# SIGN LANGUAGE AND GESTURE RECOGNITION SYSTEM USING NEURAL NETWORKS

**A THESIS SUBMITTED TO**
**THE GRADUATE SCHOOL OF APPLIED SCIENCES**
**OF**
**NEAR EAST UNIVERSITY**

**by**

**BILAL F. ALNAMRAWI**

**In Partial Fulfillment of the Requirements for**
**the Degree of Master of Science**
**in**
**Computer Information Systems**

**NICOSIA-2012**

**Bilal F.Alnamrawi**: Sign Language and Gesture Recognition System Using Neural Networks

**Approval of the Graduate School of Applied Sciences**

**Prof. Dr. İlkay Salihoğlu**
**Director**

**We certify this thesis is satisfactory for the award of the Degree of Master of Science in Computer Information Systems**

**Examining Committee in charge:**

**Prof.Dr.Rahib H.Abiyev**, Committee Chairmen, Computer Engineering Department, NEU

**Assoc.Prof. Dr. Hasan Demirel**, Committee Member, Electrical and Electronic Engineering Department, Eastern Mediterranean University

**Assoc.Prof. Dr.Nadire Çavuş,** Committee Member, Computer Information Systems Department, NEU

**Assist. Prof.Dr.Umit İlhan**, Committee Member, Computer Engineering Department, NEU

**Assist. Prof. Dr. Boran Şekeroğlu**, Committee Member, Supervisor, Computer Engineering Department, NEU

I hereby declare that all information in this document have been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Bilal Fadhil Al_namrawi

Signature :

Date:

# ACKNOWLEDGEMENTS

First of all, I would like to say Alhamdulillah for finishing the thesis at the right time.

My grateful and special thanks go to, my supervisor

Assist. Prof. Dr. Boran SEKEROGLU and my teacher Assoc.Prof. Dr.Nadire Çavuş for there invaluable guidance and sincere support through out the period I worked with him. I highly appreciate his timeless patience and encouragement.

I would like to thank Lecturers at the faculty of Computer Information Systems for supporting and encouraging me in completing the Master's Degree successfully.

I would also like to express my appreciation to my grateful friends for guiding me throughout my studying time. They also supported me and encouraged me to learn new things throughout these years and not letting me to procrastinate my works especially who made my stay in Cyprus enjoyable.

I find it an opportunity to express my love and deep appreciation to my dear family: my father, my mother, my brothers and my sisters who have been of great support to me through out my life, and to whom I owe any success I make.

## ABSTRACT

Sign Language is considered very important for the communication between dump and deaf community and normal people. The development of computer application and use of an Artificial Neural Network in many fields of life has encouraged researchers to develop intelligent system that can facilitate the communication with deaf people. The Artificial Neural Network has proved effacing in many applications in scientific and social fields and has been introduced into the sign language recognition. This work proposes the use of Back propagation for sign language recognition system. This system try to recognition the sign of 10 numbers from zero to nine, for each sign of number has been used 4 images for training and 4 images for testing in Artificial Neural Network system. There are three experiments with the different images in this system, in the first and second experiments used sketches, in the third experiment used real images. The system provides means for training the input sign language and gesture images, then with other set of input for testing the system and make the identification process. The training process will be carried out by using a back propagation learning algorithm. Several experiments will be carried out using different images in training and testing. The system implementation will be simulated using Matlab programming software tool. The result which are obtained are promising, While the best experiment achieving high performance result with 97% identification rate and 84% accuracy. The other results were obtained and discussed in chapter four in this thesis.



Keywords:

Artificial Neural Networks, Sign Language Recognition, Gesture Recognition, Finger Spelling, Computer Application.

