SELF EVOLVING INTELLIGENT SYSTEM FOR CLASSIFYING NUMEROUS RETINAL DISEASES

by

NC Syed Muhammad Fahad Aizaz
NC Saad Arif
NC Babar Aqeel
NC Usama Akbar

Submitted to Faculty of Computer Science, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfillment for the requirements of Bachelors of Engineering Degree in, Computer Software Engineering

May 2005

ABSTRACT

SELF EVOLVING INTELLIGENT SYSTEM FOR
CLASSIFYING NUMEROUS RETINAL DISEASES

Our eyes and the parts of our brain that allow us to understand the visual information one receives from our eyes, comprise of a unique and awe inspiring sense known as sight. The retina is a complex tissue at the back of the eye that contains specialized photoreceptor cells, called rods and cones. They are connected to a network of nerve cells for the local processing of visual information.

Retina is susceptible to a variety of diseases that can lead to visual loss or complete blindness. Many a people are vulnerable to such diseases, but in the presence of more deadly diseases inflicting humanity, like AIDS, cancer etc., eye diseases receive little in research funding. This work is an effort in trying to come up with some intelligent ways of detecting diseases that is cost-effective and efficient.

The signatures for the retinal image patterns were defined and then were used to train the network. 45 retinal images were presented to a multilayer perceptron neural network, which engaged a learning process using Levenberg Manquardt algorithm. The patterns out of the image signatures were extracted by the neural network, on the basis of which it classified test images pattern that is specified the category to which that test image belongs.

The sensitivity and specificity of the recognition for the three image patterns that is Diabetic Retinopathy, AMD Macular Hole and Normal Eye sampled for classification were 100%, 83.5% and 75% respectively. Thus the overall performance regarding the correct classification attained by the system was 85.7%.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another reward or qualification either in this institute or elsewhere.

DEDICATION

In the name of Allah, the Most Merciful, the Most Beneficent

To our dear parents and siblings … especially to our Mothers

ACKNOWLEDGMENTS

After the grace of Almighty Allah and the prayers and support of our parents and family, we wish to express our sincere gratitude to our supervisor Lecturer Ahmad R. Shahid for his guidance and inspiration throughout the project. We convey our appreciation to Dr. Lt. Col. Sami-ul- Hassan, CMH Rawalpindi and Dr. Maj. Amjad Akram, MH Rawalpindi, whose technical assistance have been highly valuable. We are highly gratified to Hayatabad Medical Complex for their help and assistance in gathering the data. We would also like to extend our gratitude to Lt. Col. Naveed Khattak, whose critics and technical assistance has leaded the research towards the ultimate goal.

We would like to take this opportunity to honor our fathers and mothers, whose inspiration and support enabled us to carry out the research. To them, we give out heartfelt thanks and deepest appreciation.

Finally we would like to convey our thanks and admiration to all our classmates, friends and seniors, who had been supporting helping and encouraging us throughout the thesis project. We also wish to acknowledge MCS system administration for their help and support. I would like to express my gratitude and appreciation to Aisha Khalid Khan, for her help, support and encouragement all along.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
ANN	Artificial Neural Network
MFC	Microsoft Foundation Classes
GUI	Graphical User Interface
LM	Levenberg Manquardt
TRAINDX	Training Gradient Decent
ALVINN	Autonomous
PGM	Portable Gray Map
JPEG	Joint Photographic Experts Group
BMP	Bitmap
RNA	Ribonucleic Acid

Chapter 1

1 Introduction

A new era is dawning in the domain of biosciences. The growing use of advanced computer science is making possible the study of complex phenomena that were previously unapproachable, sometimes intractable, using conventional techniques. “Many people believe the next decade will be defined as the 'Age of Biology,' not only from the business standpoint, with huge new markets, but from a rich intellectual pursuit," said Bill Todd, IM '71, president of the Georgia Research Alliance [1].

In this early part of the 21st century, scientific discovery and understanding is playing an important and growing role in meeting the challenges— environmental, health, and economic — facing humanity. At the forefront are the advances in biological sciences. Indeed, it is reasonable to say human race are entering the ‘Age of Biology’, paralleling in many ways the ‘Age of Physics’ in the first half of the 20th century.

Researchers in biosciences and biotechnology are exploring new horizons, which promise a new age of discovery in the most important of fields. Faced with the challenges of disease, aging populations, rising healthcare costs and economic needs, these scientists are pursuing opportunities hitheroto unimagined. Working across disciplines, researchers and engineers of every stripe are seeking the fundamental secrets that may "change the pattern of life in ways comparable to the chemist's invention of antibiotics and the physicist's invention of the computer chip.” Engineers will take the discoveries of biologists, biochemists and biophysicists and turn them into devices and products that will change forever the world the way it is today.

Three broad research programs will account for much of this new biological knowledge. The first is genomics. It has already reshaped the biological inquiry in some very fundamental ways. It has totally changed the way scientists look at the individuals, and their medicinal needs. And as genomics and proteomics meld further with developmental/cell biology and the agricultural sciences, this world will be transformed in ways that may not be recognized by an early twenty-first century man. That change has already begun, but the full effects of this revolution are just beginning. Bio-technology is emerging from the embryonic into the boom cycle.

The second research area is what is known as evolutionary biodiversity science. This name describes the product of uniting evolutionary and systematic biology with the environmental sciences—namely, comparative, functional, and integrative biology whose center of gravity understands Earth's biological diversity at all levels of organization.

The third research program, namely the bioinformatics, is the glue that will bind the other research efforts together. It will create entirely new fields of research as it integrates genomics-related sciences and evolutionary biodiversity sciences into earth sciences, chemistry, and other disciplines [2].

Before embarking on an attempt to explain the nitty-gritty of the technical details of this particular project, a brief overview of the two relevant domains would be in order:

1.1 Bioinformatics

In the last few decades, advances in molecular biology and the equipment available for research in this field have allowed the increasingly rapid sequencing of large portions of the genomes of several species. Information science has been applied to biology, and as a result a new field is taking shape, namely the Bioinformatics [3].

It is the recording, annotation, storage, analysis, and searching/retrieval of nucleic acid sequences (genes and RNAs), protein sequences and structural information. The simplest tasks in bioinformatics concern with the creation and maintenance of databases of biological information, thus also making it incumbent on the informaticists to delve deeper into research in databases. Nucleic acid sequences (and the protein sequences derived from them), comprise the majority of such databases. While the storage and organization of millions of nucleotides is far from trivial, designing a database and developing an interface, whereby researchers can both access existing information and submit new entries is only the beginning.

Computational Biology involves the analysis of gene-sequence information. It comprises of finding the genes in the DNA sequences of various organisms, developing methods to predict the structure and/or function of newly discovered proteins and structural RNA sequences, clustering protein sequences into families of related sequences and the development of protein models and aligning similar proteins and generating phylogenetic trees to examine evolutionary relationships between different species.

The process of evolution has produced DNA sequences that encode proteins with very specific functions [4]. It is possible to predict the three-dimensional structure of a protein using algorithms that have been derived from our knowledge of physics, chemistry and most importantly, from the analysis of other proteins with similar amino acid sequences. The Figure 1-1 summarizes the process by which DNA sequences are used to model protein structure.

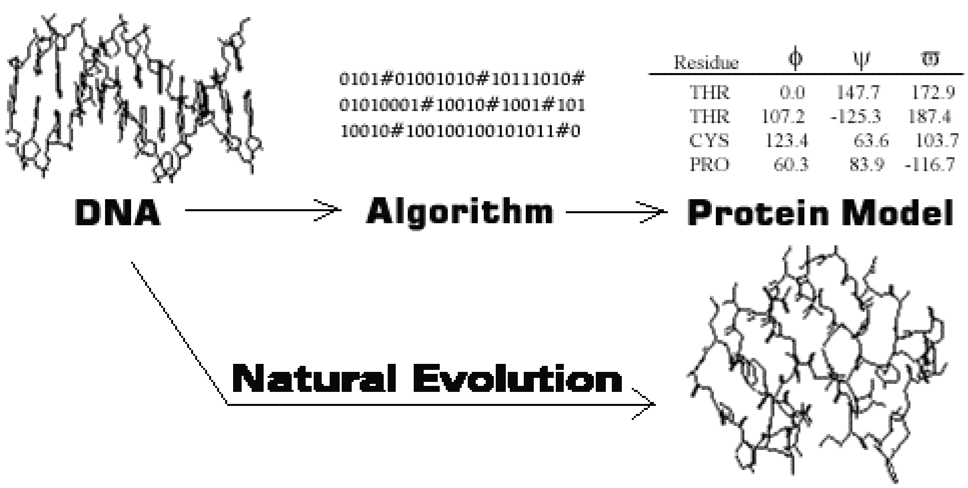

Figure 1-1: Process of Evolution [3]

1.2 Biometrics

The terms *Biometrics* or *Biometry* has been in vogue since early in the 20th century to refer to the field of development of statistical and mathematical methods applicable to data analysis problems in the biological sciences. Statistical methods for the analysis of data from agricultural field experiments to compare the yields of different varieties of wheat, for the analysis of data from human clinical trials evaluating the relative effectiveness of competing therapies for disease, or for the analysis of data from environmental studies on the effects of air or water pollution on the appearance of human disease in a region or country are all examples of problems that would fall under the umbrella of *Biometrics*, as the term has been historically used [5].

Recently, the term Biometrics has also been used to refer to the emerging field of technology devoted to identification of individuals using biological traits, such as those based on retinal or iris scanning, fingerprints, or face recognition**.** It is the science and technology of authentication (establishing the identity of an individual) by measuring the person's physiological or behavioral features.

In information technology (IT), biometrics usually refers to technologies for measuring and analyzing human physiological characteristics such as fingerprints, eye retinas and irises, voice patterns, facial patterns, and hand measurements as in Figure 1-2, especially for authentication purposes.

Figure 1-2: Eye Scan and Thumb Print [5]

Much research and development is being carried out in biometrics especially when it is utilized as a security measure at high risk areas. Examples of behavioral characteristics, which can be measured, include signature recognition, gait recognition, speaker recognition and typing recognition. Despite many misgivings, biometric systems have the potential to identify individuals with a very high degree of certainty. Forensic DNA evidence enjoys a particularly high degree of public trust at present (ca. 2004) and substantial claims are being made in respect of iris recognition technology, which has the capacity to discriminate between individuals with identical DNA.

1.3 Background

Research and Development programs in many universities around the world are being carried out in the specific area of biomedical image analysis. Experts on Computer Vision and the robust statistical analyzers are putting their efforts in this particular area and have produced many significant results.

1.3.1 The Retina Project

Professor Chuck Stewart of Computer Science Department at Rensselaer in Troy, and his team are investigating the use of computer vision techniques to aid in the diagnosis and treatment of diseases of the retina. They have focused primarily on core segmentation and registration algorithms and have produced results on these topics. Currently, they are exploring a variety of topics, especially real-time registration and the underlying operating system support, and registration-based vascular and non-vascular change detection [6].

