). Moment invariants are not generated from the analytic information of the object geometry, but rather an instantiation of an object on a 2-D pixel matrix. Ultimately the features are calculated from the pixel values of the image.

In all classification experiments of the reviewed literature that were based on moment invariants, the patterns had an interior region. This suggests the empirical conclusion that moments are not appropriate to describe patterns where only line oriented shape information is available.

5.3.2 Log -Polar Transform

The mathematical basis for the Log-Polar transform (sometimes also denoted as Complex-Log transform) is the property of the logarithm of a complex number:

$z=x+jy$

$\ln(z)=\ln(x+jy)=\ln(|z|)+j\theta+2jk\Pi$ where k is an integer number. The Euler relation

$z=|z|e^{\theta j}$ is used here, with $|z|=\sqrt{x^2+y^2}$ and $\tan\theta=\frac{y}{x}$ Normally only the principal part of the logarithm is used (k=0). The result of the logarithm of a complex number is a new complex number, thus defining the Log-Polar transform(LPT):

Log-Polar transform LPT:$(x,y)\text{-->}(\ln(|z|),\theta)$

The consequences from the property of the logarithm of a complex number are that a rotation of z by an angle $\Delta\theta$ to $z^i= |z|e^{j(\theta+\Delta\theta)}$ results in the mapped number $\ln(z^i)=(\ln(|z|), \theta+\Delta\theta)$. Hence a rotation of the original point in the Cartesian coordinate system is transformed to a translation of the ordinate in the polar coordinate system. A multiplication of the ordinate in the polar coordinate system. A multiplication of z by a scalar a results in translation of the abscissa of the polar coordinate. Thus also a scaling results in a translation $z^i = |z|e^{j\theta}$ $\ln(z^i)=(\ln(|z|),\theta) = (\ln(|z|)+\ln(a), \theta)$.

The translation of the original Cartesian point $z^i = |z|+\Delta z$ is compensated for by placing the origin of the Cartesian coordinate system in to the centroid of the pattern, before the Log-Polar transform is applied.

In[Massone et.al.1985] the Log-Polar transform is used for the recognition of 2-D objects. A region of interest of the image is transformed from the polar coordinate system.

A non-uniform sampling structure is placed over the center of gravity of the object. The sampling process is determined by several heuristics which limit the general purpose applicability of the method. The final recognition is done by template matching. A template of the Log-Polar transform of all object classes is stored. In the recognition phase the Log-Polar transform of the unknown object is correlated with all classes and the class with the highest correlation score is attributed to the unknown shape.

The disadvantages of Fourier descriptors as shape representation tool are as follows:

- Since the sampling schema of the objects has a decreasing resolution from the center to the edges, information is lost. An inverse transform from the polar coordinate system back to the Cartesian coordinates exhibits smearing effect. Details of the image far away from the center of the sampling structure become unclear.
- One of the biggest of the Log-Polar transform is the high sensibility to a slight misalignment of the center of gravity. If for the same object the center of gravity is computed differently, the resulting images in the polar coordinate system can be totally distinct. This results in poor recognition accuracy.
- The Log-Polar transform is not really scale invariant. A stretching effect can be observed when the object in different sizes I transformed to the polar coordinate system. If a shape is scaled by a factor a, the offset of the abscissa of a particular point z of the shape will always be ln(a), independently of the point z. The abscissa of the transformed coordinate ln(z) however depends always on the point z itself.
- Also the Log-Polar transform works on patterns that posses an interior region.

Chapter 6

MULTILAYER PERCEPTRON AND BACK PROPAGATION ALGORITHM

6.1 Pattern Recognition and Multilayer Perceptron

Pattern Recognition in general is defined as an automated process to learn samples of a number of categories of different patterns and to classify unknown patterns afterwards. The methods of pattern recognition vary from classical bayes decision theory to knowledge based systems.

In all of the approaches where a pattern is represented by a feature vector $X= (x_1,...x_n)$, which is a point in n dimensional feature space R_n , finding classification methods consists of finding ways to split up the pattern space into regions (Figure 6.1) such that the classification errors are minimized meaning that all labeled samples of a class should fall in the same region.

In practice, the decision boundaries have nonlinear complex shapes in general. So any method should be powerful enough to create the proper non-linear boundaries.

Perceptron [Cichocki and Unbehauen 1994]as a pattern classifier, learns in an iterative supervised manner. Unfortunately, single layer perceptron can only generate linear boundaries. Later it was shown that with proper learning algorithms, a multilayer perceptron with at least 2 hidden layers is capable of solving any pattern classification problem. This led the pattern recognition researchers to concentrate on multilayer perceptron and backpropagation learning algorithm and variations of these. Even though other neural networks are also used in pattern recognition, multilayer perceptron is the most popular so far. Compared to other approaches, Artificial neural networks are found to have similar performance with simple and easy-to-apply algorithms. If the parallel neural network architectures are realized, the approach will also have the speed advantage .

In this chapter, we were concentrated on backpropagation algorithm, its details, discussion about training problems, interpretation problems, etc..

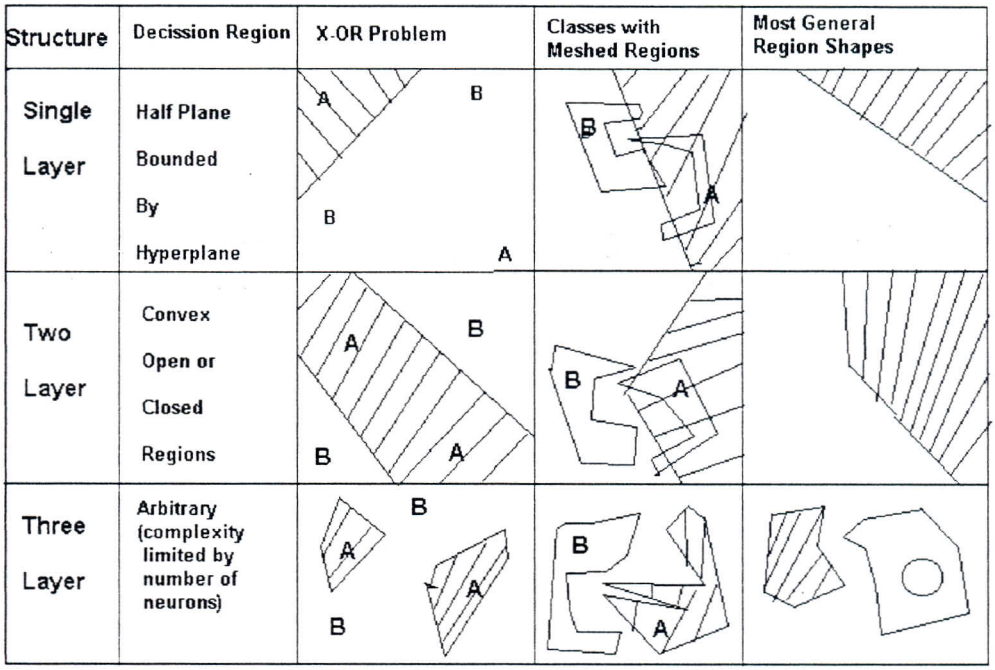


Figure 6.1 Decision Space Regions in Hyperplanes [Lippman 1992]

6.2 Introduction to Backpropagation Algorithm

The back-propagation algorithm was developed first by Werbos [Cichocki and Unbehauen 1994] in 1971 , but his achievement remained almost unknown. The technique was rediscovered by Parker [Cichocki and Unbehauen 1994] in 1982, and independently in 1986 and by [Rumelhart et. al. 1986]. A closely related technique was proposed by Le Cun [Cichocki and Unbehauen 1994]. The multilayer perceptron networks were not used in the past because of lack of an effective learning algorithm. This has recently changed, mainly owing to Rumelhart and his co-workers who have popularized the back-propagation algorithm among the scientific community [Cichocki and Unbehauen 1994]. Recently the back-propagation has also been adapted to feedback(recurrent) neural networks [Werbos 1992]. In this section, we limit our disccussion to back-propagation algorithms for the feedforward multilayer perceptron (MLP). The MLP is supposed to perform a specific nonlinear mapping or association task(e.g. classification, diagnosis, pattern recognition, etc.) which can be expressed in terms of a given input/output pattern(pairs). These input/output

relations will be called a set of learning examples. Learning of the MLP consists in the adaptation of all synaptic weights in such a way that the discrepancy between the actual output signals and the desired signals, averaged over all learning examples (input patterns), is as small as possible. The back-propagation learning algorithm can be considered as an unconstrained optimization problem of a suitably constructed error function(cost function).

6.3 Learning Mechanism in Perceptron

In order to explain the back-propagation algorithm in its basic form, let us first consider the learning of a simple neuron (located in the output layer of a multilayer perceptron (MLP)) (cf. Figure 6.2) in [Cichocki and Unbehauen 1994]. Let us assume that the nonlinear activation function is chosen to be the hyperbolic tangent function,

$$y_j = \Psi(u_j) = \tanh(\gamma_j u_j) = \frac{1 - e^{-2\gamma_j u_j}}{1 + e^{-2\gamma_j u_j}} \tag{6.1}$$

where

$$u_j = \sum_{i=1}^{n} w_{ij} x_i + \Theta_j, \gamma_j > 0.$$

The aim of learning is to minimize the instantaneous squared error of the output signal

$$E_j = 1/2(d_j - y_j)^2 = 1/2e_j^2 \tag{6.2}$$

by modifying the synaptic weights w_{ij}. The problem of learning can be formulated as follows. Given the current set of synaptic weights w_{ij} and the threshold $\Theta_j = w_{i0}$ we need to determine how to increase or decrease them in order to decrease the local error function E_j . This can be done using a steepest-descent gradient rule expressed as in [Cichocki and Unbehauen 1994]:

$$\frac{dw_{ij}}{dt} = -\mu \frac{\partial E_j}{\partial w_{ij}}, \tag{6.3}$$

where μ is a positive learning parameter determining the speed of convergence to the minimum. Taking into account that the instantaneous error can be expressed as

$$e_j = d_j - y_j = d_j - \Psi(u_j), \tag{6.4}$$ where

$u_j = \sum_{i=0}^{n} w_{ij} x_i$ with $w_{j0} = \Theta_j$ and $x_0 = +1$ and applying the chain rule

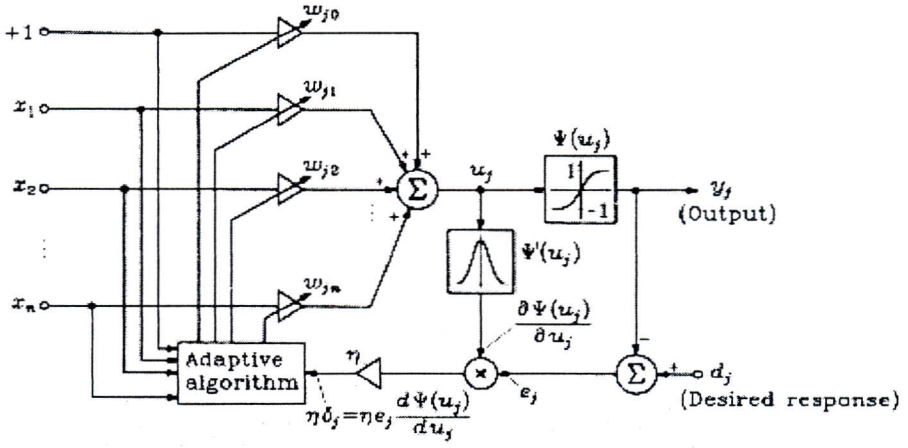


Figure 6.2 Implementation of back-propagation algorithm for the single neuron located in the output layer of the perceptron. [Cichocki and Unbehauen 1994]

$$\frac{dw_{ij}}{dt} = -\mu \frac{\partial E_j}{\partial e_j}\frac{\partial e_j}{\partial w_{ij}} \qquad (6.5),$$

then we obtain $\frac{dw_{ij}}{dt} = -\mu e_j \frac{\partial e_j}{\partial w_{ji}} = -\mu e_j \frac{\partial e_j}{\partial u_j}\frac{\partial u_j}{\partial w_{ij}}$

$$= \mu e_j \frac{d\Psi(u_j)}{\partial u_j} x_i = \mu e_j \Psi^I(u_j) x_i = \mu \delta_j x_i, \qquad (6.6),$$

where δ_j, called the learning signal or local error, is expressed as $\delta_j = e_j \Psi^I(u_j) = -\frac{\partial E_j}{\partial u_j}$.