# ÖZET

İşaret dili, sağır ve dilsiz insanların normal kişilerin arasındaki iletişim için çok önemli olarak nitelendirilmektedir. Birçok alanda bilgisayar uygulamalarının geliştirilmesi ve yapay sinir ağlarının kullanımı, araştırmacıları sağır insanlarla iletişim kurabilmek için akıllı sistemler geliştirmeleri için cesaretlendirmiştir. Yapay sinir ağları birçok bilimsel ve sosyal alanda etkinliğini ispatlamıştır ve İşaret Dili Tanımlaması Sisteminde de kullanılmıştır. Bu çalışma, Geri Yayılmalı Ağların İşaret Dili Tanımlaması Sisteminde kullanılmasını önermektedir. Sistem, sıfırdan dokuza kadar olan sayıların tanımlanmasını her rakam için yapay sinir ağlarında 4 alıştırma 4 de test resmi kullanarak denemektedir. Farklı resimler kullanarak 3 tane deney yapılmıştır. İlk iki deneyde, işaretlerin çizimi, son deneyde ise gerçek görüntüleri kullanılmıştır. Sistem, alıştırma için işaret dili ve hareketlerini giriş olarak alıp, test için farklı işaretler ve hareketler kullanarak bunların tanımlanmasını sağlamaktadır. Alıştırma işlemi geri yayılmalı öğrenme algoritması kullanılarak yapılacaktır. Alıştırma ve test aşamalarında farklı resimler kullanarak birçok deney gerçekleştirilecektir.
Sistem uygulaması, Matlab yazılım araçları kullanılarak simule edilecektir. Elde edilen sonuçlar %97 tanımlama ve %84 hassasiyete ulaşarak umut verici olmuştur.

Anahtar Kelimeler: Yapay sinir ağları, işaret dili tanımlaması, Hareket tanımlama, Bilgisayar uygulaması, parmak hecelemesi

# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS** ........ I
**ABSTRACT** ........ II
**TABLE OF CONTENTS** ........ III
**LIST OF TABLES** ........ V
**LIST OF FIGURES** ........ VI
**LIST OF ABBREVIATIONS** ........ VII
**CHAPTER ONE: INTRODUCTION AND LITERATURE REVIEW**
1.1 Introduction ........ 1
1.1.1 Research Goals ........ 3
1.1.1.1 Sign Language and Gestures Recognition ........ 3
1.1.1.2 Neural Networks ........ 4
1.1.2 Thesis Layout ........ 4
1.2 Literature Review ........ 5
1.2.1 Overview ........ 5
1.2.2 Introduction to Sign Language ........ 5
1.2.3 History of Sign Language ........ 6
1.2.4 Structural Components of Signs ........ 9
1.2.5 Natural Language ........ 10
1.2.6 Properties of Language ........ 11
1.2.7 Sign Language in Different Communities ........ 11
1.2.7.1 American Sign Language (ASL) ........ 11
1.2.7.1.1 Finger Spelling ........ 12
1.2.7.2 British Sign Language (BSL) ........ 14
1.2.7.2.1 Relationships with Other Sign Languages ........ 14
1.2.7.2.2 Finger Spelling ........ 15
1.2.7.3 Sign Language vs Oral Language ........ 15
1.2.8 Facial Action Coding System (FACS) ........ 17
1.2.9 Object Recognition ........ 19
1.2.9.1 Large Object Tracking ........ 19
1.2.9.2 Shape Recognition ........ 19
1.3Goals ........ 20
1.4Summary ........ 20
**CHAPTER TWO: NEURAL NETWORKS**
2.1 Overview ........ 21
2.2 Introduction to Artificial Neural Networks ........ 21
2.3 Teaching an Artificial Neural Network ........ 23
2.3.1 Supervised Learning ........ 23
2.3.2 Unsupervised Learning ........ 25
2.3.3 Learning Laws ........ 27
2.4 Multilayer Perceptron ........ 28
2.5 Back Propagation Neural Network ........ 30
2.5.1 Structure of Back propagation Network ........ 30
2.5.2 Back Propagation Network Algorithm ........ 31
2.5.2.1 Feed Forward Calculation ........ 32
2.5.2.2 Error Back Propagation Calculation ........ 33
2.5.3 Discussion Some Important Issues ........ 36