Four algorithms have been developed as the core of spatial mapping and referencing techniques. The first is a frame-rate algorithm for finding and describing geometric properties of the retinal vasculature. The second is a hierarchical, robust algorithm for registering pairs of retinal images. The third is a joint, linear algorithm for estimating the transformations necessary to form a mosaic of many retinal images. The fourth is an invariant indexing and alignment algorithm for determining the image-to-spatial-map transformation in real-time. All of these algorithms have been successfully demonstrated on diagnostic image sequences and soon will be tested on sequences that simulate surgical conditions [7].

1.3.2 Recognition of Main Retinal Components

In the year 1999, a research paper was published by authors Chanjira Sinthanayothin, James F. Boyce, Helen L. Cook and Thomas H. Williamson; in which they aimed to recognize automatically the main components of the fundus on the digital color images.

The main features of a fundus retinal image were defined as the optic disc, fovea, and blood vessels. Methods were described for their automatic recognition and location. 112 retinal images were preprocessed via an adaptive, local, contrast enhancement. The optic discs were located by identifying the area with the highest variation in intensity of adjacent pixels. Blood vessels were identified by means of a multilayer perceptron neural network, for which the inputs were derived from a principal component analysis (PCA) of the image and edge detection of the first component of PCA. The foveas were identified using matching correlation together with characteristics typical of a fovea—for example, darkest area in the neighborhood of the optic disc. The main components of the image were identified by an experienced ophthalmologist for comparison with computerized methods.

In this study, the optic disc, blood vessels and fovea were accurately detected. The Identification of the normal components of the retinal image would aid the future detection of diseases in these regions. In diabetic retinopathy, for example, an image could be analyzed for retinopathy with reference to sight threatening complications such as disc neo-vascularization, vascular changes, or fovea exudation [8].

1.4 AIM

The key difference between the above mentioned projects and this project is that all these projects as mentioned utilized in depth techniques of computer vision that is the use of image processing algorithms to detect different components of retina in an image, AI techniques were negligibly used for pattern recognition in these projects, and these projects concentrated on detection and recognition of specific components in a retinal image but not the differentiation among them as a whole.

The aim of this project was to classify numerous retinal diseases by processing the retinal image, data analysis and development of an intelligent system for recognition. Thus this work encapsulated intelligences in itself which made it possible for the system to recognize the differences among retinal images categorized as having some specific disease/s.

The goal was to develop a design as well as an implementation of a neural network, which has the ability to ***accurately*** recognize the different categories of retinal diseases. The neural network design constraints were that the design must be directed for accurate classification, thus it must have a powerful recognition capability, design must be efficient in terms of time or in other words one must work to reducing the time complexity of the training and the testing phases and not much emphasis must be placed to use image processing techniques, since heavy reliance upon them would reduce the focus from the neural nets.

It is the basic feature of an intelligent system that it evolves and adapts to the changes with time. That is, it has the ability to improve its performance over the set of tasks for which it was originally built. This system is supposed to behave in a similar manner. It has the ability to recognize and classify all those diseases that have not already been defined by the trainer. Thus an image processing expert is restricted to classifying and recognizing a specific retinal disease at a given time. For every new disease, the expert has to extract new patterns/symptoms and thus such system keeps on changing for every new retinal disease.

The aim was to dynamically differentiate among newly discovered diseases and the old ones. Thus a disease was named, which then became an identifier for the newly discovered disease and the system, started to classify that as well. So as long as the training was available for the newly discovered data, the system kept on predicting the disease.

1.5 Scope

Until now, a physician or an eye specialist was the person who would manually examine the retinal image and make a decision, whether the subject had a disease or not and if there existed some pattern/symptom for some abnormality, then among many abnormal patterns to find the one similar to the current one. Such decisions were subject to errors of judgment.

This project aimed to help an eye specialist in the decision making process. It can be used to identify among many different retinal diseases on the basis of sufficient amount of data. Analysis of the image could produce results that might even vindicate his/her analysis. Its self evolving ability makes life easier for a person using it as it can prove much useful as long as new diseases are being discovered.

2 Literature Review

2.1 Anatomy of the Human Eye

The eye ball, as in Figure 2-1, is like a globe with a slight forward protrusion (cornea) at the anterior pole. The antero-posterior diameter varies from 22-27 mm. the horizontal diameter varies from 20-25 mm. The globe has three main layers; each of which is further sub-divided in to portions. The outer supporting coat is composed of the transparent cornea and the opaque sclera and their junction is called the corneoscleral sculus or limbus. The middle vascular coat is called the uveal tract. It consists of the choroid, the ciliary body and the iris. The iris contains a central opening called the pupil. The inner layer consists of the retina, which is composed of inner sensory retina and an outer layer of pigment epithelium.

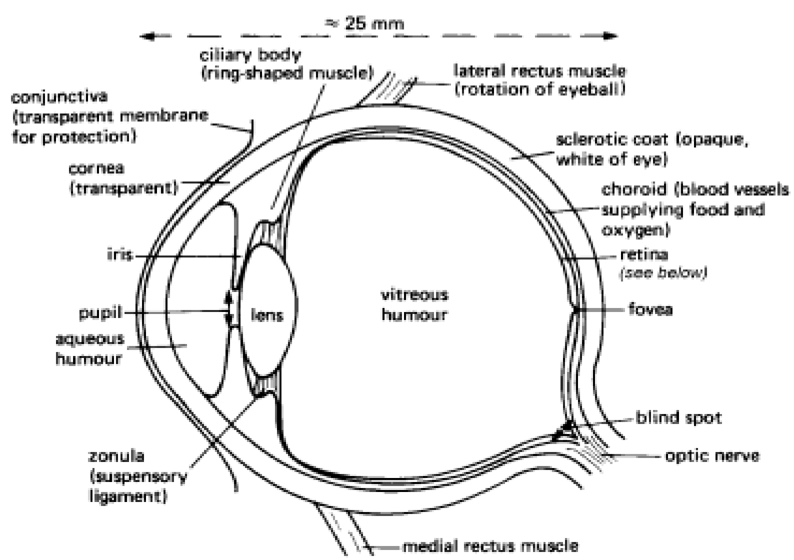

Figure 2-1: Human Eye [10, pp. 35]

The lens is a transparent structure located immediately behind the iris, and is retained in position by a circle of fine fibrils, the zonules, which are attached to the ciliary body and the capsule of the lens.

The eye encloses three chambers, the anterior chamber is bounded by the cornea anteriorly and iris and central part of the lens posteriorly. It communicates with the posterior chamber through the pupil; the posterior chamber is very small in size. It is bounded by iris in front; by the peripheral part of lens and zonules behind, the vitreous cavity is the largest of the three. It is located behind the lens and zonules; and is adjacent to retina through its extent.

Aqueous humor is secreted by the ciliary process into the posterior chamber and it passes through the pupil into the anterior chamber, trabecular meshwork opens into the canal of the Schelemm; a channel lined by endothelium that encircles the anterior chamber and drains the aqueous humor out of the eye ball [9].

2.1.1 Anatomy

The retina, as in Figure 2-2, is a delicate and thin membrane, extending from the optic disc behind to the ciliary body in front being situated between the choroid on the outer side and the hyaloid membrane of the vitreous on the inner side. The thickness of the retina is about 0.5mm near the optic disc, 0.2mm at the equator of the eyeball and 0.1mm anteriorly. Normally retina is transparent, but immediately after death, it becomes white. The anterior termination of the retina, where it is continuous with the epithelium of the ciliary body, is known as ora serrata.

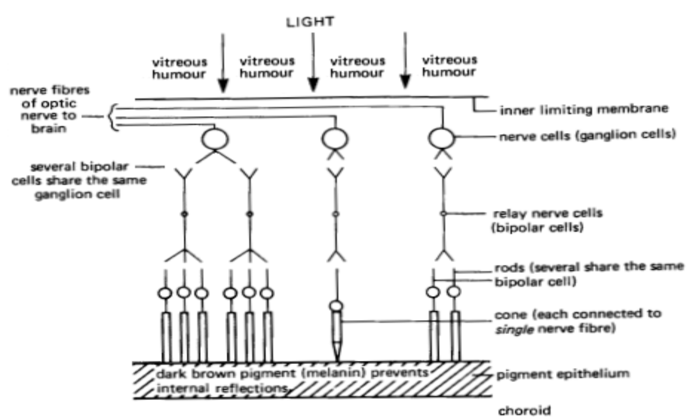

Figure 2-2: Structure of Retina [9]

On the inner surface of the retina, at the posterior pole of the eyeball, is the *macula lutea* or the yellow spot. It's an area about 5mm in diameter and at its center is a small depression, known as *fovea centralis* [9].

2.1.2 Function of Retina

The functions that retina has to perform are the central part, macula lutea, consists mainly of the cones and is responsible for the greatest visual activity in daylight and for color vision and peripheral part, which consists mainly of the rods, is responsible for the night vision. The ability to see in the darkness is due to the presence of a pigment, known as visual purple, in the rods.

2.2 Image Processing

2.2.1 Introduction

An **image** may be defined as a two-dimensional function, $f(x, y)$, where x and y are the *spatial (plane)* coordinates, and the amplitude of f at any pair of coordinates (x, y) is called the *intensity or the grey level* of the image at that point. When x, y and the amplitude values of f are all finite discrete quantities, the image is a **digital image**.

The field of **digital image processing** refers to processing the digital images by means of a digital computer. A digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are referred to as *picture elements, image elements*, *pels* and *pixels*. Pixel is the most widely used term to denote the elements of a digital image.

Unlike humans, who are limited to the visual band of electromagnetic (EM) spectrum, imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. Thus, digital image processing encompasses a wide and varied field of applications.

There is no general agreement regarding where image processing stops and other related areas, such as image analysis and computer vision, start. Sometimes a distinction is made by defining image processing as a discipline in which both the input and output of a process are images.

There are no clear-cut boundaries in the continuum from image processing at one end to computer vision at the other. However, one useful paradigm is to consider three types of computerized processes in the continuum; low-, mid-, and high-level processes. *Low-level processes* involve primitive operations such as image processing to reduce noise, contrast enhancement, and image sharpening. A low-level process is characterized by the fact that both its inputs and outputs are images. *Mid-level processes* on images involve tasks such as segmentation (partitioning an image into regions or objects), description of those objects to reduce them to make them suitable for computer processing and classification (recognition) of the individual objects. A mid-level process is characterized by the fact that its inputs generally are images, but its outputs are attributes extracted from those images (for instance edges, contours, and the identity of the individual objects). Finally, *higher-level processing* involves "making sense" of an ensemble of recognized objects, as in image analysis, and, at the far end of the continuum, performing the cognitive functions normally associated with vision.

Based on the preceding comments, a logical place of overlap between image processing and image analysis is the area of recognition of the individual regions or objects in an image. Thus digital image processing encompasses processes whose inputs and outputs are images and, in addition encompasses processes that extract attributes from images, up to and including the recognition of the individual objects [10, pp. 1-2] .