If the sigmoid activation function is chosen to be the hyperbolic tangent function given by Eq.(6.1) as in [Cichocki and Unbehauen 1994], then the derivation $\Psi^I(u_j)$ is given by:

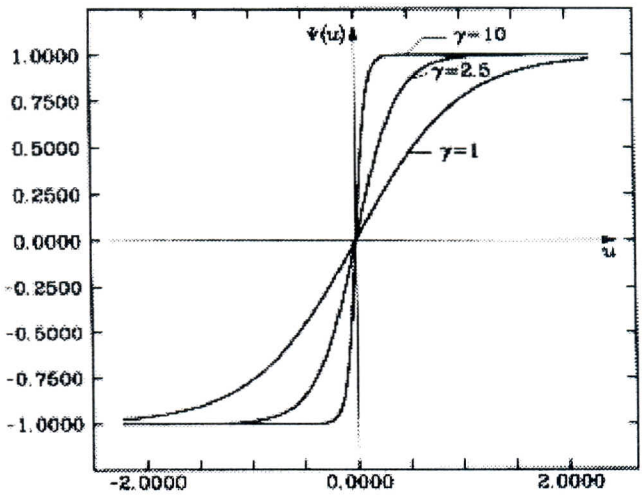
$$\Psi^I(u_j) = \frac{d\Psi(u_j)}{du_j} = \gamma_j\left[1 - (\tanh \gamma_j u_j)^2\right]\gamma_j(1 - y_j^2). \qquad (6.7)$$

In this case Eq. (6.6) can be written in the form

$$\frac{dwj_{ji}}{dt} = \mu\gamma_j e_j(1 - y^2)x_i = \mu_j e_j(1 - y_j^2)x_i, \qquad (6.8)$$

with $\mu_j > 0$. Note that the weight update stabilizes if y_j approaches -1 or +1 since the derivative $\partial y_j / \partial u_j$, equal to $(1-y_j^2)\gamma_j$, reaches its maximum for $y_j = 0$ and its minima ± 1.

a)



b)

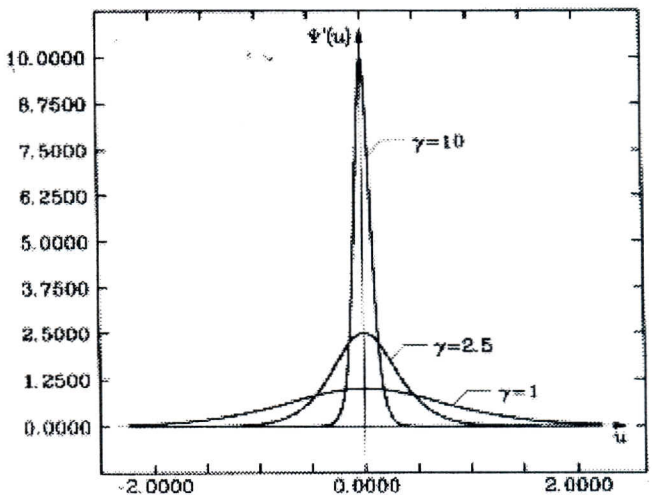


Figure 6.3 Sigmoid activation function $\Psi(u) = \tanh(\gamma u)$ and (b) its derivative for three different values of γ($\gamma=1/T$) [Cichocki and Unbehauen 1994]

perturbation upon the output error $e_j=d_j-y_j$ is memorized(registered). The components of the instantanous gradient can be expressed as an alternative algorithm for updating the synaptic weights: $\frac{\partial E_j}{\partial w_{ij}} = \frac{1}{2}\frac{\partial e_j^{\,2}}{\partial u}\frac{\partial u_j}{\partial w_{ij}} = -e_j\frac{\partial e_j}{\partial u_j}x_i$. (6.13)

For small perturbations we can write

$$\frac{\partial e_j^{\,2}}{\partial u_j} \cong \frac{(\Delta e_j)^2}{\Delta u_j}, \quad (6.14\text{ a}) \quad \text{or,} \quad \frac{\partial e_j^{\,2}}{\partial u_j} = 2e_j\frac{\partial e_j}{\partial} \cong 2e_j\frac{\Delta e_j}{\Delta u_j} \quad (6.14\text{ b})$$

Hence taking into account Eq. (6.3) and (6.12) we obtain the following two forms of

$$\Delta w_{ij} = -\frac{1}{2}\eta \frac{(\Delta e_j)^2}{\Delta u_j} x_i \qquad (6.15\ a)$$

$$\Delta w_{ij} = -\eta e_j \left[\frac{\Delta e_j}{\Delta u_j}\right] x_i \qquad (6.15\ b).$$

For small perturbations Δu_j these two forms of the algorithm are essentially identical. Which form to use in practice is a matter of convenience in the hardware implementation [Rumelhart et. al. 1986]. In order to implement the algorithm the variation of the error caused by a small perturbation signal Δu_j must be memorized and the quantity $(\Delta e_j/\Delta u_j)$ or $((\Delta e_j)^2/\Delta u_j)$ must be computed at each iteration step.

The above described approach of the adaptive learning (updating) of synaptic weights can be extended to the multi-layer perceptron (MLP). For simplicity of our further considerations let us assume that the MLP consists of three layers of neurons: the first hidden layer with n_0 , inputs and n_1 units, the second hidden layer with n_2 units and the output layer with n_3 units (Figure 6.4).

The network behavior should be determined on the basis of a set of input/output pairs(i.e. learning examples). Each learning example is composed of n_0 input signals x_i $(i=1,2,....n_0)$ and n_3 corresponding desired output signals d_j $(j=1,2,... n_0)$. The input/output pairs are expressed as stable states of neurons which are usually represented by +1 (ON) and -1 (OFF). Learning of the MLP for a specific task is equivalent to finding the values of all synaptic weights such that the desired output is generated for the corresponding input. More precisely, learning of the MLP consists in adjusting all weights such that the error measure between the desired output signals d_{jp} and the actual output signals y_{jp} averaged over all learning examples p will be minimal (possibly zero). The standard back-propagation algorithm uses the steepest-descent gradient approach to minimize the mean squared error function. Such a (local) error function for the p-th learning example can be formulated as

$$E_p = \frac{1}{2}\sum_{j=1}^{n_3}(d_{jp} - y_{jp})^2 = \frac{1}{2}\sum_{j=1}^{n_3} e_{jp}^2 \qquad (6.16)$$

and the global(total) error function (also called the performance function) as

$$E = \sum_p E_p = \frac{1}{2}\sum_p \sum_j (d_{jp} - y_{jp})^2, \qquad (6.17)[\text{Cichocki and Unbehauen 1994}]$$

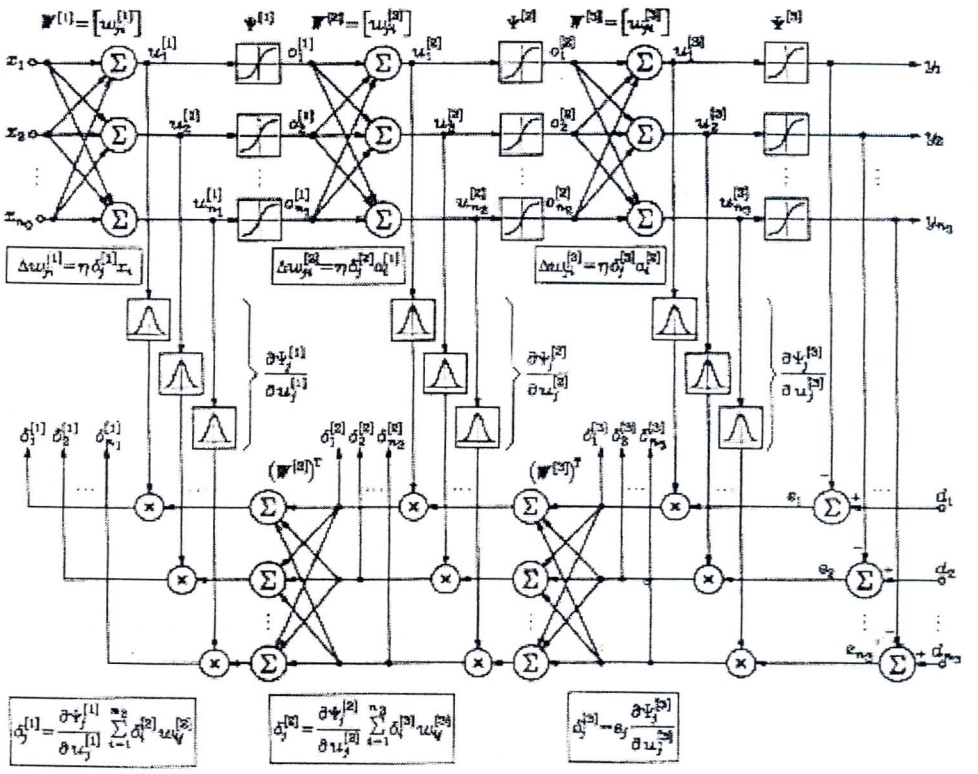


Figure 6.4. Network architecture for the standard back-propagation algorithm of a three-layer perceptron(the external biases are not shown explicitly in the figure) [Cichocki and Unbehauen 1994]

where d_{jp} and y_{jp} are the desired and actual output signal of the p^{th} output neuron for the p^{th} pattern (learning example), respectively.

There are two basic approaches to find the minimum of the global error (performance) function E. The first technique is the on-line or per example learning in which the training patterns are presented sequentially, usually in random order. For each learning example the synaptic weights $w_{ij}^{[s]}$ (s=1,2,3) are changed by an amount $\Delta w_{ji}^{[s]}$ is proportional to the respective negative gradient of the local error function E_p which can be written mathematically as

$$\Delta w_{ij}^{[s]} = \eta \frac{\partial E_p}{\partial w_{ij}^{s}}, \eta > 0$$

It has been proved that, if the learning parameter η is sufficiently small, this procedure minimizes the global error function $E=\Sigma E_p$. Of course, the above discrete gradient descent rule can be replaced by the continuos differential equations

$$\frac{dw_{ij}}{dt} = -\mu \frac{\partial E_p}{\partial w_{ij}^{[s]}}, \mu > 0.$$

Such equations can be realized by analog circuits or solved by applying more sophisticated iterative algorithms. In the second approach, sometimes called "batch learning", the total error function E is minimized in such a way that the weight changes are accumulated over all learning examples before the weights are actually changed.

At first we shall discuss the on-line learning approach in which the gradient search in the synaptic weight space is carried out on the basis of a local error function E_p . Let us first determine an updating formula for the synaptic weights $w_{ji}^{[s]}$ (s=3) of the output layer. Using the chain rule for Eq. 6.16 we can write

$$\Delta w_{ij}^{\ 3} = -\eta \frac{\partial E}{\partial w_{ij}^{[3]}} = -\eta \frac{\partial E_p}{\partial u_j^{[3]}} \frac{\partial u_j^{[3]}}{\partial w_{ji}^{[3]}}. \tag{6.18}$$

Taking into account that,

$$u_j^{[3]} = \sum_{i=1}^{n_3} w_{ij}^{[3]} x_i^{[3]} = \sum_{i=1}^{n_3} w_{ji}^{[3]} x_i^{[2]} \tag{6.19}$$

and defining the local error, called delta, by

$$\delta_J^{[3]} := -\frac{\partial E_P}{\partial u_j^{[3]}} = -\frac{\partial E_p}{\partial e_{jp}} \frac{\partial e_{jp}}{\partial u_j} = e_{jp} \frac{\partial \Psi^{[3]}}{\partial u_j^{[3]}} \tag{6.20}$$

we obtain a general formula for updating the weights in the output layer as

$$\Delta w_{ji}^{[3]} = \eta \delta^{[3]} x_i^{[3]} = \eta \delta^{[3]} o_i^{[2]}, \tag{6.21}$$

where $\delta_j^{[3]} = e_{jp} (\Psi_j^{[3]})^I = (d_{jp} - y_{jp}) \frac{\partial \Psi^{[3]}}{\partial u_j^{[3]}}$. Updating the synaptic weights in the hidden layers is a little more complicated. For the second hidden layer we can still write

$$\Delta w_{ji}^{[2]} = -\eta \frac{\partial E_p}{\partial w_{ji}^{[2]}} = -\eta \frac{\partial E_p}{\partial u_{ji}^{[2]}} \frac{\partial u_j^{[2]}}{\partial w_{ji}^{[2]}} = \eta \delta_j^{[2]} x_i^{[2]} = \eta \delta_j^{[2]} o_i^{[1]} \tag{6.22}$$

where the local error for the second hidden layer is defined as

$$\delta_j^{[2]} = -\frac{\partial E_p}{\partial u_j^{[2]}} (j = 1,2,.....n_2). \tag{6.23}$$

However, this local error cannot be directly evaluated as is done for the local errors in the output layer. Instead, we try to express it in terms quantities(signals) which are

already known, and other quantities which are easily evaluated. Using the chain rule we can write

$$\delta_j^{[2]} = -\frac{\partial E_p}{\partial u_j^{[2]}} = -\frac{\partial E_p}{\partial o_j^{[2]}} \frac{\partial o_j^{[2]}}{\partial u_j^{[2]}} \tag{6.24}$$

Taking into account that

$$o_j^{[2]} = \Psi_j^{[2]}(u_j^{[2]}). \tag{6.25}$$

we have

$$\delta_j^{[2]} = -\frac{\partial E_p}{\partial o_j^{[2]}} \frac{\partial \Psi^{[2]}}{\partial u_j^{[2]}} \tag{6.26}$$

The factor $-\partial E_j/\partial o_j^{(2)}$ can be evaluated as

$$-\frac{\partial E_p}{\partial o_j^{[2]}} = -\sum_{i=1}^{n_3} \frac{\partial E_p}{\partial u_i^{[3]}} \frac{\partial u_i^{[3]}}{\partial o_j^{[2]}} = \sum_{i=1}^{n_3} (-\frac{\partial E_p}{\partial u_i^{[3]}}) \frac{\partial (\sum_{k=1}^{n_3} w_{ik}^{[3]} x_{ik}^{[3]})}{\partial o_j^{[2]}} = \sum_{i=1}^{n_3} \delta_i^{[3]} \frac{\partial \left(\sum_{k=1}^{n_3} w_{ik}^{[3]} o_k^{[2]} \right)}{\partial o_j^{[2]}} = \sum_{i=1}^{n_{3\delta}} \delta_i^{[3]} w_{ij}^{[3]} \tag{6.27}$$

Thus the local error in the second layer can be evaluated by using the formula

$$\delta_j^{[2]} = \frac{\partial \Psi^{[\,]}}{\partial u_j^{[2]}} \sum_{i=1}^{n_3} \delta_j^{[3]} w_{ij}^{[3]}. \tag{6.28}$$

Analogously, we can derive an updating for the first hidden layer as

$$\Delta w_{ji}^{[1]} = \eta \delta_j^{[1]} x_i^{[1]} = \eta \delta_j^{[1]} o_i^{[0]} = \eta \delta_j^{[1]} x_i, \tag{6.29}$$

where the local error is determined as

$$\delta_j^{[1]} = \frac{\partial \Psi_j^{[1]}}{\partial u_j^{[1]}} \sum_{i=1}^{n_2} \delta_i^{[2]} w_{ij}^{[2]} \tag{6.30}$$

Generally the local error of the internal (hidden) layer is determined on the basis of the local errors at an upper layer. Starting with highest output layer we compute $\delta_j^{[3]}$ using formula(6.17), next we can propagate the error $\delta_j^{[3]}$ backward to the lower layers. Figure 6.4 shows function scheme (the diagrammatic representation) of the back-propagation algorithm. Such a diagrammatic representation provides not only a topological and visual representation of the entire learning procedure, but also suggests novel modifications as given in Figure 6.5 and generalizations of the basic algorithm which may be computationally more efficient.