2.5.3.1 Input Normalization and Weights Initialization ........ 36
2.5.3.2 Training Conversion Criteria ........ 37
2.5.3.3 Techniques and Arising Problems ........ 38
2.5,3,4 Generalization ........ 39
2.6 Summary ........ 39
**CHAPTER THREE: SIGN LANGUAGE RECOGNITION SYSTEM**
3.1 Overview ........ 40
3.2 The Sign Languages System ........ 41
3.2.1 Database Collecting ........ 41
3.2.2 Preprocessing ........ 43
3.3 Image Reading ........ 45
3.3.1 Image Vectorization ........ 45
3.3.2 Using Matlab ........ 46
3.4 Implementation of the Neural Network ........ 46
3.4.1 Define BPNN Architecture and Design ........ 46
3.4.2 Formulation of Weight Adjustment ........ 47
3.4.3 Define the Learning Rate and Momentum Factor ........ 47
3.4.4 Define the Maximum Error ........ 47
3.5 Training Using Back Propagation ........ 47
3.6 Testing SLR ........ 51
3.7 Summary ........ 52
**CHAPTER FOUR: EXPERIMENTAL RESULTS AND DICUSSION**
4.1 Overview ........ 53
4.2 General Experimental Setup ........ 53
4.3 First Experiment ........ 53
4.4 Second Experiment ........ 58
4.5 Third Experiment ........ 62
4.6 Discussion ........ 66
4.7 Summary ........ 66
**CONCLUSION** ........ 67
**REFERENCES** ........ 68
**APPENDIX I: DATABASE** ........ 72
**APPENDIX II: MATLAB SOURCE CODE** ........ 78

# LIST OF TABLES

**Table 4.1** Training Parameters and Training & Testing Time First Experiment. .......... 54
**Table 4.2** Training Recognition Rate and Accuracy for First Experiment . .................. 56
**Table 4.3** Testing Recognition Rate and Accuracy for First Experiment ...................... 57
**Table 4.4** Training Parameters and Training & Testing Time Second Experiment....... 59
**Table 4.5** Training Recognition Rate and Accuracy for Second Experiment ............... 60
**Table 4.6** Testing Recognition Rate and Accuracy for Second Experiment.................. 61
**Table 4.7** Training Parameters and Training & Testing Timing Third Experiment. ..... 63
**Table 4.8** Training Recognition Rate and Accuracy for Third Experiment.................. 64
**Table 4.9** Testing Recognition Rate and Accuracy for Third Experiment.................... 65

# LIST OF FIGURES

**Figure 1.1** ASL Finger Spelling Hand Shapes ..... 13
**Figure 1.2** British Sign Language Finger Spelling Hand Shapes ..... 15
**Figure 1.3** Washoe the First Chimpanzee Learn Sign Language ..... 18
**Figure 2.1** Single - Input Artificial Neuron ..... 23
**Figure 2.2** Architecture of Supervised Artificial Neural Network ..... 24
**Figure 2.3** Architecture of Unsupervised Artificial Neural Network ..... 26
**Figure 2.4** Architecture of Multilayer Perceptron ..... 29
**Figure 2.5** Block Diagram of Back Propagation Network ..... 30
**Figure 2.6** Back Propagation Network Architecture ..... 31
**Figure 2.7** A model Neuron Structure ..... 32
**Figure 2.8** Sigmoid Activation Function ..... 33
**Figure 2.9** Typical Curve between Overall Error and A single Weight ..... 34
**Figure 3.1** Example of original signal Images ..... 42
**Figure 3.2** Example of Training signal Images ..... 42
**Figure 3.3** Example of Testing signal Images ..... 43
**Figure 3.4** explain the RGB image ..... 44
**Figure 3.5** explain the Grayscale image ..... 44
**Figure 3.6** Explain the Binary Images ..... 44
**Figure 3.7** Neural Network Design ..... 48
**Figure 3.8** Block Diagram of SLR ..... 49
**Figure 3.9** Flow Chart Diagram of the System (Training Process) ..... 50
**Figure 3.10** Example of Testing Image ..... 51
**Figure 3.11** Flow Chart Diagram of the System (Testing Process) ..... 51
**Figure 4.1** An Example of Training and Testing sketches for Experiment One ..... 54
**Figure 4.2** First Experiment training performance Curve ..... 55
**Figure 4.3** An Example of Training and Testing sketches for Experiment Two ..... 58
**Figure 4.4** Second Experiment training performance Curve ..... 59
**Figure 4.5** An Example of Training and Testing images for Experiment Three ..... 62
**Figure 4.6** Third Experiment training performance Curve ..... 63