As a simple example to clarify these concepts, consider the area of automated analysis of text. The process of acquiring an image of the area containing the text, preprocessing that image, extracting (segmenting) the individual characters, describing the characters in the suitable form for computer processing, and recognizing those individual characters, all lie in the domain of digital image processing. Making sense of the content of the page may be viewed as being in the domain of image analysis and computer vision, depending on the level of complexity implied by the statement "making sense".

2.2.2 Applications of Image Processing

Today, there is almost no area of technical endeavor that is not influenced in anyway by digital image processing. The areas of application of digital image processing are so varied that some form of organization is desirable while attempting to capture the breadth of this field. One of the simplest ways is to categorize images according to their sources. The principal energy source for images in use today is the electromagnetic energy spectrum. Other important sources of energy include acoustic, ultrasonic and electronic. Synthetic images, used for modeling and visualization, are generated by computer [10, pp. 7].

2.2.2.1 Gamma Ray Imaging

Major uses of imaging based on gamma rays include nuclear medicine and astronomical observations. In nuclear medicine, the approach is to inject a patient with a radioactive isotope that emits gamma rays as it decays. Images, such as in Figure 2-3, are produced from the emissions collected by gamma ray detectors. The figure shows an image of a complete bone scan obtained by using gamma-ray imaging. Images of this sort are used to locate sites of bone pathology, such as infections or tumors [10, pp 11].

Figure 2-3: Gamma Ray Effect [10, pp. 8]

X-rays are among the oldest sources of EM radiation used for imaging. The best known use of x-rays is medical diagnostics, but they are also used extensively in industry and other areas, like astronomy. X-rays for medical and industrial imaging are generated using an X-ray tube. The intensity of the X-rays is modified by absorption as they pass through a patient, and the resulting energy falling on the film develops it, much in the same way that light develops the photographic film.

In digital radiography, digital images, such as in Figure 2-4, are obtained by one of the two methods: (1) by digitizing X-ray films; or (2) by having the X-rays that pass through the patient fall directly onto devices (such as phosphor screen) that convert X-rays to light. The light signal in turn is captured by a light-sensitive digitizing system.

Angiography is another major application in an area called contrast-enhancement radiography. This procedure is used to obtain images (called angiograms) of blood vessels. Techniques similar to the ones described above, but generally involving higher-energy X-rays, find their applications in the industrial processes [10, pp. 9-11].

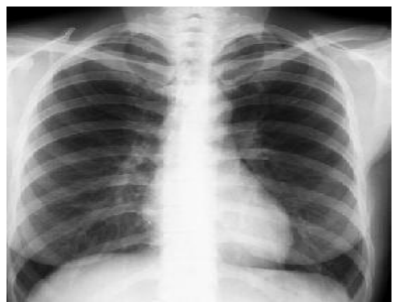

Figure 2-4: X-Ray Effect [10, pp. 10]

2.2.2.2 Imaging in Ultraviolet Band

Applications of the ultraviolet light are varied. They include lithography, industrial inspection, microscopy, lasers, biological imaging, and astronomical observations. Ultraviolet light is used in fluorescence microscopy, one of the fastest growing areas of microscopy. Fluorescent microscopy (as in the Figure 2-5) is an excellent method for studying the materials that can be made to fluoresce, either in their natural form (primary fluorescence) or when treated with chemicals capable of fluorescing (secondary fluorescence) [10, pp. 11].

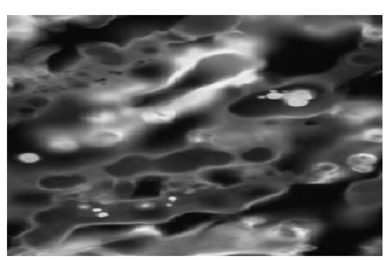

Figure 2-5: Fluorescent Microscopy [10, pp. 12]

2.2.2.3 *Imaging in Visible and Infrared Band*

Considering that the visual band of the electromagnetic spectrum is the most familiar in all our activities, it is not surprising that imaging in this band out-weighs by far all other methods in terms of scope of application. As an illustration of image processing in the visual spectrum, consider Figure 2-6 showing a thumb print.

Figure 2-6: Thumb Print [10, pp. 19]

Images of fingerprints are routinely processed by computer, either to enhance them or to find features that aid in the automated search of a database for potential matches. The vehicle images, as one in the Figure 2-7, are examples of automated license plate readings. The light rectangles indicate the area in which imaging system detected the plate. The black rectangles show the result of automated reading of the plate content by the system. License plate reading and other applications of character recognition are used extensively for traffic monitoring and surveillance [10, pp. 12-18].

Figure 2-7: Vehicle License Plate Image [10, pp. 19]

2.2.2.4 Imaging in Microwave Band

The dominant application of imaging in the microwave band is the radar. The unique feature of radar imaging is its ability to collect data over virtually any region at any time, regardless of weather or ambient lightning conditions. Some radar waves can penetrate the clouds, and under certain conditions can also see through vegetation, ice, and extremely dry sand. Imaging radar works like a flash camera in that it provides its own illumination (microwave pulses) to illuminate an area on the ground and take a snapshot image. Instead of a camera lens, radar uses an antenna and digital computer processing to record its images. In radar image, one can see only the microwave energy that was reflected back towards the radar antenna [10, pp. 19].

2.2.3 Fundamental Steps in Image Processing

The study of digital image processing can be partitioned into two broad categories that is the methods whose inputs and outputs are images and the methods whose inputs may be images, but whose outputs are attributes, extracted from those images.

This organization is summarized in the Figure 2-8. The diagram does not imply that every process is applied to an image. Rather, the intention is to convey an idea of all the methodologies that can be applied to the images for different purposes and possibly with different objectives [10, pp. 25-28].

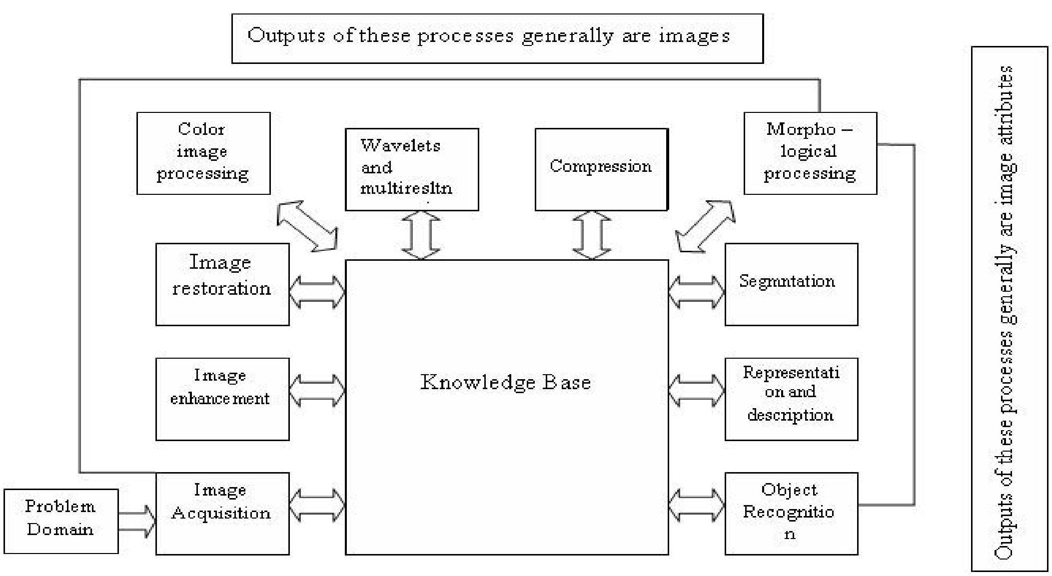

Figure 2-8: Fundamentals of Image Processing [10, pp. 26]

2.2.3.1 Image Acquisition

Image acquisition is the first process as in the Figure 2-8. Acquisition could be as simple as being given an image that is already in the digital form. Generally, the image acquisition process involves preprocessing, such as scaling. Some techniques used to acquire an image that is not in a digital form are: image acquisition using a single sensor, image acquisition using a sensor strips and image acquisition using sensor arrays.

2.2.3.2 Image Enhancement

Image enhancement is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image. A familiar example of enhancement is when increase in the contrast of an image because “it looks better.”

2.2.3.3 Image Restoration

Image restoration is an area that also deals with improving the appearance of an image. However, unlike the enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on the mathematical or probabilistic models of image degradation. Enhancement, on the other hand, is based on the subjective preferences regarding what constitutes a “good” enhancement result.

2.2.3.4 Color Image Processing and Wavelets

Color image processing is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet. Wavelets are the foundation for representing images in various degrees of resolution.

2.2.3.5 Compression and Morphological Processing

Compression, as the name implies, deals with the techniques for reducing the amount of storage required to save an image, or the bandwidth required to transmit it. Although storage technology has improved significantly over the past decade, the same cannot be said for transmission capacity. This is true particularly in uses of the Internet, which are characterized by significant pictorial content. Image compression is familiar to most users of computers in the form of image file extensions, such as jpg file extension used in the JPEG (Joint

Photographic Expert Group) image compression standard. “Morphological Processing” deals with tools for extracting image components that are useful in the representation and description of the shape.

2.2.3.6 Segmentation

“Segmentation” procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in the digital image processing. A rugged segmentation procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. On the other hand, weak or erratic segmentation algorithms almost guarantee eventual failure. In general, the more accurate the segmentation, the more likely recognition is to succeed.

2.2.3.7 Representation and Recognition

“Representation” and description almost always follow the output of the segmentation stage, which usually is raw pixel data, constituting either the boundary of region or all the points in the region itself. In either case, converting the data to a form suitable for computer processing is necessary. Boundary representation is appropriate when the focus is on the external shape characteristics, such as corners and inflections. Regional representation is appropriate when the focus is on the internal properties, such as texture or skeletal shape. In some applications these techniques complement each other. Description, also called feature selection, deals with extracting the attributes that result in some quantitative information of interest or are basic for differentiating one class of objects from another. “Recognition” is the process that assigns a label (e.g., “vehicle”) to an object based on its descriptors.

2.2.3.8 Knowledge Base

Knowledge about a problem domain is coded into an image processing system in the form of the knowledge database. This knowledge may be as simple as detailing regions of an image where the information of interest is known to be located, thus limiting the search that has to be conducted in seeking that information. The knowledge base also can be quite complex, such as an inter-related list of all major possible defects in a materials inspection problem or an image database containing high resolution satellite images of a region in connection with change-detection applications. In addition to guiding the operation of each processing module, the knowledge base also controls the interaction between modules.

2.3 Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by the biological nervous systems. As in nature, the network function is determined largely by the connections between these basic elements. Neural networks, like people, learn through examples. A Neural Network can be trained to perform a particular function by adjusting the values of the connections (weights) between the elements.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is can be seen in Figure 2-9. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are used, in supervised learning, to train a network.