This somewhat mysterious generalizing ability of the multilayer perceptron can be interpreted as follows. The multilayer perceptron performs a nonlinear mapping between an input and output space. The learning of the perceptron can be regarded as synthesizing an approximation of a multidimensional function which is performing a simple fitting operation or a hypersurface reconstruction in a multidimensional space to a finite set of data points(the training examples). Form this point of view the generalization is nothing more that interpreting the test set on the fitting(or reconstructed) hypersurface.

6.4 Back-Propagation Algorithm with Momentum Updating

The learning used in traditional back-propagation algorithm has some drawbacks. First of all, the learning parameter η should be chosen small to provide minimization of the total error function E. However, for a small η the learning process becomes very slow. On the other hand, large values of η correspond to rapid learning, but lead to parasitic oscillations which prevent the algorithm from converging to the desired solution. Moreover, if the error function contains many local minima, the network might get trapped in some local minimum, or get stuck on a very flat plateau.

One simple way to improve the standard back-propagation learning algorithm is to smooth the weight changes by overrelaxation by adding the momentum term

$$\Delta w_{ij}^{[s]}(k) = \eta \delta_j^{[s]} o_i^{[s-1]} + \alpha \Delta w_{ij}^{[s]}(k-1) \qquad (6.32\ a)$$

where $\eta > 0, 0 \le \alpha \le 1$(typically α=0.9) (s=1,2,3).

The weights are now updated using the formula

$$w_{ij}^{[s]}(k+1) = w_{ji}^{[s]}(k) + \Delta w_{ji}^{[s]}(k).$$

The second term in Eq. 6.32a is the so called momentum term which may improve the convergence rate and the steady state performance of the algorithm(by damping parasitic oscialltions). Intuitively, if the previous weight change is large, then adding a fraction of this amount o the current weight update will accelerate the convergence process.

More precisely, if we are moving through a plateau of the performance surface function then the gradient component $\partial E_p/\partial w_{ji}$ will be the same at each time step and Eq. 6.32a can be written as

$$\Delta w_{ji}(k) = -\eta \frac{\partial E_p}{\partial w_{ji}^{[s]}(k)} + \alpha \Delta w_{ji}^{[s]}(k-1) \cong -\frac{\eta}{1-\alpha} \frac{\partial E_p}{\partial w_{ji}^{[s]}(k)} \qquad (6.32\ b)$$

This means that the effective learning rate increases to the value $\eta_{ee}=\eta/(1-\alpha)$ without magnifying the parasitic oscillations.

6.5 Determination of Number of Neurons in Hidden Layers

The number of neurons to be used in the hidden layers(hidden units) is not known in advance, and is usually estimated by a trial and error approach.

One possible approach is to construct a neural network with an excessive number of hidden units, i.e. a network which is known or suspected to be larger than required is used and then some redundant units are removed during the learning process. All neurons that do not contribute to the solution or give information not essential next layer. Those could be considered as unnecessary hidden units. In order to find which neurons in hidden layers can be removed, the output of all of the hidden units is monitored and analyzed across all the training examples after the network achieves convergence. If the output of a certain hidden unit is approximately constant for all training examples this unit can be removed since it does not contribute essentially to the solution because it is acting as an additional bias to all neurons to which it feeds. Analogously, if two hidden units give approximately the same or opposite output for all the training examples, only one of these units is needed since both neurons convey the same information. After removing some hidden units which do not contribute to the solution the weights of the reduced network must be appropriately updated or the network must be trained once again to ensure the desired performance. The process of removing the redundant hidden units to produce the smallest neural network capable of performing the desired task is called pruning.

The above described technique enables us only to decrease the number of hidden units. Sometimes, at the first phase of learning it is convenient to gradually increase the number of hidden units and next, after achieving desired convergence, to try to remove some of them in order to find the minimal size of the network which performs the desired task.

The purpose of gradually increasing the number of hidden units is two-fold:

(i) The number of hidden units is not known in advance. So gradually increasing the number of units during the training process we can find the optimal number of units.

(ii) The learning process may become trapped in a local minimum or a very flat plateau. If one or more extra hidden units are added to the network it may escape from a local minimum since the error(performance) function in the weight space changes its shape.

Recently, a heuristic back-propagation algorithm by Hirose et. al. [Cichocki and Unbehauen 1994] has been proposed which varies the number of hidden units dynamically. According to this algorithm a new hidden unit is added when the network becomes trapped in a local minimum or a very flat plateau. For this purpose a total error function E is monitored and examined during the learning process. If the error function does not decrease or decreases very slowly, say less than 1% every 50 weight corrections, then the network is probably trapped in a local minimum. In this case, some hidden units are added to solve local minima problem.. The initial values of the weights of this unit can be set randomly in some way as in the standard back-propagation algorithm.

The network is trained again and if the error function E fails to decrease by more then 1% within the next 50 weight corrections one more hidden unit is added. This procedure is repeated until the network eventually converges, i.e. the total error is less than a specified value.

In some cases, the number of hidden units becomes excessively large in the second phase of training, one can try to reduce the number of hidden units according to the technique described above as given in Siestma et. al. [Cichocki and Unbehauen 1994]. It should be pointed out that too many hidden units usually degrades the network capability to generalize, i.e. its ability to extract the essential feature from the training set which enables us to classify the pattern which has never been presented to the network before.

6.6 Premature Saturation

The first step in back-propagation learning is, of course, to initialize the network. A good choice for the initial values of the free parameters (i.e., adjustable synaptic weights and threshold levels) of the network can be of tremendous help in a successful network design. In cases where prior information is available, it may be better to use the prior information to guess the initial values of the free parameters. But how do we initialize the network if no prior information is available? The customary practice is to set all the free parameters of the network to random numbers that are uniformly distributed inside a small range of values [Haykin 1994].

The wrong choice of initial weights can lead to phenomenon known as premature saturation. This phenomenon refers to situation where the instantaneous sum of squared errors E(n) remains almost constant for some period of time during the learning process. Such a phenomenon cannot be considered as a local minimum, because the squared error continues to decrease after this period is finished. In more direct terms, the premature saturation phenomenon corresponds to a saddle point in the error surface.

At the initial stage of back-propagation learning, depending on the prevalent conditions, both unsaturated neurons and incorrectly saturated ones may exist in the output layer of the network. As the learning process continues, the synaptic weights associated with the unsaturated output neurons change rapidly, because the corresponding error signals and gradients have relatively large magnitudes, thereby resulting in a reduction in the instantaneous sum squared errors E(n). If, however, at this point in time the incorrectly saturated output neurons remain saturated for some particular training patterns, then phenomenon of premature saturation may arise with E(n) remaining essentially constant.

In Lee et al. [Haykin 1994], a formula for the probability of premature saturation in back-propagation learning has been derived for the batch mode of updating, and it has been verified using computer simulation. The essence of this formula may be summarized as follows:

1) Incorrect saturation is avoided by choosing the initial values of synaptic weights and threshold levels of the network to be uniformly distributed inside a small range of values.
2) Incorrect saturation is less likely to occur when the number of hidden neurons is maintained low, consistent with a satisfactory operation of the network.
3) Incorrect saturation rarely occurs when the neurons of the network operate in their linear regions.

For pattern-by-pattern updating, computer simulation results show similar trends to the batch of operation referred herein.

Russo [Haykin 1994] recommends an empirical formula for the initial size of the weights that helps avoid the saturation of the neurons. This formula for the initial size of the weights that helps is given below.

The initialization of the synaptic weights and threshold levels of the network should be uniformly distributed inside a small range. The reason for making the range small is to reduce the likelihood of the neurons in the network saturating and producing small error

gradients. However, the range should not be made too small, as it can cause the error gradients to be very small and the learning therefore to be initially very slow. For an asymmetric activation functions in the form of a hyperbolic tangent for which the constants a and b are specified above, a possible choice for initialization is to pick random values for the synaptic weights and threshold levels that are uniformly distributed inside the range

$$\left(-\frac{2.4}{F_i},+\frac{2.4}{F_i}\right) \qquad (6.33)\ \text{[Haykin 1994]}$$

where F_i is fan-in (i.e., the total number of inputs) of neuron i in the network; in other words, the weight initialization is done on a neuron-by-neuron basis.

Chapter 7

AN APPLICATION OF SHAPE RECOGNITION USING ARTIFICIAL NEURAL NETWORKS

7.1 Introduction

The X-ray images are helpful tools to the physician in the process of diagnosis and medical treatment. Although it is useful to detect human body's invisible parts using X-ray images, blurring and uneven illumination always occurs. It may cause loss of some meaningful part of information and failure in diagnosis process. In order to decrease such errors, some computational methods have been developed by means of image processing. To help the physician, in diagnosis and grading of vesicle ureteral reflux disease X-ray images has been chosen as a case study. The system designed for the solution of VUR disease grading is given in Figure 7.1

In order to analyze the problem discussed above, a brief description of the VUR disease must be done first. "*The vesico ureteral reflux is the declining of the complex vesicle ureteral valve mechanism which lets the urine enter into the bladder but it stops the urine go backwards*". [Cichocki and Unbehauen 1994]

7.2 Medical Imaging

We have utilized VUR medical images as input for our image recognition system. The thesis covers recognition of the grade of vesicle ureteral reflux [Cichocki and Unbehauen 1994] X-ray images.

The study of medical imaging is concerned with the interaction of all forms of radiation with tissue and the development of appropriate technology to extract clinically useful information from observations of this interaction. Such information is usually displayed in an image format. Medical images can be as simple as a projection or shadow

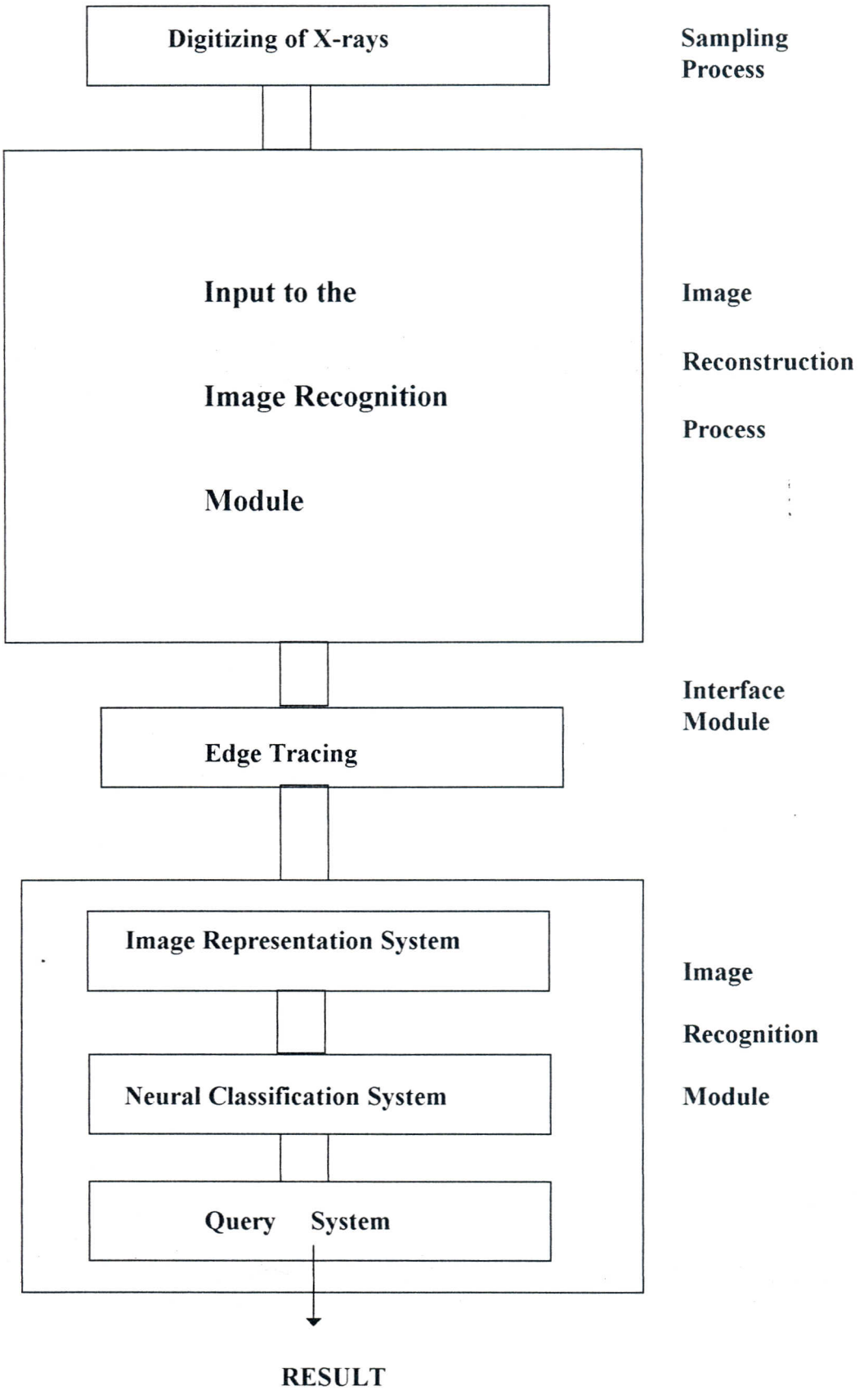


Figure 7.1 X-ray Image Recognition System Overview

image -as first produced by Roentgen nearly 100 years ago and utilized today as a simple chest X-ray - or as complicated as a computer reconstructed image- as produced by computerized tomography (CT) using X-rays or by magnetic resonance imaging (MRI) using intense magnetic fields, [Jain et. al. 1995].