# LIST OF ABBREVIATIONS

| | |
|---|---|
| **SLR** | Sign Language Recognition |
| **ANN** | Artificial Neural Networks |
| **NN** | Neural Networks |
| **ASL** | American Sign Language |
| **BSL** | British Sign Language |
| **WFD** | World Federation of the Deaf |
| **SE** | Signed English |
| **ISL** | Irish Sign Language |
| **UK** | United Kingdome |
| **VESPA** | Volume, Echo, Speed, Pitch, Attenuation |
| **FACS** | Facial Action Coding System |
| **MLP** | Multilayer Perceotron |
| **I/O** | Input & Output |
| **BPNN** | Back Propagation Neural Networks |
| **RGB Images** | Red Green Blue Images |

# CHAPTER ONE

## INTRODUCTION AND LITERATURE REVIEW

### 1.1 INTRODUCTION

In computer recognition of spoken language, speech data is captured using a microphone connected to an analog-to-digital converter. Similarly, a data capturing device is required in order to recognize sign language, in this case measuring the position and movement of the signer's hands. Until recently such technology did not exist, but the growing interest in virtual reality has lead to the development of sensing systems for measuring the movements of the human body, and in particular the human hand.

There are two separate sets of values which need to be measured for sign language recognition (and also for many virtual-reality applications). The first is the flexion of the individual joints of the fingers and wrist which define the posture of the hand. The second is the spatial positioning and orientation of the hand as a whole. In general different technologies have been used to measure these two sets of attributes.

The aim of this research is to develop intelligent system for the automatic recognition of sign language, based on artificial neural networks techniques.

The research is motivated by two contrasting but complementary goals. The first is that a sign language system would be potentially beneficial in aiding communication between members of the deaf community and the hearing community. The second is that the process of developing such a system using neural networks gives opportunities for studying and extending the networks themselves.

The first motivating factor is the possibility of reducing the communications barrier which exists between the deaf and hearing communities. The problems that deaf people encounter in trying to communicate with the general community are well documented. Moscovitz and Walton use the term deaf in two distinct senses distinguished by whether the word is capitalized or not[1]. In its uncapitalized form deaf refers purely to an individual's ability to hear, as it would be used in common parlance. The capitalized form deaf is used to indicate the cultural aspects of being deaf. This convention is also used within this thesis.

In many ways the deaf community is similar to an ethnic community in that they form a subgroup within society, complete with its own culture and language (in this case sign language). People who become deaf later in life after learning a spoken language in general do not use sign language as much and are less involved in the deaf community than those whose hearing loss occurred earlier in life. The inability to hear means that many deaf people do not develop good skills in the English language and prefer not to use it. This is because the sign languages most commonly used within the deaf community are not grammatically related to English[2].

In addition very few hearing people have much knowledge of sign language, and so communication between sign-language users and hearing people poses many problems. For this reason the deaf community tends to be insular and somewhat separate from the rest of society. When it is necessary to communicate with hearing people (for example when shopping) signers often have to resort to pantomimic gestures or written notes to communicate their needs, and many are uncomfortable even in using notes due to their lack of English writing skills.

An automated sign language translation system would help to break down this communication barrier (in much the same way that an automated English-to-French translator would help Australian tourists visiting Paris to communicate). Ideally such a system should allow signers to use their native sign language, as this language is an integral component of deaf culture.

The aim of this thesis is not to develop a full sign language to English translation system; such a task is too large and complex to attempt at this stage. Instead the aim is to create a prototype system for the recognition of signs, and developing techniques which could later be incorporated into a more complete translation system. It is also envisioned that the system developed could be adapted for use as a training tool to aid hearing people attempting to learn sign language, And the use of ANN system to solve this problem and similar problems that we face in our lives. A large proportion of neural networks research has been performed on toy or contrived problems which may bear little relevance to real-world tasks. Whilst this research has been invaluable in developing the basic techniques used in neural networks, attempting to apply these techniques to a real problem is seen as likely to produce new insights into neural network methodologies.

In particular the temporal component involved in signing forms a challenging task for neural networks as the majority of research so far has been focused on purely static problems. Attempting to address this aspect of signing aims to yield insights into the methods of extending neural networks to this temporal domain.