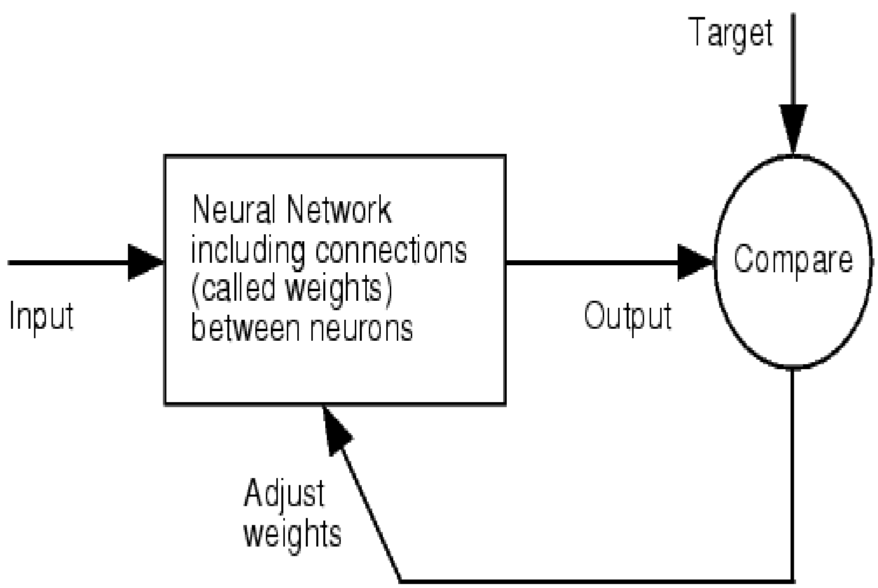

Figure 2-9: A General Neural Network Diagram [11]

Batch training of a network proceeds by making weight and bias changes based on an entire set (batch) of input vectors. Incremental training, changes the weights and biases of a network as needed after presentation of each individual input vector. Incremental training is sometimes referred to as "on line" or "adaptive" training.

Neural networks have been trained to perform complex functions in various areas of applications including pattern recognition, identification, classification, speech, vision and control systems.

The supervised training methods are commonly used, but other networks can be obtained from unsupervised training techniques or from direct design methods. Unsupervised networks can be used, for instance, to identify groups of data. Certain kinds of linear networks and Hopfield networks are designed directly. In summary, there are a variety of kinds of design and learning techniques that enrich the choices that a user can make [11].

2.3.1 Neuron Model

2.3.1.1 Simple Neuron

A neuron with a single scalar input and no bias appears on the left in Figure 2-10.

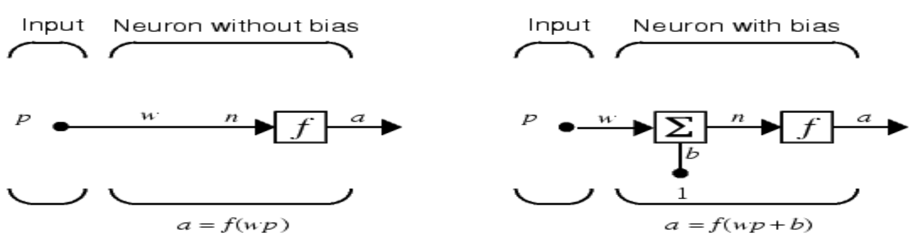

Figure 2-10: A Simple Neuron [11]

The scalar input *p* is transmitted through a connection that multiplies its strength by the scalar weight *w*, to form the product *wp*, again a scalar. The weighted input, *wp*, is the only argument of the transfer function *f*, which produces the scalar output a. The neuron on the right has a scalar bias, *b*. The bias may be viewed as simply being added to the product *wp* as depicted by the summing junction or as shifting the function *f* to the left by an amount *b*. The bias is much like a weight, except that it has a constant input of 1.

The transfer function *net* input *n*, again a scalar, is the sum of the weighted input *wp* and the bias *b*. This sum is the argument of the transfer function *f*. *f* is a transfer function, typically a step function or a sigmoid function, which takes the argument *n* and produces the output *a*. One can train the network to do a particular job by adjusting the weight or bias parameters, or the network itself may adjust these parameters to achieve some desired end.

2.3.1.2 Transfer Functions

Many transfer functions are in vogue. Three of the most commonly used functions are as in Figure 2-11, Figure 2-12 and Figure 2-13.

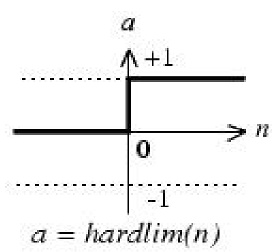

Figure 2-11: Hard Limit Transfer Function [11]

The hard-limit transfer function in Figure 2-11 limits the output of the neuron to either 0, if the net input argument *n* is less than 0; or 1, if *n* is greater than or equal to 0. The linear transfer function is in Figure 2-12.

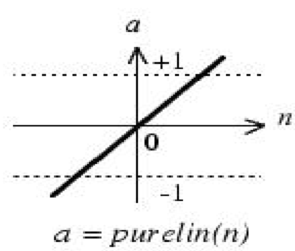

Figure 2-12: Pure Linear Transfer Function [11]

Neurons of this type are used as linear approximators within the Linear Filters. The sigmoid transfer function as in Figure 2-13 takes the input, which may have any value between

$+\infty$ and $-\infty$, and squashes the output into the range 0 to 1, the reason it is known as the "squashing function."

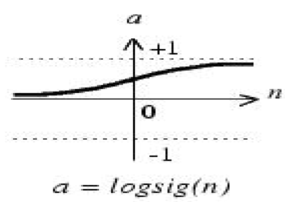

Figure 2-13: Sigmoid Transfer Function [11]

This transfer function is commonly used in the back-propagation networks, in part because it is differentiable.

2.3.2 Neuron with Vector Input

A neuron with a single R-element input vector is as in Figure 2-14. The individual element inputs

$$p_1, p_2, \dots p_R \qquad \text{Equation 2-1}$$

and multiplied by weights

$$w_{1,1}, w_{1,2}, \dots w_{1,R} \qquad \text{Equation 2-2}$$

and the weighted values are fed to the summing junction. Their sum is simply *Wp,* the dot product of the (single row) matrix *W* and the vector *p*.

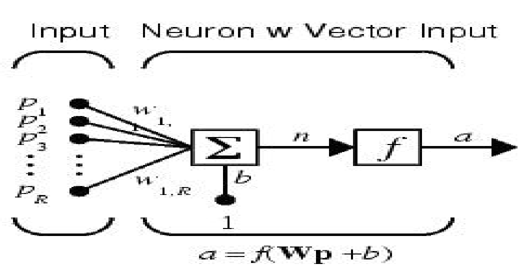

Figure 2-14: Neural Network with Biases [11]

where *R* is the number of elements in the input vector. The neuron has a bias *b*, which is summed with the weighted inputs to form the net input *n*. This sum, *n,* is the argument of the transfer function *f*.

$$n = w_{1,1}p_1 + w_{1,2}p_2 + \ldots + w_{1,R}p_R + b$$ Equation 2-3

2.3.3 Network Architectures

Two or more of the neurons described earlier can be combined in a layer, and a particular network could contain one or more such layers. First consider a single layer of neurons.

2.3.3.1 Single Layer Neurons

A one-layer network with R input elements and S neurons are in Figure 2-15.

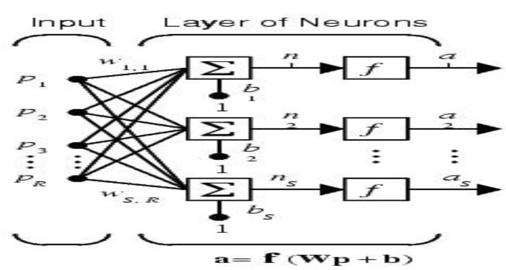

Figure 2-15: Single Layer Neurons [12]

In this network, each element of the input vector p is connected to each neuron input through the weight matrix W. The ith neuron has a summer that gathers its weighted inputs and bias to form its own scalar output n(i). The various n(i) taken together form an S-element net input vector n. Finally, the neuron layer outputs form a column vector “a”.

Note that it is common for the number of inputs to a layer to be different from the number of neurons (i.e., R $\neq$ S). A layer is not constrained to have the number of its inputs equal to the number of its neurons.

One can create a single (composite) layer of neurons having different transfer functions simply by putting two of the networks described earlier in parallel. Both networks would have the same inputs, and each network would create some of the outputs. The input vector elements enter the network through the weight matrix W as in Figure 2-16.

$$\mathbf{W} = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,R} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,R} \\ & & & \\ w_{S,1} & w_{S,2} & \cdots & w_{S,R} \end{bmatrix}$$

Figure 2-16: Weight Matrix [12]

Note that the row indices on the elements of matrix W indicate the destination neuron of the weight, and the column indices indicate which source is the input for that weight. Thus, the indices in $\boldsymbol{W_{1,2}}$ say that the strength of the signal from the second input element to the first (and only) neuron is $\boldsymbol{W_{1,2}}$. [12]

2.3.3.2 Multiple Layers of Neurons

A network can have several layers. Each layer has a weight matrix W, a bias vector b, and an output vector a. To distinguish between the weight matrices, output vectors, etc., for each of these layers in figures, the number of the layer is append as a superscript to the variable of interest. It can see the use of this layer notation in the three-layer network in **Error! Reference source not found.**.

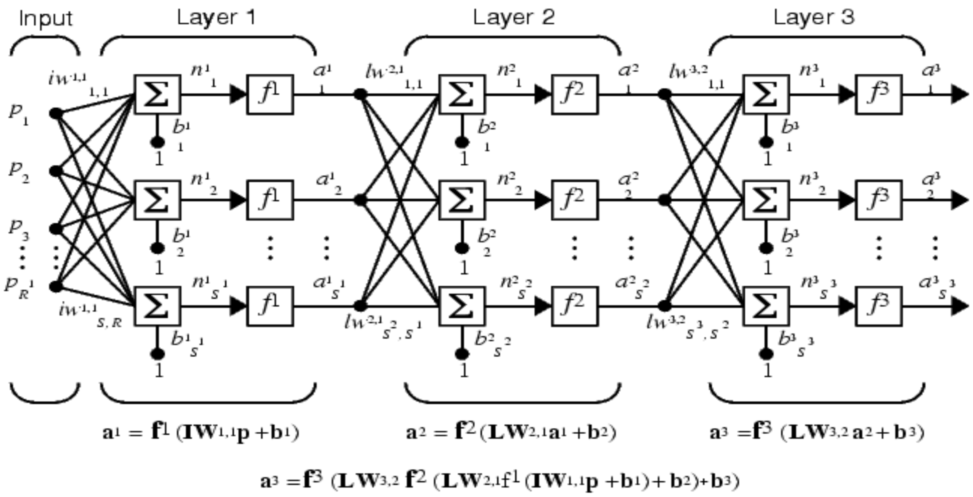

Figure 2-17: Multiple Layer Neural Network [12]

The layers of a multilayer network play different roles. A layer that produces the network output is called an *output layer*. All other layers are called *hidden layers*. The three-layer network described earlier has one output layer (layer 3) and two hidden layers (layer 1 and layer 2). Some authors refer to the inputs as a fourth layer.