Classical, medical imaging utilizes images that are a direct manifestation of the interaction of some form of radiation with tissue. Three examples will illustrate what we mean by classical imaging. The first one is the conventional X-ray procedure in which a beam of X-rays is directed through the patient onto a film. The developed film provides a shadow the image of the patient which is the direct representation of the passage of X-rays through the body. Although such images are not quantitative, they do provide some measure of the attenuation of X-rays in tissue. Thus, a section of soft tissue will appear darker than an equally thick section of bone, which attenuates more of the X-rays. It should be noted that even with the current technological developments conventional X-ray imaging still represents the major imaging procedure at most of the medical facilities[Jain et.al. 1995].

The coverage of medical imaging consist of two different processes:1. the collection of data concerning the interaction of some form of radiation with tissue, and 2. the transformation of these data into an image (or a set of images) using specific mathematical methods and computational tools. In this study we have examined the second part of the medical imaging and more.

7.3 Vesico Ureteral Reflux Disease in More Detail

Vesico ureteral reflux (VUR) is defined as retrograde passage of urine from the bladder into the ureter and renal collecting system. VUR occurs in 0.5 - 1.0% of neonates, but it is significantly more common in children who have symptomatic with urinary tract infection, lower urinary tract obstruction, or neurogenic bladder dysfunction. In over 90% of cases, incompetence of the valve-like mechanism at the ureterovesical junction results from congenital lateral ectopia of the ureteral orifice with an abnormally short submucosal ureteral tunnel. Dysfunctional voiding may contribute also. Approximately 10% of reflux is "secondary" to neurogenic dysfunction of the bladder or bladder outlet or urethral obstruction. Reflux can also result from distortion of the bladder trigone by a bladder diverticulum, ureterocele, or tumor.

Reflux can result in renal injury by two mechanisms. If a child with an acute bladder infection (cystitis) has VUR, infected urine refluxes into the kidney(s) resulting in acute renal infection (pyelonephritis). If inadequately treated, the renal infection can lead to renal parenchymal scarring and loss of renal function (chronic atrophic pyelonephritis). In children who have reflux under high pressure (e.g. children with obstruction, neurogenic dysfunction, or high grade reflux and severely dysfunctional voiding) the transmission of this high pressure into the kidney can also damage the renal tubules.

VCUG and radionuclide cystography are the only reliable examinations for diagnosing and characterizing vesicoureteral reflux. Renal injury secondary to reflux and infection is best characterized with nuclear renal scintigraphy, although RUS is used more often because it does not expose the patient to radiation and is considerably less expensive. In Figure 7.2 the cross section of kidney is shown. Reflux is most reliably diagnosed and characterized at VCUG.

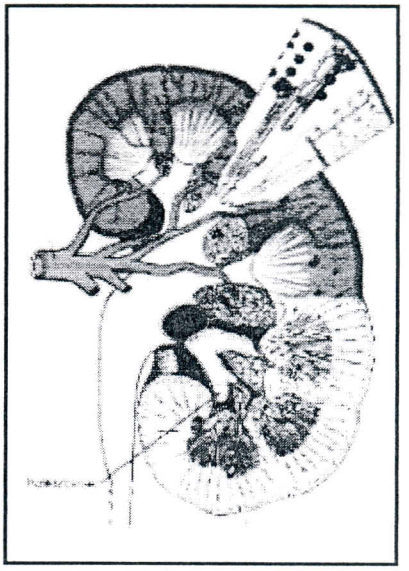


Figure 7.2 Cross section of kidney

When present reflux is graded on a scale of 1 to 5 using the system of the International Reflux Study Committee, it is constructed as follows:

Grade I: Reflux into the ureter only

Grade II: Reflux into the upper collecting system without dilatation of the calyces.

Grade III: Reflux into the upper collecting system with mild dilatation of the renal pelvis and calyces, but no dilatation of the ureter.

Grade IV: Reflux into the upper collecting system with moderate dilatation of the collecting system and ureter.

Grade V: Severe dilatation and ectasia of the ureter and dilatation of the upper collecting system with calyceal blunting.

Malfunctioning of reflux system causes some deformations on calyxes which are part of kidney and located inside of it. The disease appears as a shape changing on calyxes. Level of changing is related with grade of the disease. As shown in Figure 7.3, further grades means further deformations.

Occasionally, the collecting tubules are opacified with refluxed contrast material during VCUG. When this happens, the renal parenchyma is opacified as in this case. Whether or not pyelotubular backflow is documented at the VCUG, intrarenal reflux is presumed to be the mechanism by which infection is transmitted into the renal parenchyma in a child with bacteruria and VUR.

Bilateral Grade V Reflux With Megalocystis and Megaloureter is defined as: With very severe reflux, the bladder is refilled at the end of voiding when the refluxed urine drains back into the bladder from above. This phenomena is called "aberrant micturition." Aberrant micturition results in retention of urine within the bladder and collecting systems with increased bladder capacity (megalocystis) and chronic ureteral dilatation (megaloureter) and hydronephrosis. Over time, this predisposes not only to both urinary tract infection but also to progressive renal injury and loss of function.(this section is prepared by using extensive searching through Internet and by [Aktuğ 1995])

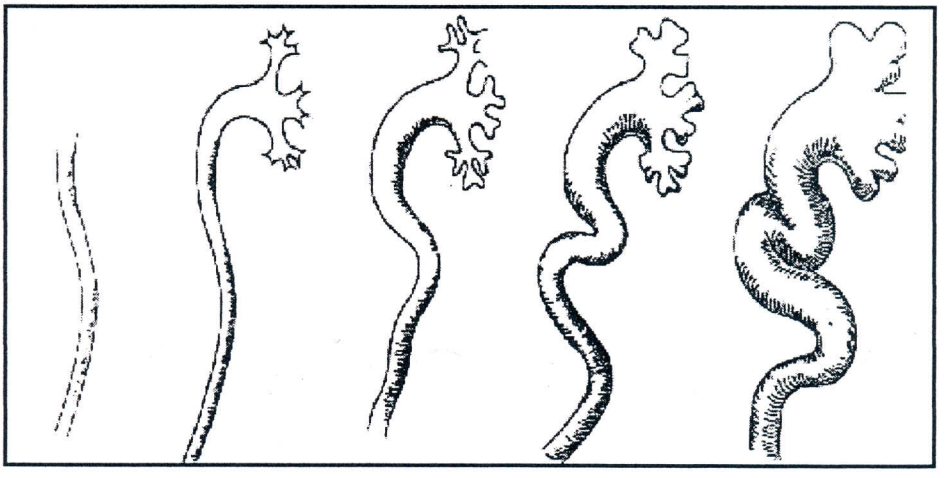


Figure 7.3 Grades of reflux as seen on the voiding cystography. The first grade starts from left end, and ends with fifth grade at the right end.

7.4 Software Prototype Developed for the Solution of VUR X-ray Image Recognition

In this part of the study, you will find information about ProVISION image recognition tool which is developed for physicians to help them in the diagnosis process. It can also be used for educational purposes in medical environment. The brief user guide of the PROVISION software is summarized in appendix B. Please refer to appendix B for further usage details and other related information. The software was developed by Cumhur Güzel and H. Cüneyt Aka. The image recognition modules is performed by Cumhur Güzel, image reconstruction modules are performed by H. Cüneyt Aka. Additionally XV software is used as viewing tool.

Some details about software:

Compiler : C programming language,

Libraries : X-Windows OSF/Motif application interface library, math libraries.

Environment : X-Windows Release 5, UNIX operating system windowing interface.

Operating System : Software is developed under the IBM AIX 3.2.5 version operating system.

The Hardware Platform: IBM RS/6000 25 T Workstation

7.5 Digitizing of X-rays

The object of this study is recognition of stages of vesico ureteral reflux disease using X-ray images mentioned as above. In medical environment, disease is detected from X-rays which is produced from human's body using the technique known as voiding cystography. Digitized form of X-ray images must be obtained in order to be able to process the X-ray images. Since X-ray images contains blurring and uneven illuminations the complexity of its processing is more than video images and photographs. In this study, X-ray films which are obtained from physicians have been digitized firstly. This is performed by a scanner which has 600 DPI resolution.

Results of digitizing via conventional scanners are not sufficient because of low level illumination. Adding some external light sources has solved the problem practically. Figure 7-4 is illustrating this process.

The processed X-rays in this study, are provided by Dokuz Eylül University, Faculty of Medicine, Department of Pediatric Surgery and the total amount of X-rays is ten. Figure 7.5 shows the results of the digitizing process.

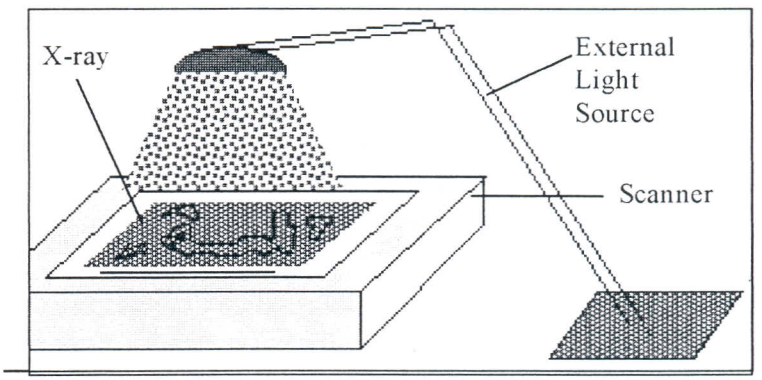
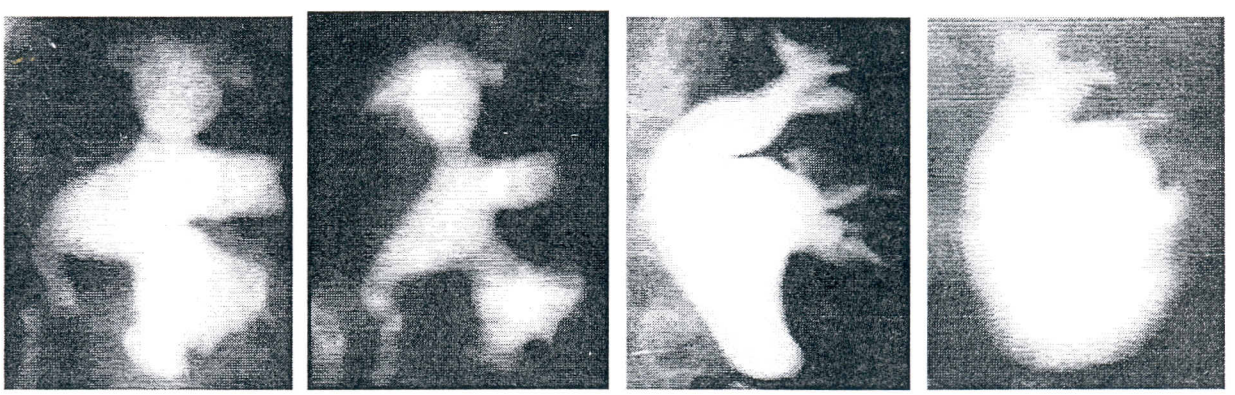


Figure 7.4 X-ray digitizing system

All of the digitized images which are shown as above have 8-bit depth. In other words, all of them are one of the combination of 256 different gray levels. Therefore, all of procedures that will be explained below has been performed on 8-bit based images.