Multiple-layer networks are quite powerful. For instance, a network of two layers, where the first layer is sigmoid and the second layer is linear, can be trained to approximate any function (with a finite number of discontinuities) arbitrarily well. This kind of two-layer network is used extensively in back-propagation.

2.3.4 Feed Forward Networks

Feed-forward neural networks allow signals to travel one way only; from input to output. There is no feedback (loops) i.e. the output of any layer does not affect that same layer. Feed-forward neural networks tend to be straight forward networks that associate inputs with outputs. They are extensively used in pattern recognition. This type of organization is also referred to as bottom-up or top-down.

2.3.5 Feed Back Networks

Feedback networks can have signals traveling in both directions by introducing loops in the network. Feedback networks are very powerful and can get extremely complicated. Feedback networks are dynamic; their 'state' is changing continuously until they reach an equilibrium point. They remain at the equilibrium point until the input changes and a new equilibrium needs to be found. Feedback architectures are also referred to as interactive or recurrent, although the latter term is often used to denote feedback connections in single-layer organizations.

2.3.6 Training Styles

Depending on the ways, the weights and biases of a network are updated; one can classify the training styles into two broad categories, Incremental Training: the weights and biases of the network are updated each time an input is presented to the network. It can be applied to both static and dynamic networks, although it is more commonly used with dynamic networks, such as adaptive filters and Batch training: the weights and biases are only updated after all of the inputs are presented. Its application is common for both static and dynamic networks.

2.3.7 Learning Process

One can distinguish two major categories of neural networks those are Fixed Networks: in which the weights cannot be changed. In such networks, the weights are fixed a priori according to the problem to be solved and Adaptive networks: they are able to change their weights.

All learning methods used for adaptive neural networks can be classified into two major categories:, Supervised Learning: this incorporates an external teacher, so that each output unit is told what its desired response to input signals, ought to be. During the learning process global information may be required. An important issue concerning supervised learning is the problem of error convergence, i.e., the minimization of error between the desired and computed unit values. The aim is to determine a set of weights which minimizes the error. One well-known method, which is common to many learning paradigms, is the least mean square (LMS) convergence, which takes its origin from stochastics and Unsupervised Learning: this uses no external teacher and is based upon only local information. It is also referred to as self-organization, in the sense that it self-organizes data presented to the network and detects their emergent collective properties. Thus the neural network learns off-line if the learning phase and the operation phase are distinct. A neural network learns on-line if it learns and operates at the same time. Usually, supervised learning is performed off-line, whereas the unsupervised learning is performed on-line.

2.3.8 Back Propagation

Back propagation was created by generalizing the Widrow-Hoff learning rule to multiple-layer networks and nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined. Networks with biases, a sigmoid layer, and a linear output layer are capable of approximating any function with a finite number of discontinuities.

Standard back propagation is a gradient descent algorithm in which the network weights are moved along the negative of the gradient of the performance function. The term back propagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

Properly trained back propagation networks tend to give reasonable answers when presented with inputs that they have never seen. Typically, a new input leads to an output similar to the correct output for input vectors used in training that are similar to the new input being presented. This generalization property makes it possible to train a network on a representative set of input/target pairs and get good results without training the network on all possible input/output pairs [13].

2.3.9 Back Propagation Algorithm

There are many variations of the back propagation algorithm. The simplest implementation of back propagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly - the negative of the gradient. A single iteration of this algorithm can be written as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k$$ Equation 2-4

where $\mathbf{x}_k$ is a vector of current weights and biases, $\mathbf{g}_k$ is the current gradient, and α_k is the learning rate.

There are two different ways in which this gradient descent algorithm can be implemented: the incremental mode and the batch mode. In the incremental mode, the gradient is computed and the weights are updated after each input is applied to the network. In the batch mode all of the inputs are applied to the network before the weights are updated.

The implementation of back propagation algorithm for practical problems is suited for converging the results much faster than normal. These faster variations fall into two main categories, the first category uses heuristic techniques, which were developed from an analysis of the performance of the standard steepest descent algorithm and the second category of fast algorithms uses standard numerical optimization techniques.

2.3.10 Levenberg Marquardt

The Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares (as is typical in training feed-forward networks), then the Hessian matrix can be approximated as:

$$\mathbf{H} = \mathbf{J}^T \mathbf{J}$$ Equation 2-5

And the gradient can be computed as:

$$\mathbf{g} = \mathbf{J}^T \mathbf{e}$$ Equation 2-6

Where **J** is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and **e** is a vector of network errors. The Jacobian matrix can be computed through a standard back-propagation technique that is much less complex than computing the Hessian matrix. The Levenberg-Marquardt algorithm uses this approximation to the Hessian matrix in the Newton-like update:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T\mathbf{J} + \mu\mathbf{I}]^{-1}\mathbf{J}^T\mathbf{e}$$ Equation 2-7

When the scalar μ is zero, this is just Newton's method, using the approximate Hessian matrix. When μ is large, this becomes gradient descent with a small step size. Newton's method is faster and more accurate near an error minimum, so the aim is to shift towards Newton's method as quickly as possible. Thus, μ is decreased after each successful step (reduction in performance function) and is increased only when a tentative step would increase the performance function. In this way, the performance function will always be reduced at each iteration of the algorithm [14].

Chapter 3

3 Approach and Design

An approach has to be established to design a system that is intelligent enough to recognize the pattern of a disease from a retinal image. Therefore the project is mainly divided in to two phases: developing a suitable approach and developing a design according to the approach developed in the first phase.

3.1 Developing an Approach

For development of an approach, certain consideration of the end product was necessary. These constraints were the system is inducted with a huge amount of data in form of a color retinal image; the system must be efficient in terms of time and memory usage, reduction of complexity during training and testing of the system and to achieve a higher degree of classification.

Starting from scratch, two approaches were developed and the one selected was the most suitable according to constrains mentioned. The approach dealt with amount of data to be induced into the neural network. Has a deep impact on the computational complexity and affects the usage of memory and time during the training and the test phases.

Also the approach leads to the neural network design. It serves as a guide to the construction of the neural network which is the most essential portion of the whole architecture.

3.2 Developing Neural Network Design

It is the design of the neural network which in actual poses the element of complexity into the system. The basic unit of the neural network is a neuron. The structure of neural network consists of neurons connected to other neurons in parallel. Thus the amount of parallelism in the structure determines the nature and complexity of the system.

The structure effects the nature of the system i.e. increase in parallelism of neural network increases the ability of the system to learn. And thus increases the degree of classification. But at some point increase in parallelism doesn't affect the degree of classification up to a greater extends, rather it only affect the complexity.

The structure directly affects the complexity of the system. Increase in parallelism certainly increases the complexity and system usage of memory and time becomes enormous, making it inefficient. There must be a trade off made between the complex nature of the system and the degree of classification.

Design of the neural network is therefore dependent upon the approach developed in the first phase. If the approach is developed keeping in view the constraints, the design of the neural network may achieve higher degree of classification and less complexity with out inducing much trade of between the two. Hence the system developed is time and memory efficient as well.

3.3 First Approach – Pixel by Pixel

3.3.1 Character Recognition

Images for the characters 0 – 9 were acquired. The background of all the images had the same gray scale values therefore the images processing phases like the background elimination and the feature extraction were not required. All the images had the same image formats, i.e., bitmaps (BMP). The dimension of each image used for this experiment was 4x6.

The images were converted into matrix form, containing the values of the pixels. Each pixel represented the gray scale value of the respective images. The image matrix was equal to the dimensions of the image. Multiple instances for each character were acquired. These Images were properly aligned to the center and were having a uniform background color. Thus the data was assembled uniformly for the input of neural network.

3.3.2 Neural Network Design

The network used for character recognition was based on the simple back-propagation feed forward neural network, which back-propagation algorithm uses the gradient descent algorithm for training this network. The network had three layers.

Input Layer: The number of inputs to this network was equal to the dimension of the matrix containing the grey scale value of the image for a given character.

Hidden Layer: The hidden layer had 5 neurons. The number of neurons for the hidden layer was only taken as a design issue and had little effect, whatsoever, on the degree of classification. Increasing the number of neurons increases the parallelism of the network, which increases the degree of classification. But because of the rise in complexity of the network requires greater time for training. Therefore there is a trade off between the training time and the accuracy of the classification. The transfer function used for the hidden layer was log sigmoid. This function squashed the inputs to a range of 0 to 1.

Output layer: Since the total number of characters to be recognized was ten, which could be distinguished by a four digit binary number. Each binary number referred to one of the characters. Thus the output produced one of the binary numbers corresponding to the particular character between 0 and 10. Therefore, the output layer for this network contained four neurons, using the transfer function same as that for the hidden layer, i.e., log sigmoid.

3.3.3 Learning Rate

The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and may never reach a global minimum, thus becoming unstable. If the learning rate is too small, the algorithm might take just too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface. The performance of the algorithm can be improved if the learning rate is allowed to change over the training process. An adaptive learning rate attempts to keep the learning step size as large as possible while keeping learning stable. The training function used in this network was traingdx, which implemented the adaptive learning rate technique. It combined the adaptive learning rate with the momentum training.

3.3.4 Performance Goal

The performance goal is the accuracy, which is desirable, with which the neural network can classify the unseen instances. It gives an error margin to the classification and thus it must be chosen such that the threshold between different categories of classifications is maintained. The smaller the value, the greater is the accuracy and vice versa. But reducing the value increases the number of epochs – where one epoch is the entire pass of all the training examples - therefore, increasing the time taken to converge to the goal. It also increases the likelihood of never reaching the goal within allowed number of epochs. The performance goal chosen in this case was $\boldsymbol{1e^{-3}}$, which quiet reasonable is given that the neural net was observed to perform quiet well with this margin of error.

3.3.5 Result

The training was successfully commenced and the desired performance goal was achieved. A number of test cases i.e. images of the characters were introduced to the character recognition system. The number of training examples for each character is descript in form of **Error! Reference source not found.** which as can be seen is different for different characters depending upon the number of variations a character image has, and is necessary to be taken under consideration for a higher degree of classification.

Table 3-1: Training Examples used for each Character

Character Image	Number of Training Examples
Character 0	15
Character 1	25
Character 2	15
Character 3	15
Character 4	15
Character 5	20
Character 6	20
Character 7	25
Character 8	15
Character 9	15

After submission of hectic amount of data for training, 90% classification was achieved. The summary of the results can be seen as in **Error! Reference source not found.**. In access of 50 character images, 45 images were classified correctly.

Table 3-2: Recognition Results for Character Recognition

Test Cases	Number of Correct	Percentage of Correct
Character 0	5 out of 5	100%
Character 1	3 out of 5	60%
Character 2	5 out of 5	100%
Character 3	5 out of 5	100%
Character 4	5 out of 5	100%
Character 5	5 out of 5	100%
Character 6	4 out of 5	80%
Character 7	3 out of 5	60%
Character 8	5 out 5	100%
Character 9	5 out 5	100%
Overall	45 out of 50	90%

3.4 Second Approach

3.4.1 Facial Recognition

Images for the human faces were acquired. The background of all the images was uniform and had the same gray scale values. Thus image processing phases like the background elimination and the feature extraction were surpassed. All the images had the same image formats, i.e., portable gray map (PGM). The dimension of each image used for this experiment was 122x92. There were a total of 60 images, 10 each for 6 persons. Nine of these ten images were of each person, and were used for training the neural network. The 10th image was given as a test case to the ANN. The images taken were the frontal, the left and the right poses with different face expressions.

The images were converted into a matrix form containing the values of the pixels. Each pixel represents the gray scale value of the respective images. The image matrix was equal to the dimensions of the image. Multiple instances for each character were acquired.

3.4.2 Image Histogram

For each of the images, a histogram matrix was constructed as depicted in Figure 3-1, which contained the gray level frequency, i.e., occurrence of each gray level (0-255) in the entire image. This histogram matrix became the input to the neural net. Thus the histogram uniquely identified each image, since every distinct image has a different histogram. There is a possibility that two different images might have the same histogram but the probability for this is quite small. Even if the error does occur in the huge training sessions, ANN learning methods are quite robust to noise in the training data.

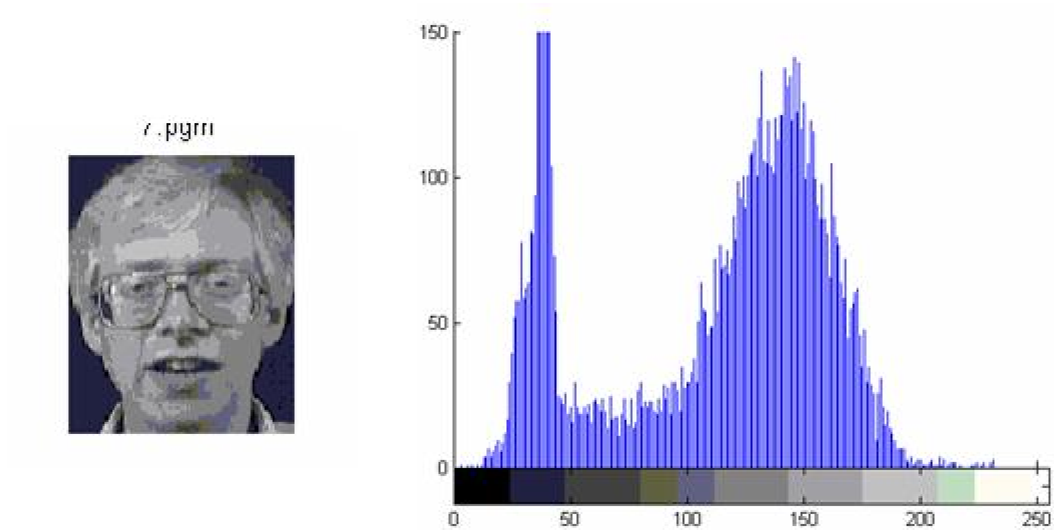

Figure 3-1: Image Histogram of a Gray Scale Image

3.4.3 Neural Network Design

The network used for face recognition was based on the back-propagation feed forward algorithm for the neural networks. Back-propagation uses the *Levenberg-Marquardt* algorithm for training this network. This is the fastest training algorithm for networks of moderate size. It has memory reduction feature for use when the training set is large. The network had four layers.

Input layer: The number of inputs to this network was equal to the dimension of the histogram matrix (256 inputs), containing the grey scale values of the images.

2 - Hidden layers: Each hidden layer had 5 neurons. The number of neurons for the hidden layer is only a design issue and has little effect on the degree of classification. Increasing the number of neurons increases the parallelism of the network, which in turn increases the degree of classification. Due to the in the complexity of the network, it requires greater time for training. Hence there is a trade off between the training time and the accuracy of classification. The hidden layers used the log sigmoid transfer function.

Output layer: The total number of persons to be recognized was six, which could be distinguished by six output neurons. The network's output encoded the person identified. The output layer used the pure linear transfer function. The output can be translated in form of a graph as in Figure 3-2. The x-axis represents the number of passes over data (epochs) whereas the y-axis describes the value of performance goal acquired at a certain epoch. The Figure 3-2 shows the decline of the graph towards the performance goal line and which can be seen as achieved in Figure 3-3.

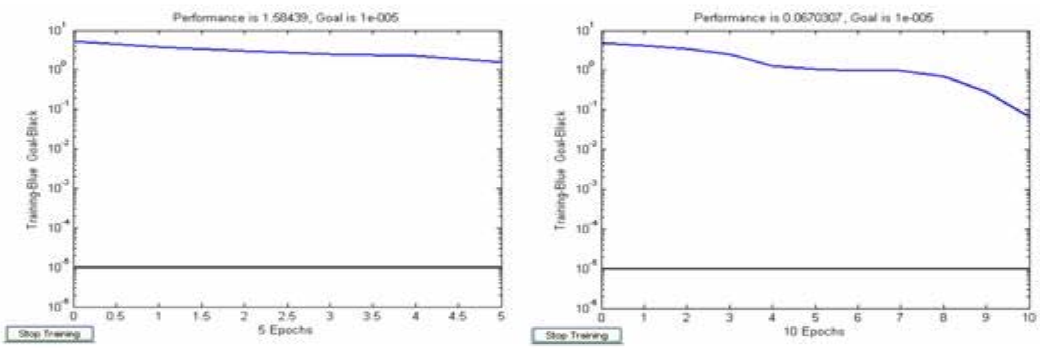

Figure 3-2: During Training is being Conducted

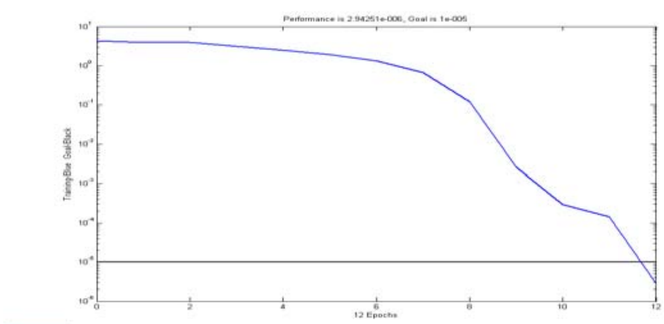

Figure 3-3: After Training is Complete

At each epoch, the sum of the squared error between the output and target is calculated in the Figure 3-2 and Figure 3-3. The sum is then propagated backward into the network such that neurons must reduce this error and therefore affecting the weights values of the neurons. This phenomenon is called the learning process of the neurons. The weight values are updated in such a way that the neurons contributing most in the error must reduce the error content. Thus during the training of the neural network, the error must be reduced until it reaches the goal that is the acceptable value of the sum of the squared error.

3.4.4 Learning Rate

The learning rate chosen for the training of the network was $\boldsymbol{5e^{-2}}$. The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge. It is not practical to determine the optimal setting for the learning rate before training. Therefore it is chosen on the experience of the previously used ANN such as ALVINN [15, pp. 82-85; 16] etc.

3.4.5 Performance Goal

The performance goal that is chosen in this case is $1\mathbf{e}^{-5}$. The smaller the value, the greater is the accuracy and vice versa. It is a measure that determines the accuracy of neural network convergence towards goal.

3.4.6 Network Design Constraints

The histogram matrix contained the value of occurrence of each gray level (0 - 255). Thus each value of the gray levels in the histogram matrix had an impact on the learning of the neural network. Each gray level could have a value between 0 (no occurrence of the gray level) to the number of image pixels (complete occurrence of the gray level).

The constraint on the log sigmoid transfer function, used in the network, is that it thrashes the values in a range from 0 to 1. Therefore any value greater than 10 is thrashed to exact 1 and any value less than -10 is thrashed to zero. It could lead to information loss, e.g., gray scale values of both 300 and 500 were thrashed to 1.

Therefore the histogram matrix was normalized to the values in the range of -10 to 10, so that the effect of the value of each gray level on the training of the network remained the same even if it was subjected to the neurons having the transfer function log sigmoid**.** A single normalization value was chosen for all the images so that they all remained consistent with each other (same proportionality among them).

3.4.7 Results

The performance goal was achieved by the neural network in less than 100 epochs. The total number of training examples was sixty. Ten images for each person were used during training while five other unseen instances were used for testing. The degree of classification achieved was 89.5% as can be seen from the **Error! Reference source not found.**, which is as good as in case of character recognition but with more complex data and in a very less time.

Table 3-3: Recognition Results for Facial Recognition

Test Cases	Number of Correct	Percentage of Correct
Person 1	5 out of 7	71.4%
Person 2	12 out of 12	100%
Person 3	8 out of 8	100%
Person 4	6 out of 9	66%
Person 5	7 out of 7	100%
Person 6	5 out of 5	100%
Overall	43 out of 48	89.5%

Thus this approach produced results that were inspiring enough for its adoption in project RETINA.

3.4.8 Advantages of Histogram Approach over Pixel by Pixel

The number of the inputs to the neural network is always fixed (i.e. in case of facial image recognition it is 256 input values) independent of the size of the image as it was in case of pixel by pixel approach where the size of the input was the dimensions of an image. Because the input values acquired in the histogram approach are independent of image dimensions, this number is significantly low and therefore reduces the complexity of the system, thus affecting the training time which reduces significantly.

In case of pixel by pixel approach, same image having a different location coordinate was treated as a different image and thus more training examples were required for the same image. Histogram approach makes the classification location independent.

The neural network architecture developed for pixel by pixel approach is dependent on the dimensions of images. The architecture has to be altered if the images used as training examples are used with different dimensions, thus making the system hardware (the one that captures images) dependent.

The system using the histogram approach is hardware independent because of the fact that input (in form of histogram of image) is always fixed and independent of the image dimensions. Certain amount of pre- processing is required while using pixel by pixel approach for achieving high degree of classification and reduction in the training errors e.g. image alignment, aspect ratio adjustment. Histogram approach does not require such pre processing phases. It can certainly work upon geometrically corrected images (retinal).

3.4.9 Potential Threat and its Solution

There is a very low probability of having same histograms for two different images especially in case of colored images. In case of retinal images, different images mean images having two different diseases but possessing same histogram. If such a scenario does occur, then this is a training error and might lure the training in a false direction. But since the probability of such a scenario ever occurring is very low, it needs not be worried about.

The neural network provides the solution to the problem. Neural network is robust to training errors. This means that the training of the network is not based upon only one image which might be faulty. The neural network trains upon a number of images and thus converges to solution. Therefore the training and classification is not affected even if there are errors in training examples as long as number of examples is sufficiently large.