Figure 7.5 Samples of digitized form of X-rays exposed to VUR disease



(a) **(b)** **(c)** **(d)**

Figure 7.5 Samples of digitized form of X-rays exposed to VUR disease

7.6 Image Reconstruction and Low Level Processing of Images

We have utilized the reconstructed VUR X-ray images in our study. Since we are dealing with the high level imaging in this study, we didn't considered the low level imaging techniques.

In general, low level image processing techniques are used for normalization, conversion and etc. We have utilized high level imaging techniques in this study by using reconstructed images.

The reconstruction process performed on those images in summary is given below: The related study about this subject could be found in Aka et. al [Aka et. al. 1996] as a part of our study.

Low Level Processing of Images:

1. First, variable thresholding is applied for cancellation of uneven illumination,

2. Logarithmic filtering is applied for reducing the gray-levels from 256 to 2,

3. Morphological filtering is applied for cancellation of undesirable background noises, remaining after.

4. Laplace filtering is applied for obtaining the contours on the images,

7.7 Edge Tracing

The image produced after image reconstruction process is traversed by a new edge tracing algorithm which is also developed by Güzel et. al. [Güzel et. al. 1996]. After applying the procedures applied to the image given in Figure 7.5 d, Figure 7.6 a is obtained. Reconstructed VUR images as depicted in Figure. 7.5 d, is given as input to the edge tracing algorithm. The steps of the edge tracing algorithm rotated in anti-clockwise are as follows:

1. Trace the image contours in an ordered manner (anti-clockwise) and find the first white pixel.

2. Assign the black value to the found pixel (x[m],y[m]) and save (x[m],y[m]) coordinate pairs.

3. Search the 8 neighborhoods of the found pixel

4. Set the first found neighbor as an initial point and repeat 2 and 3 steps.

5. If no white pixel is left, then stop the process.

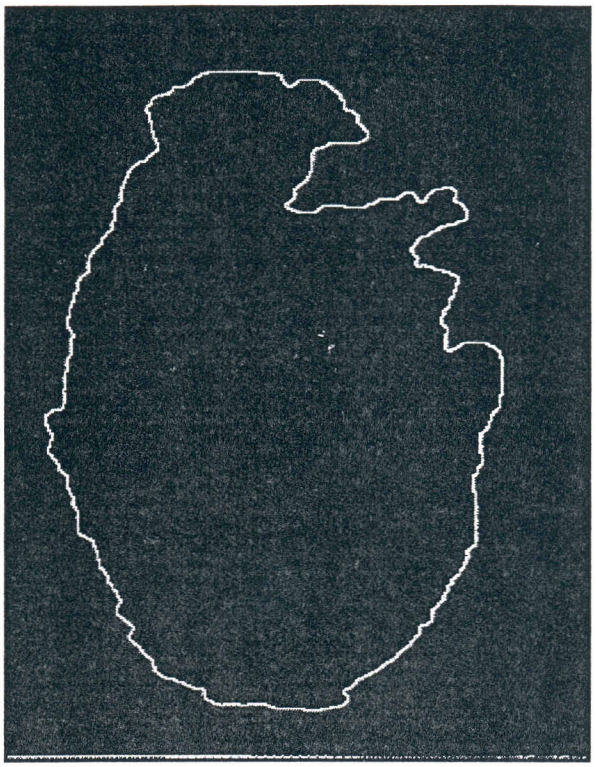


Figure 7.6a)Reconstructed X-ray image of kidney exposed to the VUR disease (taken from Figure 7.5 d)

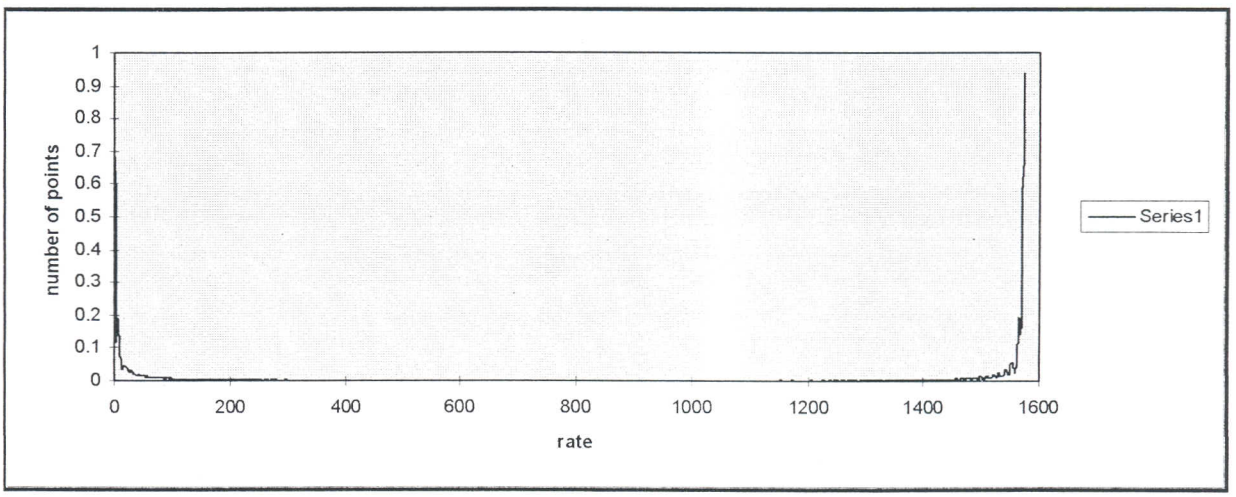


Figure 7.6 b) Fourier Descriptors obtained from the above image.

Edge tracing algorithm produces input to the image representation module of the designed system. By using counter-clockwise ordered sequence of points, we have managed to obtain the Fourier descriptors which are explained in detail in the following section.

7.8 Recognition Of Images

Recognition stage consists of two steps. First step is the representation of the shape, and the second one is the classification by a neural network classifier, as given in section 7.8.2, 7.8.3.

7.8.1 Representation of Images and Fourier Descriptors

Many representations have been proposed for shape matching. These representations can be broadly classified as contour based or region based. Contour based approaches are more popular. Most classifier designs are based on a polygonal approximation of the boundary of the shape. A localized representation of the contour is obtained by segmenting the boundary of the points that corresponds to the vertices of the approximating polygon. Generally, the vertices used are the points of maximum curvature along the contour of the object. Typical local features used are segment lengths, slopes of segments, angles between adjacent segments and a combination of these features. Typically, neural network work with vectors as input and thus the representations are also usually in vector form. Clearly, this vector representation should carry the essential shape information and must also be normalized with respect to basic shape transformations such as translation, rotation and scaling.

We have utilized the Fourier descriptors so as to represent X-ray images in our study. Using the Fourier coefficients of the boundary data, we obtain a set of Fourier coefficients that have the invariant property with respect to rotation, translation and size[Persoon and Fu 1986]. A common feature for the Fourier descriptors is that, typically, the global shape is given rather well by a few of the low-order terms[Shridhar and Badreldin 1984]. Hence, the Fourier descriptors are well suited for recognizing the objects using the characteristics of the global shape. In this study, we made feature

vectors for each phase of the X-ray image of the VUR disease by using the initial 20 Fourier descriptors.

Taking the discrete Fourier transform of the data, (x[m], y [m]) For $1 \le m \le L$, we obtain the Fourier coefficients for $0 \le k \le L-1$.

$$a[k] = \frac{1}{L}\sum_{m=1}^{L} x[m]e^{-jk(2\pi/L)m} \qquad (7.1)$$

$$b[k] = \frac{1}{L}\sum_{m=1}^{L} y[m]e^{-jk(2\pi/L)m} \qquad (7.2)$$

a [0] and b [0] are not used since they carry data about the position of the image center. Then let,

$$r[k] = \sqrt{|a[k]|^2 + |b[k]|^2} \qquad (7.3)$$

$$s[k] = \frac{r[k]}{r[1]} \qquad (7.4)$$

For $0 \le k \le L-1$, a[k], b[k], denotes the complex number of polar coordinate pairs. It is trivial that r[k] are rotation and translation invariant. To achieve scale invariance, s[k] descriptors were developed in [Shridhar and Badreldin 1984], and they are called Fourier descriptors. The graphical distribution of the Fourier descriptors obtained from the contours of Figure 7.5 d is given in Figure 7.6 b. It can be easily seen that function is periodic and the degradation in value of descriptors as "N" increases up to the period.

It should be noted that, for some l_1 , l_2 , $l_1 < l_2 < l_{(L-1)/2}$ defines the low frequency part of the image as given in [Jain et. al. 1995], [Shridhar and Badreldin 1984] $s[l_2], \ldots s[(l_{(L-1)/2}]$ defines the high frequency part of the image. The low frequency part of s[k] is determined by the overall shape of an object, while the high frequency part gives the detailed shape. Since the high frequency part is easily degraded by noise and doesn't contribute much to object recognition compared with low frequency part, we usually discard it. This compresses the data significantly, which is one of the reasons for using Fourier descriptors. In average, 1500-2000 contour points are obtained in the X-ray images we have studied. According to calculations, first 20 components (s[1]...s[20]) have sufficiently defined the images out of 2000 coefficients as feature vectors. After several calculations, it has been verified that 20 components constitutes %99 of the total

length. Total length can be calculated as in (7.5) as given in [Shridhar and Badreldin 1984].

$$Tot.Len=\sum_{i=1}^{L/2} s[i] \qquad (7.5)$$ (it is used for computation of Fourier Descriptors length)

Moreover, the use of Fourier descriptors provides the compact representation of images (compression in image size). According to the calculations performed by using (7.5), first 20 descriptors were sufficient to define average a 300 x 400 pixel images, indicating a compression ratio of up to the 20/120000 = 1/6000 compression ratio.

7.8.2 Construction of Neural Networks

The learning paradigm utilized in this study is supervised neural network learning. Feed-forward type neural network architecture is used in our design. Popular backpropagation algorithm is chosen as training algorithm as given in detail in chapter 6. Input layer is consisted of 20 nodes. There are also two hidden layers and output layer. First hidden layer contains 25 nodes and second hidden layer contains 15 nodes. All nodes are fully connected to each other. Each node has an offset value. Sigmoid function is used as activation function. The method proposed by D.E Rumelhart [Rumelhart et. al 1986] is used in determining offset values. Output layer is consisted of 5 nodes.

7.8.3 Training of Neural Network and Recognition

The initial weights and offset values of neural network are assigned between (-0.5, 0.5) with random values. Momentum term is used in training. Training of target vectors is performed by the vectors given below.

Target output vector for first phase type images:

$(t_1,t_2,t_3,t_4,t_5) = (0.98, 0.02, 0.02, 0.02, 0.02)$.

Target output vector for second phase type images:

$(t_1,t_2,t_3,t_4,t_5) = (0.02, 0.98, 0.02, 0.02, 0.02)$

Target output vector for third phase type images:

$(t_1,t_2,t_3,t_4,t_5) = (0.02, 0.02, 0.98, 0.02, 0.02)$

Target output vector for fourth phase type images:

$(t_1, t_2, t_3, t_4, t_5) = (0.02, 0.02, 0.02, 0.98, 0.02)$

Target output vector for fifth phase type images:

$(t_1, t_2, t_3, t_4, t_5) = (0.02, 0.02, 0.02, 0.02, 0.98)$

Error rate is reduced to 0.0199 after 2000 iterations. Learning rate is chosen as 0.4 and momentum value is chosen 0.3. The output vectors for neural network are y_1, y_2, y_3, y_4, y_5.

Test vectors ($out_1, out_2, out_3, out_4, out_5$) are used in the classification of the test images.

if $y_i > 0.7$ then $out_i = 1$,

if $y_i < 0.3$ then $out_i = 0$,

if $0.3 < y_i < 0.7$ then out_i = unknown value

if output vector is , ($out_1, out_2, out_3, out_4, out_5$) = (1,0,0,0,0) then shape is first phase,

If output vector is, ($out_1, out_2, out_3, out_4, out_5$)= (0,1,0,0,0) then shape is second phase

if output vector is , ($out_1, out_2, out_3, out_4, out_5$)= (0, 0, 1, 0 ,0) then shape is third phase

if output vector is , ($out_1, out_2, out_3, out_4, out_5$)= (0, 0, 0, 1 ,0) then shape is fourth phase,

if output vector is , ($out_1, out_2, out_3, out_4, out_5$)= (0, 0, 0, 0 ,1) then shape is fifth phase

if any unknown value is set in out_i then image is classified as unknown.