3.5 Design

The architectural design describes the pattern according to which project was developed. This includes conceptual and implementation model. These models were designed once all the approaches to be used in the development of the complete system were taken under consideration. This helped in the construction of a layered architecture of the end product.

3.5.1 Conceptual Model

This model of the project provides an overview of the conceptually achieved functional specifications of the product. The Top Level Conceptual Model of the Architecture is depicted in Figure 3-4.

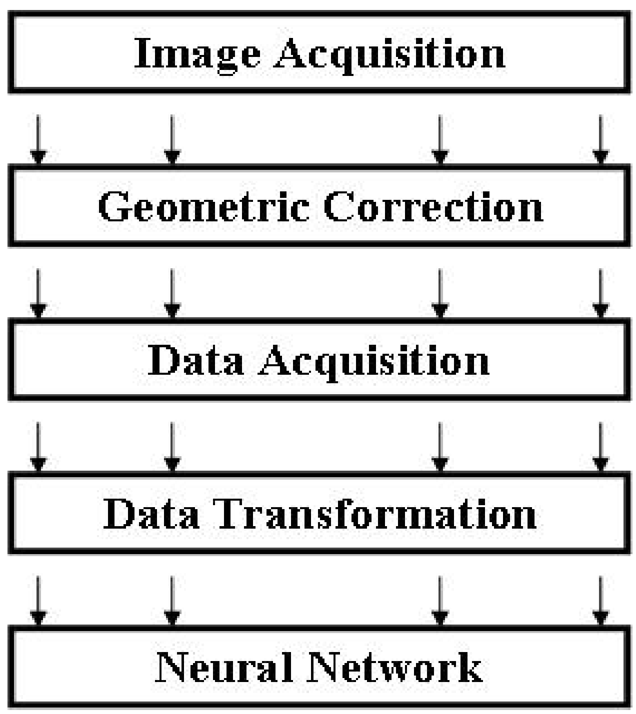

Figure 3-4: Conceptual Model

The functionality is divided into five layers. All Intermediate layers are dependent upon the outcome of the previous layer. Image acquisition is the foremost part of the project. Once images are acquired they need to be geometrically corrected for further manipulation (In our case the retinal images were already in digital form and geometrically corrected). Data in these images is acquired and then transformed into compatible format with the neural network. This format was in form of image signatures, serving a dual purpose of compatibility and creativity of patterns. Thus neural network is trained for pattern extraction and disease recognition.

3.5.2 Implemented Design

The implemented design of the project is as in Figure 3-5. MFC acts as an intermediate layer providing the user the functionality of underlying layers with the blend of Graphical User Interface, inheriting the functionality from Windows API.

Testing and training are handled separately in this architecture, where as the image acquisition in both cases use the functionality of image acquisition layer. This architecture provides the complete functionality as has been mentioned in the conceptual design. The difference lies in the grouping of sub tasks such as data acquisition, image histogram etc that comes under the task of testing for a better and efficient of resources.

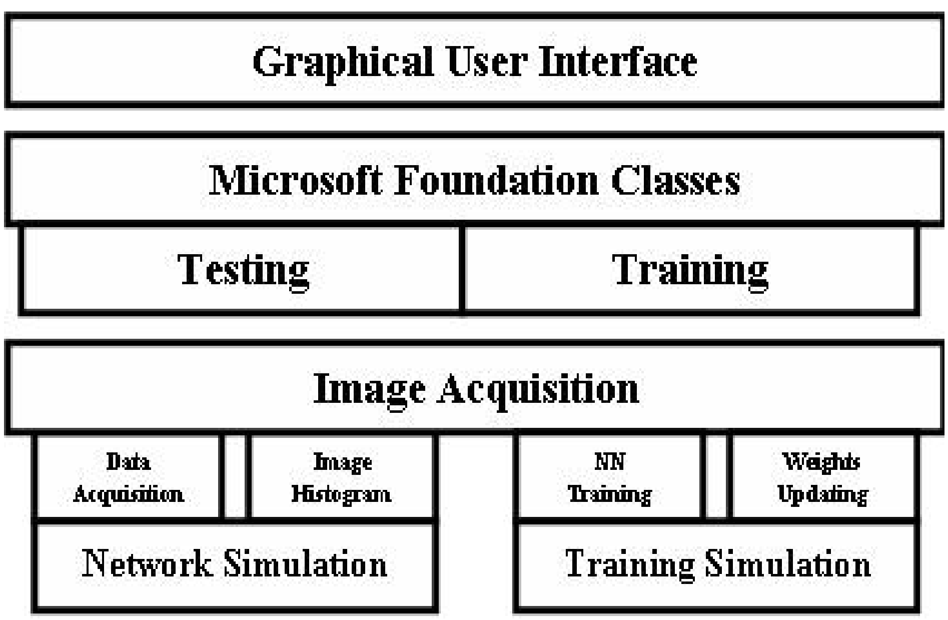

Figure 3-5: Implemented Design for Project RETINA

3.5.3 System Work Flow

The working of the system developed can be seen in Figure 3-6. Images are acquired, corrected and transformed for training and testing. The output of the system is the classification of the image having some disease.

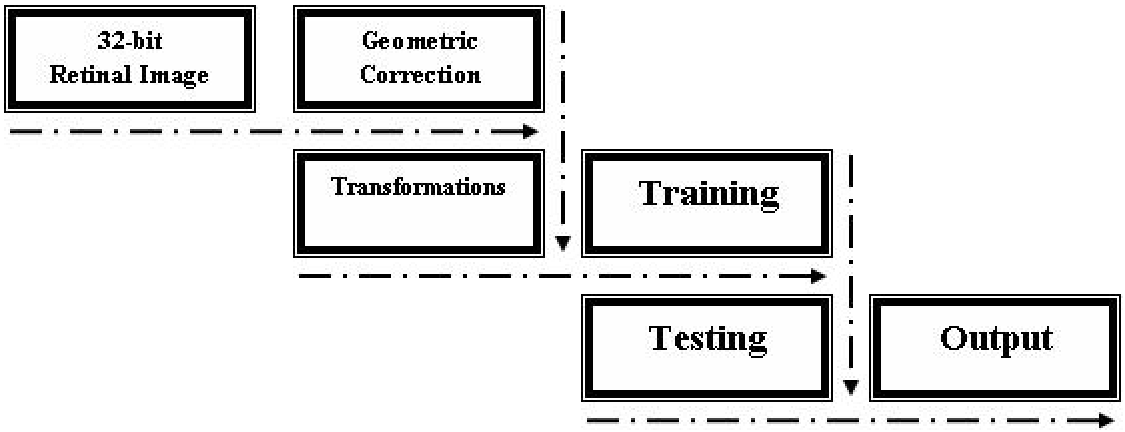

Figure 3-6: System Work Flow Diagram

One must keep a note the same work flow is used for both testing and training. The difference between the working of the two is, while training the output acts as measure to reduce the error between the correct and incorrect classification, during training of the system image data (in form of histogram signatures) is fed continuously until the performance measure or goal to be reached is met, during testing the output reveals the classification by the system correct or incorrect and only the test image is fed into the system and has nothing to do with the updating of state of the system.

3.5.4 Implemented Platform

Project RETINA is implemented using Visual C++ and its MFC support. All the underlying code providing the functionality of neural network and image manipulation is written in C++. The Graphical User Interface is composed of two tabs grouping the functionality of testing and training respectively.

A special image processing library MST [17], for reading, writing and manipulation of images is used. This library has also been used for data transformation in the intermediate stages of the project life cycle.

The neural network structure, function, algorithm has completely designed with any use of neural network library. The reason for not using any existing neural network library available is the structure of the neural network is specific to the design specifications and thus need not to be changed and the algorithm thus used to pattern extraction and recognition is not implemented by most of the neural network libraries available open source.

4 Implementation

4.1 Introduction

The implementation of the system that had the ability to recognize, learn, extract a pattern took place once a proper approach had been decided upon. This included data being properly assembled, design of the neural network and the algorithm to be used. Therefore, the approach provided the infrastructure necessary for creating an intelligent machine, with the capability to make decisions on its own.

Normal as well as retinal images containing diseases were obtained. All these images were 32 bit true color in nature. The image format used was (JPEG). The dimension of these images was (768x576). A total of 45 images were used as training examples for the neural network, fifteen images of the same pattern and therefore three patterns were to be classified by the system developed. The test cases were randomly selected among the images of the individual diseases. These test images were unseen for the network that is test cases were exclusive of those images used during the training of the neural network.

As mentioned earlier, images were 32 bit true color in nature. Out of the 32 bits, 24 bits contained the color information and remaining eight had other characteristics such as brightness, alpha and gamma. The color information in 24 bits had three components, red, green and blue having 8 bits each and therefore yielding 16 million colors.

The images were first converted into matrices, having dimensions equal to three times the size of the image (768x576x3). In another way of translation, the image was divided into three matrices having pixel values for the three component colors that is red green and blue. Thus these matrices were further manipulated to form a compatible input to the system.

4.2 Color Image Histogram

For a colored image an image signature was first formed by creating histograms for the three color components of the image as can be seen in Figure 4-1. Thus three matrices of dimensions

(256x1) having values of each intensity level corresponding to red, green and blue pixel color was generated which then became an input to the neural network.

The combination of these three histogram forms a unique signature for an image containing a specific disease. These image signatures were fed to neural network reducing the number of input as well as the training time as compared to the approach where the whole image becomes an input and thus increases the complexity and the training time of the neural network.

The input matrix fed to the neural network had dimensions (768x1) for each image. Each row of the matrix contained the value of occurrence of a pixel for red green and blue color in that image. Therefore for 30 images the input matrix became 768x30, where each column represented data for one of the retinal images.

Normalization of above matrix was crucial. As sigmoid function required values between -10 to 10 to be thrashed into a range of 0 to 1, the input matrix was normalized in such a way that it contained values that had a range from -10 to 10. Therefore the application of sigmoid function on the matrix produced results deeply impacting on the training of neural network.