7.9 Experimental Results

We compared the performance of the back-propagation neural-network with nearest-neighbor rule and minimum-mean-distance classifiers.[Khotanzad and Lu 1990] and [Murphy 1992]. Each feature vector contains 20 elements according to the calculations. The example pattern set used in training of neural-network is also used in training of these classifiers. We have five classes and 20 example patterns for each class. We denote $k^{th.}$ feature vector $t_k^i = \left[t_{k1}^i, t_{k2}^i, \ldots t_{k20}^i\right]^T$ in class i. For $1 \le m \le 20$, we also denote t_m^{-i}, σ_{tm}^i the mean value and standard deviation of the m^{th}. components of the feature vectors in class i, respectively. Further, we let $t^{-i} = \left[t_1^{-i}, t_2^{-i}, \ldots t_{20}^{-i}\right]^T$ denote the mean feature vector of class i. The distance between a test pattern $X = [x_1, \ldots x_{20}]$ and t_i^k, is obtained by,

$$d(X, t_i^k) = \sum_{m=1}^{20} \frac{\left| x_m - t_{km}^i \right|}{\sigma_{tm}^i} \qquad (7.6)$$

With the nearest-neighbor rule, the class label i is assigned to X if feature vector in class i has the minimum distance from X among all the training patterns, i.e.,

$$X \in class \ \text{i} \quad \text{if } d(X,t_k^i) \le d(X,t_p^j) \ \text{ for } 1 \le j \le 5, 1 \le k, p \le 20 \quad (7.7)$$

But with the minimum-mean-distance rule, class label i is assigned to X if the mean feature vector of class i has the minimum distance from X among all mean feature vectors, i.e.,

$$X \in class \ \text{i} \ \text{,if } d(X,t^i) \le d(X,t^j) \quad \text{for } 1 \le j \le 5 \quad (7.8)$$

For the evaluation of the three classifiers and enriching the training set, we have created another 90 images by varying position and scale from the original 10 images. Using different transformations and addition of noise, the number of training set has increased to 100 images, so that there were transformed images for each one of the original image. By this way the number of training set has been reached to 100 images (20 X-ray image for each grade, and 10 variations of each image). Experiments for testing the recognition rate was performed as follows: Firstly 90 images were given to the system for training and then remaining 10 images were used for classification, rotating the test set until all images tested. Then recognition rate was recorded. Each time different ten images are tested for each grade.

Table 7.1 shows the recognition results of the three classifiers. In the case of neural network, the rate of accurate classification was 98 % on the average, while those of two statistical classifiers were below 98%.

Table 7.1 The performance evaluations of recognition rates of the neural network and two traditional statistical classifiers

Phase	Neural Network Classifier	NNR Classifier	MMDR Classifier
1. phase	20 / 20 (100 %)	20 / 20 (100 %)	20 / 20 (100 %)
2. phase	18 / 20 (90 %)	18 / 20 (90 %)	18 / 20 (90 %)
3. phase	20 / 20 (100 %)	18 / 20 (90 %)	18 / 20 (90 %)
4. phase	20 / 20 (100 %)	18 / 20 (90 %)	19 / 20 (95 %)
5. phase	20 / 20 (100 %)	18 / 20 (90 %)	19 / 20 (95 %)
Average Recognition Rate	98 / 100 (98 %)	92 /100 (92 %)	94 / 100 (94 %)

Inherently, the minimum-mean-distance and nearest neighbor rule do not have the ability of telling unfamiliar shapes from the trained ones. That is, even for an entirely new object, both rules label it as the one that is closest according to (7.7), (7.8), but the neural network can recognize unfamiliarity. We also have made experiments with new objects. According to the neural network output, it tells us that those objects are not in the training classes. In other words if you feed different image other than VUR shape, the system recognizes that given image is not VUR image.

Chapter 8

CONCLUSION AND FUTURE WORKS

8.1 Discussions

We have utilized the Fourier descriptors so as to represent X-ray images in our study. A common feature for the Fourier descriptors is that, typically, the global shape is given rather well by a few of the low-order terms. Hence, the Fourier descriptors are well suited for recognizing the objects using the characteristics of the global shape. If the local details of the shape determines the shape characteristics, this method is not suitable. According to the problem we have faced, Fourier Descriptors may be utilized as feature descriptors. Since, in VUR classification problem it is useful, we have used Fourier descriptors, as shape representation tool. Furthermore, better results may be obtained, if Fourier-Mellin transform [Chen et. al. 1991], image moments or image moments are used as shape representation tool. However, Fourier -Mellin transform is not used in this study, because it is very much computationally intensive when we compare with Fourier descriptors. Image moments are not used because they are also very much computationally intensive and the main approach in diagnosis in VUR is contour based not region based. So, according to the conditions, Fourier descriptors are the one of best tools, that provides our requirements.

VUR phases are very much similar to each other. Since feed-forward neural network trained by back-propagation algorithm is successful in performing very much complex decision region discrimination, it is chosen as classifier. Performance of other classifiers (k-NN rule and minimum mean distance rule classifiers) are experimented in chapter 7 and given in Table 7.1. Neural network classifier gives better results than others. Bayes based classifiers may be experimented in future studies for obtaining better results, if possible. Since we have designed and worked on the neural network based classifier, Bayes based classifiers were not experimented in this study.

In this project we have the problem of shortage of sample size. Furthermore, low quality X-ray images also brought in extra reconstruction problems. We had ten original training images. To increase the training set to 100 images, original 10 images have been

modified by transformation and by adding noise. If there is a possibility of having big training set, success ratio may be increased.

8.2 Conclusions

Various 2-D image classification techniques are investigated, assessed and a computational method to classify the 2-D X-ray images is developed and evaluated. As an application, those techniques have been applied for the classification of the VUR (vesicle ureteral reflux) images. In this study, a hybrid method is proposed. In summary, feature extraction is performed, and extracted features are used as pattern vectors for training neural network. Representation of X-ray images is performed by using Fourier descriptors which provides PSRI property in vector form. Feed-forward neural network is trained by using these vectors as input. Backpropagation is chosen as training algorithm. After training is finished, system is ready for questioning.

X-ray images of five phases of VUR disease are used in our experiments. The results showed that neural network is more successful than traditional conventional nearest-neighbor rule and minimum mean distance rule classifiers. Additionally, neural network also provides the capability of discrimination images which is not of trained type. We have designed the neural network classifier so that, if unknown shape is input to the system, it has the ability to tell that the shape is in unknown class. (unknown shapes)

Using of Fourier descriptors as shape representation tool provides compact representation of images. Initial 20 descriptors are enough to define average 300 x 400 pixel images. Although it is lossy compression, its compression ratio is enormous (up to the 20/120000 = 1/6000 compression ratio).

The proposed method may be used as a general shape recognition tool for the 2-D image classification of X-ray images. Extensive experiments could be done for the validity of the method for other type of X-ray image classification problems.

Additionally, proposed method may be utilized as general object recognition system for the 2-D image classification, in industrial environments, target recognition etc. after some small modifications.

Medical imaging software is developed and newly designed extensions are added to the system on RS/6000 Workstation under the AIX 3.2.5 Operating System with an X-

Windows Motif interface which provides ease of use to the physicians. The usage details of software may be found in appendix B.

8.3 Current Design vs. Future Studies

In medical environment, the X-ray images are very much helpful to the physician in the process of diagnosis and medical treatment. We are planning to implement an integrated approach in retrieving, reconstructing, storing, and querying images that are obtained from noisy X-rays as future work of this thesis. We are planning to construct multimedia archiving system for VUR X-rays. The implementation will be performed in future.

Storing the huge amount of data is an another problem in our area of interest, because of image characteristics. There are so many proposed methods such as Discrete Cosine Transform, Discrete Fourier Transform , etc., which are now being used . These are both subdivisions of *digital image compression*[Rabbani and Jones 1991].

Digital image compression techniques consider every kind of image as an object with a consistent invariant approach. This makes a very flexible environment for researchers in many cases. Besides its advantages, it has also some drawbacks. All kinds of image formats (compressed or uncompressed) have their own header information about the image. This results with a considerable loss of space on storage device. Another disadvantage of using image compression is that, there is usually data loss during the compression process. When we consider the problem from our point of view, it is not very suitable to use digital image compression approaches, if we are seeking for best results. Returning to our case, every kidney image consists of nearly 120.000(around 300x400) pixel and a 1 Kb header (for bitmaps).With a rough estimation, 121 Kb storage device space must be used for every image. This means that at least approximately 250 Kb disk space is needed for each kidney of each patient. However, in our case, the boundaries of kidney region are sufficient for diagnosis. In other words, storing the boundaries instead of complete image has the same precision. Under the light of this approach, we have assumed that if we detect and store the kidney's boundary coordinates on both x and y axis, then we can easily reconstruct the same image whenever we want. Although this was sufficient for our study, we have decided to develop a much

more flexible file format by ordering x and y coordinate couples in anti-clockwise direction with the same information for further studies such as computer aided diagnosis systems.

In order to perform the procedures summarized as above, we have developed an algorithm which has two distinct parts, one of them is edge detection and the other is edge tracing. We have used Laplace edge detection filter for obtaining the boundaries with one-pixel thickness. The image produced by Laplace Edge Detector[Jain et. al. 1995] is traversed by a new edge tracing algorithm which is also developed in this study.

The image produced via Laplace edge detector reduces the image size to around 2000 pixels, which provides an enormous ratio as 2000/120000. Although this is a lossy compression, reconstructed images are fully sufficient and useful for diagnosis.

Additionally, we are planning to implement the relational patient database with an X-ray image storage and retrieval which provides historical information about patient and some queries about patient and intelligent queries which use image recognition techniques about patient disease. Moreover, intelligent queries will provide multimedia database query techniques which are useful in querying different image features. Content similarity based queries might be implemented in this multimedia database for the X-ray images. Artificial neural network based system will be database engine for querying the similarity based image content queries.

The planned system may be used as a general multimedia archiving system for the 2-D X-ray images. Extensive experiments and some modifications to planned system could be done for the validity of the method for other type of X-ray images.

The planned system may be used as a general multimedia archiving system for the 2-D objects. Extensive experiments and some modifications to planned system could be done for the validity of the method for other type of 2-D image of objects.

References

1. **[Aktuğ 1995]** T. Aktuğ, "Çocuk Cerrahisi", Güleç Kırtasiye ve Matbaacılık, 1995.

2. **[Jain et. al. 1995]** Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, "Machine Vision", McGraw-Hill, 1995.

3. **[Shridhar and Badreldin 1984]** M. Shridhar and A. Badreldin, "High accuracy character recognition algorithm using Fourier and topological descriptors", Pattern Recognition, vol. 17, no. 5, pp.515-524, 1984.

4. **[Kim and Nam 1995]** H. Kim, K. Nam, "Object Recognition of One-DOF Tools by a Back-Propagation Neural Networks", IEEE Transctions on Neural Networks, vol. 6, no. 2, pp.484-487, March 1995.

5. **[Ghorbel and Tocnaye 1990]** F. Ghorbel and J.L. Bougrenet De La Tocnaye, "Automatic control of lamellibranch larva growth using contour invariant feature extraction", Pattern Recognition, vol.23, No. 3 / 4, pp.319-323, 1990.

6. **[Shen et. al. 1994]** Liang Shen, Rangaraj M. Rangayyan and J. Leo Desautels, " Application of shape analysis to mamographic calcifications", IEEE Transactions on Medical Imaging, vol.13, no.2, pp. 263-274, June 1994.

7. **[Pao 1989]** Yoh-Han Pao, "Adaptive Pattern Recognition and Neural Networks", Addison-Wesley Publishing Company Inc., 1989.

8. **[Rumelhart et. al. 1986]** D. E. Rumelhart, J.L. McClelland, and the PDP Research Group, Eds., "Parallel Distributed Processing", vol. 1, and 2. Cambridge, MA: MIT Press, 1986.

9. **[Fukunaga 1990]** K.Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press,, 1990.

10. **[Lippmann 1992]** R. P. Lippmann, "An Introduction to Computing with Neural Nets", (The name of the article in the book) Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau , pp. 1-23, IEEE Press, 1992.

11. **[Cichocki and Unbehauen 1994]** A. Cichocki and R. Unbehauen, "Neural Networks for Optimization and Signal Processing", John Wiley and Sons, pp.100-130, 1994.

12. **[Hopfield 1992]** J.J. Hopfield, "Neural Networks and physical systems with emergent collective computatioanal abilitiies", (The name of the article in the book), Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau , pp. 74-90, IEEE Press, 1992.

13. **[Kohonen 1992]** T. Kohonen,"The self organizing map", (The name of the article in the book) Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau , pp. 74-90, IEEE Press, 1992.

14. **[Cerpenter and Grossberg 1992]** G. A. Carpenter, S. Grossberg "A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine", (The name of the article in the book) Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau , pp. 147-208, IEEE Press, 1992.

15. **[Fausset 1994]** L. Fausett "Fundamentals of Neural Networks: Architectures, Algorithms and Applications", Prentice Hall, 1994.

16. **[Sejnowski and Rosenberg 1986]** T. Sejnowski, C. R .Rosenberg "NET talk: A Parallel Network that Learns to Read Aloud", Technical Report, JHU/EECS-8601, Johns Hopkins University, 1986.

17. **[Barnard and Casasent 1991]** E.Barnard and D.Casasent, "Invariance and Neural Networks, IEEE Transactions on Neural Networks", vol. 2, no. 5, pp. 498-508, September 1991.

18. **[Yüceer and Oflazer 1993]** C. Yüceer and K.Oflazer, "A rotation, scaling and translation invariant pattern classification system", Pattern Recognition, vol. 26, no. 5, pp. 687-710, May 1993.

19. **[Khotanzad and Lu 1990]** Khotanzad, A., Jiin-Her Lu, " Classification of invariant image representation using a neural network", IEEE Transactions on Acoustics Speech and Signal Processing, vol. 38, no.6, pp. 489-497, June 1990.

20. **[Li and Nasrabadi 1989]** W.Li and N.M. Nasrabadi, "Object Recognition based on graph matching implemented by a Hopfield style Neural Network", IEEE International. Joint Conference on Neural Networks, vol. 2, pp. 287-290, June 1989.

21. **[Tsao et. al. 1993]** E.C Tsao, W.Lin and C.Chen, "Constraint satisfaction Neural Networks for image recognition", Pattern Recognition, vol. 26, no. 4, pp. 553-567, April 1993.

22. **[Gupta et. al. 1990]** Lalit Gupta, M.R.Sayeh and Ravi Tamanna, "A neural network approach to robust shape classification" , Pattern Recognition, vol. 23, no. 6, pp. 563-568, June 1990.

23. **[Pal 1993]** N.R.Pal, Pratik Pal and Anupam Basu, "A new shape representation scheme and its application to shape discrimination using a neural network", Pattern recognition, vol. 26, no. 4, pp. 543-551, April 1993.

24. **[Mitzias and Mertzios 1994]** D. A. Mitzias and B.G. Mertzios, "Shape recognition with a neural classifier based on a fast polygon approximation technique", Pattern Recognition, vol. 27, no. 5, pp. 627-636, May 1994.

25. **[Gallant 1992]** S.I. Gallant , Chapter 17, "Extracting rules from networks" in "Neural Network Learning and Expert Systems", MIT Press, 1992.

26. **[Towell and Shavlik 1994]** G.G. Towell and J.W. Shavlik, "Knowledge based artificial neural networks", Artificial Intelligence vol. 70, pp. 119- 165, 1994.

27. **[Kim and Yang 1994]** H.J. Kim and H.S. Yang, "A Neural Network capable of learning and inference for visual pattern recognition", Pattern Recognition,vol. 27, no. 10, pp. 1291-1302, 1994.

28. **[Murphy 1992]** O.J. Murphy, "Nearest Neighbor Pattern Classification Perceptrons", (The name of the article in the book), Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau , pp. 263-267, IEEE Press, 1992.

29. **[Neti et. al. 1992]** C. Neti, M.H. Schneider and E.D. Young, "Maximally fault tolerant Neural Networks, IEEE Trans. on Neural Networks", vol.3 no.1, pp. 14-23, January 1992.

30. **[Jou et. al. 1994]** I.C. Jou, S.S. You and L.W. "Chang, "Analysis of hidden nodes for multi-layer perceptron neural networks", Pattern Recognition, vol. 27, no. 6, pp. 859-864, 1994.

31. **[Khunasaraphan et. al. 1994]** C. Khunasaraphan, K. Vanapipat and C. Lursinsap, "Weight shifting techniques for self-recovery neural networks", IEEE Transactions on Neural Networks, vol. 5, no. 4, July 1994.

32. **[Masters 1993]** Timothy Masters, Chapter 11, "'Interpreting the weights: How does this thing work?", Practical Neural Network recipes in C++, Academic Press, 1993.

33. **[Rauber 1986]** Thomas W. Rauber "Two Dimensional Shape Description" ,Technical Report, GR UNINOVA-RT-10-94, Nova University, Lisbon, Intelligent Robotics Center, 1986.

34. **[He and Kundu 1991]** He, Y., and Kundu , A., "2-d Shape Classification Using Hidden Markov Model," IEEE Transactions On Pattern Analysis and Machine Intelligence , vol. 13, no. 11, pp. 1172-1184, Nov. 1991.

35. **[Persoon and Fu 1986]** E. Persoon, and K. -S. Fu, "Shape discrimination using Fourier Descriptors", IEEE Transaction on PAMI, vol.8, pp.388-397, May 1986.

36. **[Li 1992]** Yajun Li, "Reforming the theory of invariant moments for pattern recognition", Pattern Recognition, vol. 25 no. 7, pp.723-730, 1992.

37. **[Massone et. al. 1985]** Massone, L., Sandini, G., and Tagliasco, V., "From - invariant" topological mapping strategy for 2D shape recognition," Computer Vision , Graphics and Image Processing, vol. 30, pp. 169-188, 1985.

38. **[Werbos 1992]** P. Werbos, "Backpropagation through time: what it does and how to do it", (The name of the article in the book) Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau , pp. 211-221, IEEE Press, 1992.

39. **[Widrow and M.A. Lehr 1992]** B.Widrow and M.A. Lehr , "30 years of adaptive neural networks: Perceptron, Madaline , and Backpropagation",(The name of the article in the book) Neural Networks: Theoretical Foundations and Analysis, edited by Clifford Lau, pp.27-53, IEEE Press, 1992.

40. **[Haykin 1994]** Simon Haykin, "Neural Networks, a comprehensive foundation", Macmillan College Publishing Company, Inc., New York, 1994.

41. **[Aka et. al. 1996]** H. C. Aka, C. Güzel, H. Püskülcü, S. Aytaç, T. Aktuğ, "Vesiko Üreteral Reflü Hastalığının Yapay Sinir Ağları Yardımıyla Sınıflandırılması", Proceedings of Biyomut- 96, 1996 Biomedical Engineering National Congress, pp. 99-106, Istanbul 21-22 October 1996.

42. **[Güzel et. al. 1996]** C. Güzel, H. C. Aka, H. Püskülcü, S. Aytaç, T. Aktuğ, "Recognition and Reconstruction of V.U.R Disease X-ray Images", Proceedings of the

Biyomed 3, Biomedical Science and Technology Symposium", pp. 91-92, Bursa, 11-14 December 1996.

43. **[Chen et. al. 1991]** Qin-sheng Chen, Michel Defrise and F. Deconick, " Symetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition" , IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.16, no.12, pp.1156-1168, December 1994.

44. **[Rabbani and Jones 1991]** Majid Rabbani, Paul W. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press, 1991.

APPENDIX A

RECONSTRUCTED SAMPLES OF VUR DISEASE AFFECTED X-RAY IMAGES

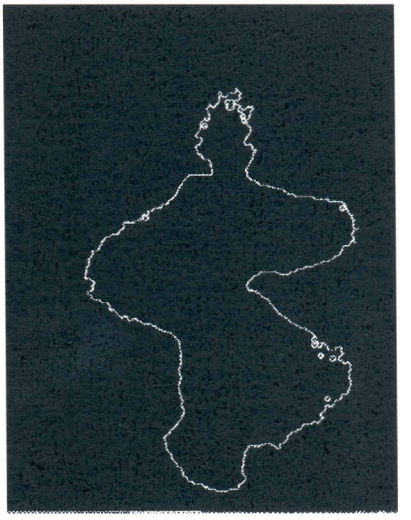


Figure A-1 VUR disease affected kidney X-ray image

Figure A-2 VUR disease affected kidney X-ray image

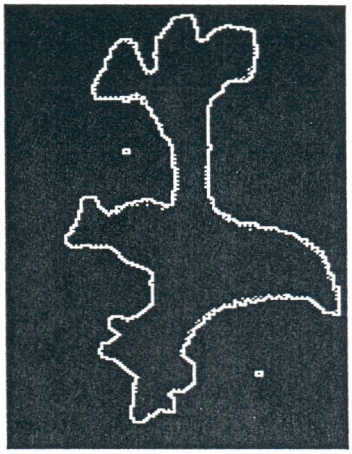


Figure A-3 VUR disease affected kidney X-ray image

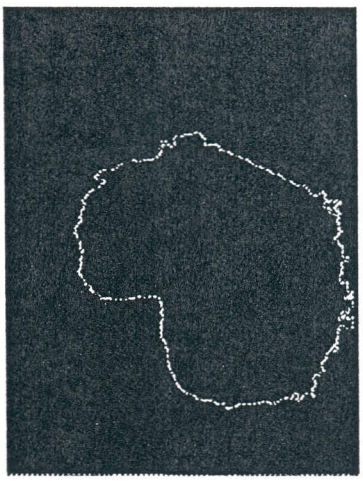


Figure A-4 VUR disease affected kidney X-ray image.

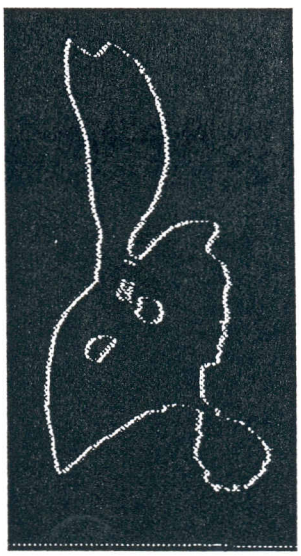


Figure A-5 VUR disease affected kidney X-ray image

Appendix B

PROVISION IMAGE RECOGNITION TOOL

Interface and brief user guide of the image reocgnition system is summarized with a screen snapshots below.

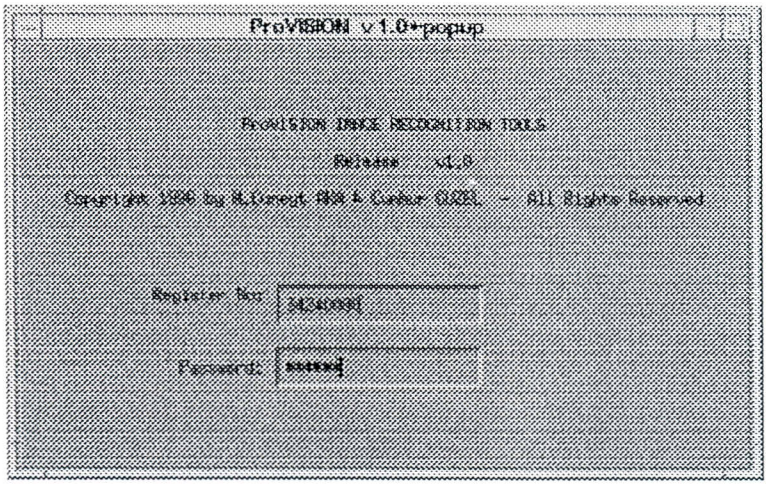


Figure B-1 **Login** frame

The security frame of the application is given in Figure A-1, above. It prevents unauthorized intruder attacks to patient database. It provides two-layered protection; Application register number and password. When a user would like to use PROVISION, he has to enter appropriate register number and password. Then usage of the program is possible.

The main frame consists of five different pull-down menu options. *File, Image, Generate Data, Recognition, Image Database*. In this appendix, File, Generate Data, Recognition, Image Database is explained. Image option is not explained, since it is about low-level image processing which is not our interest.