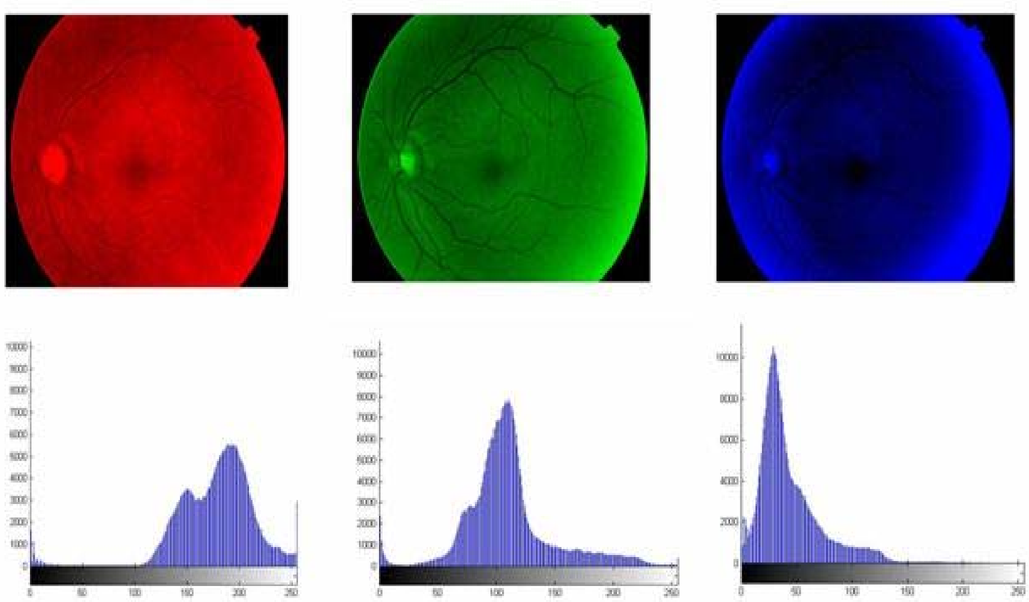

Figure 4-1: Image Color Components and Histograms

4.3 The Neural Network Design

The neural network designed for retinal disease recognition implemented Levenberg-Marquardt back-propagation algorithm. This is an advanced non-linear optimization algorithm. It is reputably the fastest algorithm available for such training. The network had four layers:

4.3.1 Input Layer

The number of inputs to this network was equal to the dimension of the histogram matrix (768 inputs), containing the intensity levels of color components red, green and blue of the images.

4.3.2 2 – Hidden layers

Each hidden layer had 5 neurons. The number of neurons for the hidden layer is only a design issue and has a little effect on the degree of classification. Increasing the number of neurons increases the parallelism of the network, which in turn increases the degree of classification. Due to the complexity of the network, it requires greater time for training. Hence there is a trade off between training time and the accuracy of classification. The hidden layers used the log sigmoid transfer function.

4.3.3 Output Layer

The total number of patterns to be recognized was three, which could be distinguished by three output neurons. The network's output encoded the pattern identified. The output layer used the pure linear transfer function.

4.3.4 Learning Rate and Performance Goal

The learning rate chosen for the training of the network was $5\mathbf{e}^{-2}$. If the learning rate is set too high, the algorithm may oscillate and become unstable. If the learning rate is too small, the algorithm will take too long to converge as it is very sensitive to the proper setting of the learning rate. The performance goal chosen in this case was $1\mathbf{e}^{-5}$. All these measures were based upon experience and previously experimented projects.

4.4 Results

The neural network was trained over fifteen instances of three different patterns. The number of passes (epochs) were limited up to 300. The required performance goal was met within this specified limit and thus training was successfully achieved. The names of the patterns for which the training commenced were Diabetic Retinopathy, AMD Macular Hole and Normal Retina.

Error! Reference source not found. describes the results for 21 randomly selected test cases, unseen by the system:

Table 4-1: Recognition Results of Project RETINA

Test Cases	Number of Correct	Percentage Correct
Diabetic Retinopathy	7 out of 7	100 %
AMD Macular Hole	5 out of 6	83.5 %
Normal Retina	6 out of 8	75%
Overall	18 out of 21	85.7%

From these results it can be easily determined that the system has successfully classified the patterns, intended to be distinguished on unseen examples of to a level as higher as 86%.

Chapter 5

5 Analysis and the Future Work

5.1 Analysis

The purpose of this thesis experiment was to verify that pattern recognition can be achieved successfully using Machine Learning and Artificial Intelligence techniques. At the conclusion of the project, retinal images with and without diseases was recognized and thus classified by the system without much intervening of image processing and such other techniques. The results of the experiment convincingly depict this fact (of successful classification) in form of figures i.e. 86% of test samples were recognized successfully.

The important outcome of this experiment can be seen, it is the amount of data/information that plays a crucial part in successful convergence of the neural network leading to accurate recognition by the system. Thus increase in the amount of information regarding in instances to be recognized, increases the degree of accurate recognition.

Training data might contain errors, it the robustness of the neural network system that it ignores such errors. This brings up an important issue; the decision of the neural network is not based upon individual images for some specific disease. The ability to recognize acquired by the neural network is based upon all the instances used in the training.

This system hence developed cannot be compared with some usual pattern matching machine. This system acquires a pattern out of the given data by itself and uses it for making a decision. And unlike any database the decision is based upon information learned (weights obtained during training).

5.2 Complexity

Since the dawn of the project complexity has been an important issue. The approaches were specifically developed to over this issue as it also affected the design of neural network and eventually the whole system. The equations for complexity observed during the different phases are:

Feed Forward: epochs ((exR*exC)(h1*h2*O)) Equation 5-1

Error Calculation: epochs ((exR*exC)*O) Equation 5-2

Back Propagation: epochs((exR*exC)((h1*h2)3+ O)) Equation 5-3

Classification: h1*h2*O Equation 5-4

where *h1* is hidden layer one neurons, *h2* is hidden layer 2 neurons, *O* is the output neurons, *exC* is the number of training examples, *exR* shows the number of input matrix, *epochs* is the number of passes made.

In case of current scenario, retinal disease recognition using the histogram approach the variables have these values: h1 equals five, h2 equals five, O equals three, exC equals forty five (training instances) and exR equals Seven hundred and sixty eight

It can be easily concluded that the training time is very huge as compared to time taken during classification of an image. Further more if pixel by pixel approach is used instead, *exR* becomes equal to the image dimension which is 768x576, which increases the complexity to huge extend.

Thus this concluded that the histogram approach serves it purpose, compatible data transformation, keeping the complexity at possible minimum.

The performance measure chosen for neural network was $1\mathbf{e}^{-5}$. Anything lesser than this or greater affected the classification of the system. Any value chose for the performance goal lesser than this correctly classified rather a smaller subset of test images. Any value chosen greater than this, failed the system to converge properly and hence didn't classify test images correctly.

As can be seen in Figure 5-1, the percentage of recognition increases even at high levels of noise (dotted line), only if the training has been accomplished taken into consideration the performance goal.

Thus performance goal which is actually depicting the acceptable sum of squared error at the end of training session is also held responsible to a higher degree of classification and thus training sessions with different performance measures must be undertaken. And the measure suited best must be then made constant.

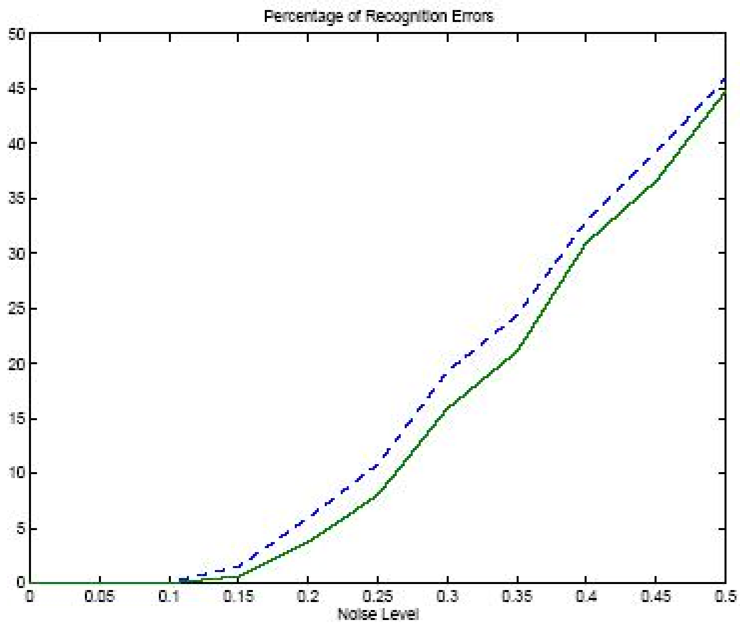

Figure 5-1: Performance Measure's Effect on Classification

5.3 Future Work

Every Intelligent system needs to evolve over time. In theory, neural network lacks the ability to evolve keeping the existing knowledge it has attained. But the phenomenon of evolution can be acquired.

Once the neural network is trained over number of instances, it retains in this state as long as training is not instantiated again. After the training of the neural network, the decision to place the test image in some specific category (i.e. either tag the image with some disease or normal retina), solely depends upon the neural network and the information it has learned. Whereas neural network cannot acquire knowledge about those instances which are not yet introduced as training examples. Thus it cannot evolve without being trained over retinal images containing some disease out of the scope of the system.

The answer to this problem lies in the question. An Intelligent system needs to evolve i.e. to enhance its knowledge base and in this case cover more number of retinal diseases. This can be done by training the system again over these new images to a new state where it has the ability to classify even these new patterns. Thus as new information arrives; the system captures it and evolves to a next stage.

REFERENCES

[1] Victor Rogers and Hoyt Coffee, "Dawn of a New Era", *http://gtalumni.org/Publications/magazine/win98/ageofbio.html*.

[2] Joel Cracraft, "A New AIBs for Age of Biology", *www.aibs.org/bioscience-editorials/*, November 2004.

[3] "Bioinformatics", *http://biotech.icmb.utexas.edu/pages/bioinfo.html*, July 23, 1998.

[4] Economist, "Climbing the Helical Staircase", *http://www.economist.com/surveys/displaystory.cfm?story_id=1647556*, May 27, 2003.

[5] "Biometrics", *http://www.tibs.org/DefinitionofBiometrics.htm*.

[6] Chuck Stewart, "The Retina Project", *http://www.cs.rpi.edu/~stewart/*, October 2004.

[7] Chuck Stewart "Computer Science Seminar", *http://www.rh.edu/~rhb/css/css_0101.html*, January 19, 2001

[8] Chanjira Sinthanayothin, James F. Boyce, Helen L. Cook, Thomas H. Williamson, "Automated localization of optic disc, fovea and retinal blood vessels from digital color fundus images", in *Proc. IEEE Conference of Intelligent Robots and Systems*, Feb. 12, 1999.

[9] "The Eye: Information about vision loss and blindness", *http://www.99main.com/~charlief/Blindness.htm*, September 2002.

[10] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Singapore: Pearson Education, 2004.

[11] *Introduction to Neural Networks: Matlab*, Vers. 7, Math Works, 2004.

[12] *Neuron Model: Matlab*, Vers. 7, Math Works, 2004.

[13] *Back propagation: Matlab*, Vers. 7, Math Works, 2004.

[14] *Levenburg Manquart: Matlab*, Vers. 7, Math Works, 2004.

[15] Tom M. Mitchell, *Machine Learning*, New York: McGraw Hill, 1997.

[16] "Autonomous Land Vehicle in Neural Network (ALVINN)", *http://www.ri.cmu.edu/projects/project_160.html*, April 1996.

[17] Yurij S. Musatenko "MST Image C++ Library", *http://tetis.uazone.net/~yura/*, 1995 – 2005.