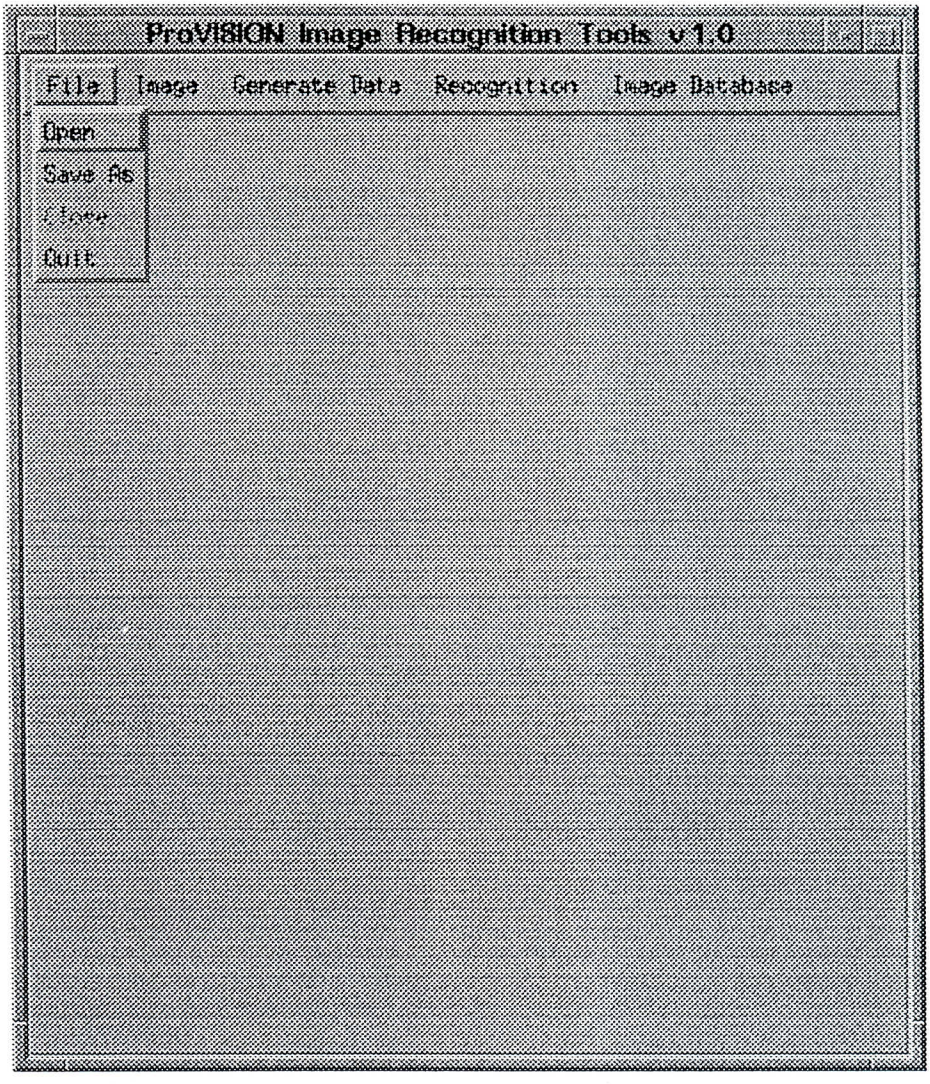


Figure B-2 Options of the **File** pulldown menu

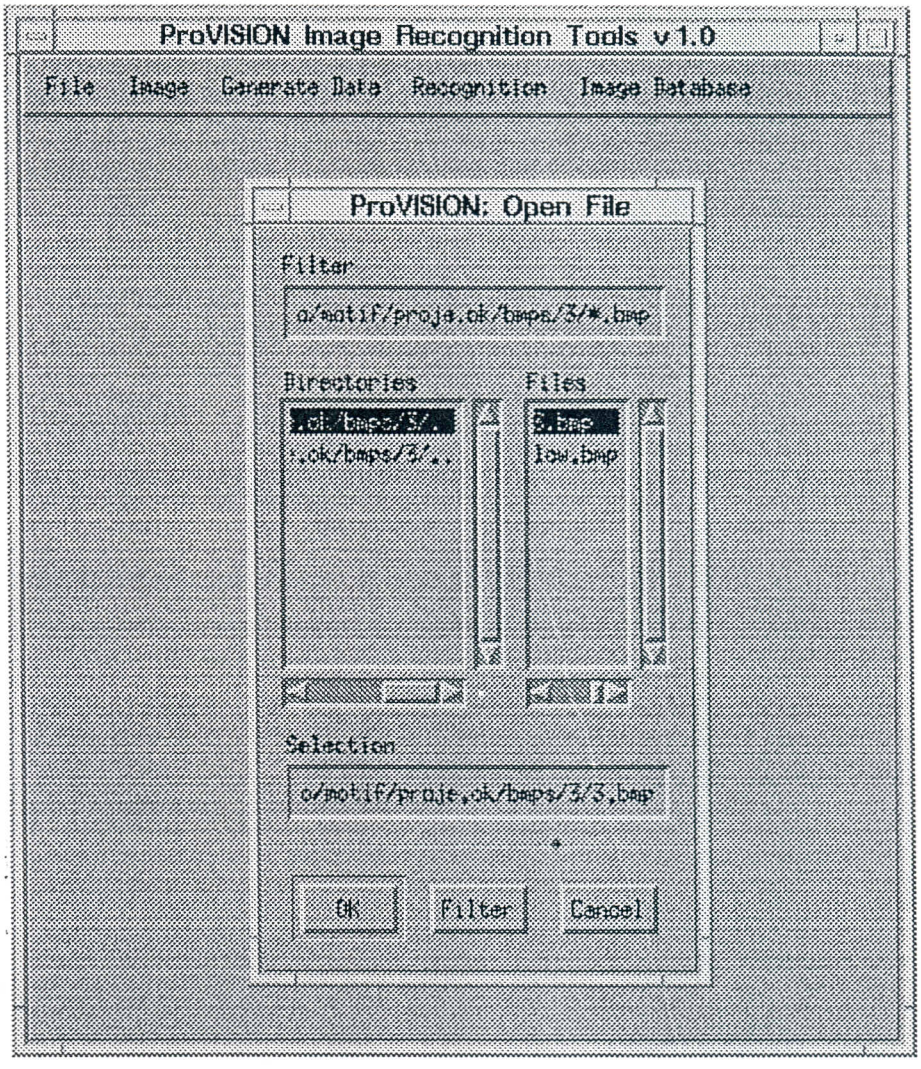


Figure B-3 **Open** popup menu

When **Open** option is chosen in File menu this popup menu appears for file selection

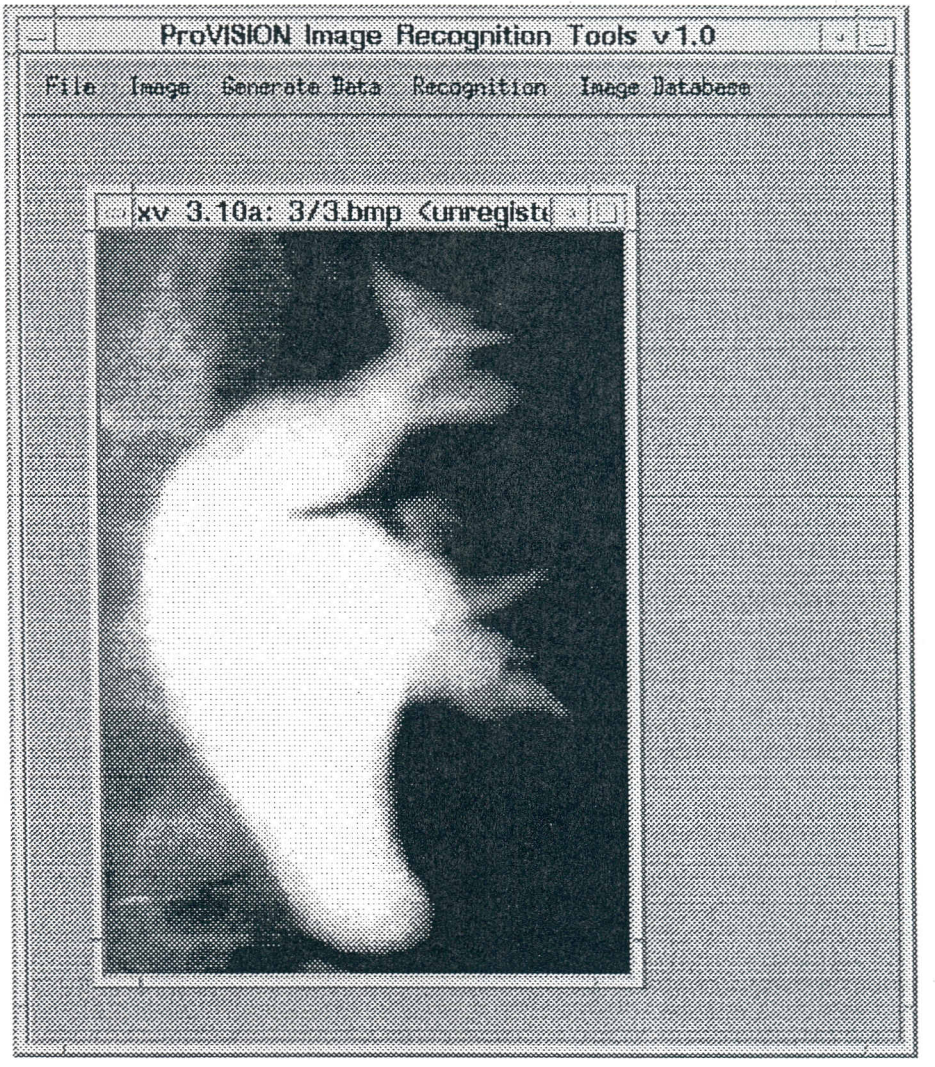


Figure B-4 Screen snapshot After **File** selection in **Open** pulldown menu

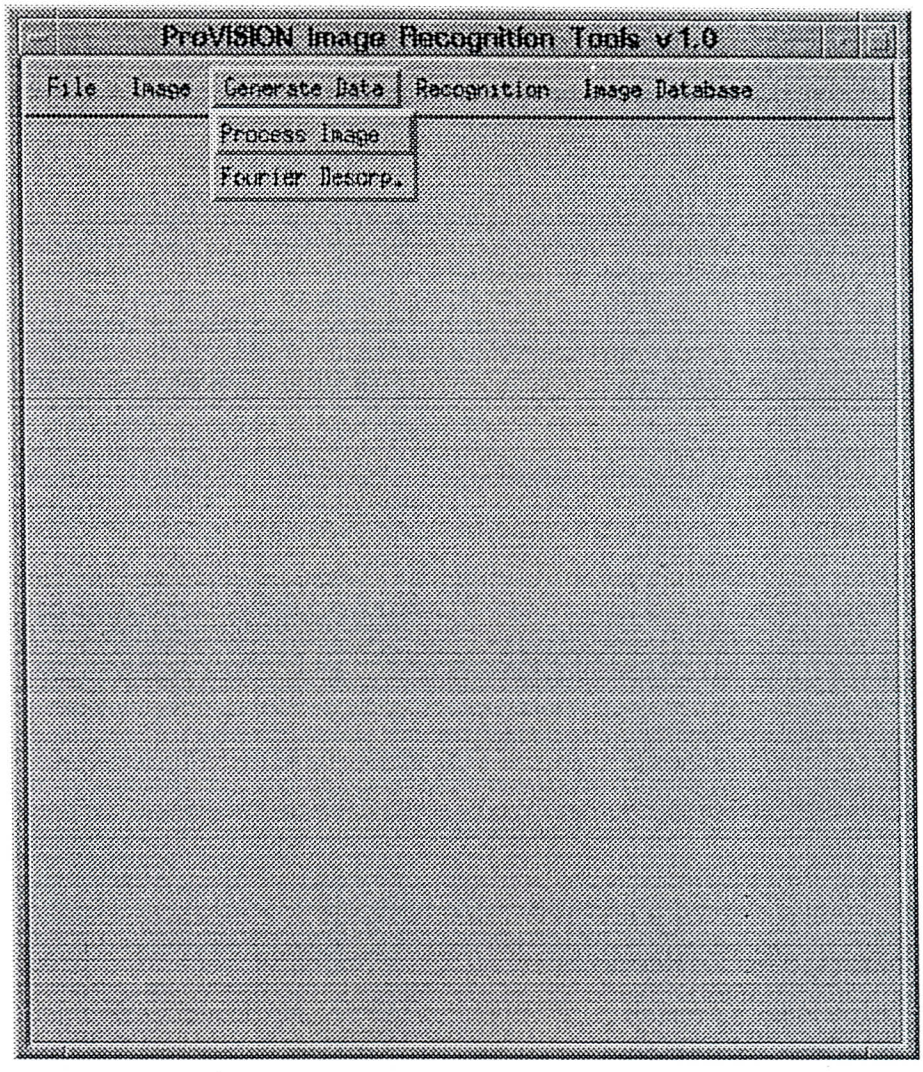


Figure B-5 Options of the **Generate Data** pulldown menu

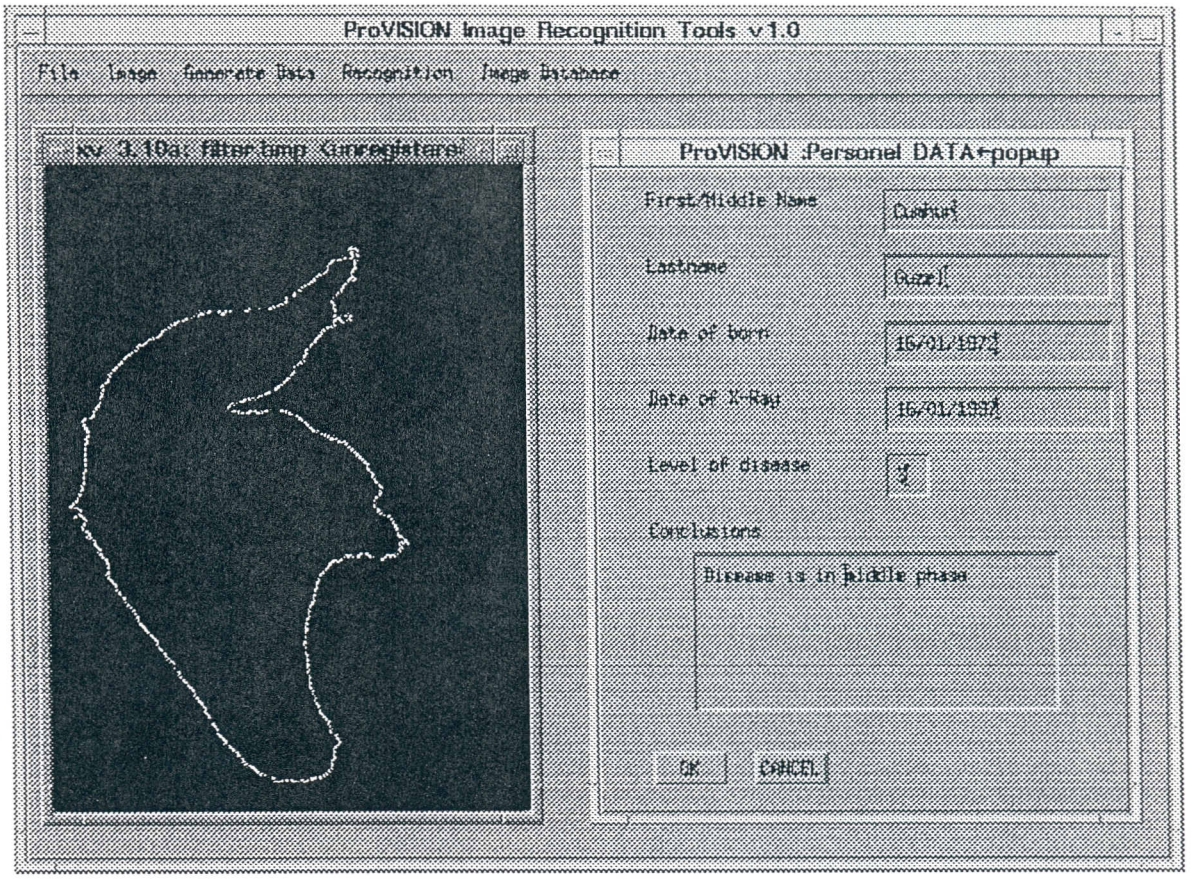


Figure B-6 Screen snapshot After **Process Image** option is chosen in **Generate Data** menu

The reconstructed image is given on the left. On the right part of the screen there is a patient information form for storing patient essential data. After OK is chosen, registration of the patient is done to the database. If Cancel is chosen nothing is stored.