These twin motivating factors influence the directions taken during this research, and are reflected in this thesis which addresses both the performance of the final system for recognizing signs, and also the issues related to neural networks arising from the development of this system.

### 1.1.1 Research Goals

Clearly the aims of this thesis also reflect the two factors motivating the work, and try to develop sign language to English translation system; such a task is too large and complex to attempt at this stage. Instead the aim is to create a prototype system for the recognition of signs, and developing techniques which could later be incorporated into a more complete translation system. It is also envisioned that the system developed could be adapted for use as a training tool to aid hearing people attempting to learn sign language, And the use of ANN system to solve this problem and similar problems that we face in our lives, therefore need to be discussed separately.

#### 1.1.1.1 Sign Language and Gesture Recognition

One aim of this research is to improve on the results obtained by previous work on hand-gesture and sign-language recognition. Existing systems have two main shortcomings, namely a relatively small vocabulary and a capacity to recognize only static hand shapes or simple motions. Development of the SLR system is focused on improvements in these areas.

Previous systems have generally supported only a relatively small vocabulary (being the number of signs or gestures recognized) and the size and contents of the vocabulary are fixed. One of the main goals of this research is a desire to design the system in a manner which will allow the vocabulary to be extended in the future without requiring major modification of the system.

The second goal is to increase the complexity of the signs recognized, particularly with regard to their motion component. The majority of the research into hand-gesture recognition has concentrated on static hand configurations or very simple motions, rather than the potentially complicated movements involved in Deaf sign languages.

A further problem which has been rarely researched is the automatic segmentation of signs from within a continuous sequence of signing. Most systems developed so far have dealt only with individual signs and therefore have not addressed the issue of distinguishing the end of one sign from the start of the next sign. The ability to handle continuous signing will be a fundamental quality of any practical sign-language recognition system.

Although it is considered to be outside the primary scope of this thesis, a possible method of tackling this problem arises naturally out of the development of the system. Therefore some preliminary results are included amongst the discussion of future work.

#### 1.1.1.2 Neural Networks

The goals related to the use of neural networks are less easily defined in advance, as it is not possible to anticipate the specific issues which would arise during the development of the system. However there are some general issues which could be seen as likely to be encountered during this development process.

The issue is the use of neural networks to recognize temporal patterns. As described above one of the goals related to gesture recognition is the recognition of complex hand motions, and therefore it is necessary to consider how neural networks can be extended from the relatively well explored domain of spatial pattern processing to handle spatiotemporal patterns.

### 1.1.2 Thesis Layout

The chapters of this thesis can be divided into four logical sections.

Chapter One Introduction and Literature Review, In this chapter discuss what sign language, what is its history, how they have evolved and how they have developed , who use them, how to deal with technology and how computers that distinguishes it and translate them into the language of normal humans.

The second chapter Artificial Neural Network: In this chapter discuss What artificial neural networks when and how It found , how they developed, How can you help us to solve some problems such as the problem of discrimination signals deaf and dumb, also discussed back propagation function and what work in the system.

Third Chapter system Discussion: In this chapter discuss system recognition signals the deaf and dumb in detail by using ANN and the back propagation function specifically, And we can see a selection of images database and the general structure of the system.

The fourth chapter discuss the results that were obtained from experiments carried out in this system.

## 1.2 LITERATURE REVIEW

### 1.2.1 Overview

In this chapter an explanation of history of sign language. Explain the term of the sign language and its importance, the problems that are faced in this kind of language, the areas that could be used in. In this chapter we also explaining the kind of sign language and Structural components of signs. Also the natural language and its representation in sign, Sign language in different communities, American and British sign language. Facial Action Coding System will be explain and one of the most common story of learning sign language to the animal, finally object recognition.

### 1.2.2 Introduction to Sign Language

Sign language is a language that uses hand signals, hand gestures, body language and lip patterns instead of sound to convey meaning; simultaneously combining hand shapes, orientation and movement of the hands, arms or body and facial expressions to express fluidly a speakers thoughts[1].

Sign language is a form of manual communication which has developed as an alternative to speech amongst the deaf and vocally impaired. Although many deaf people can speak clearly (particularly those whose hearing impairment was acquired after early childhood) and can use skills such as lip-reading when communicating with hearing people, such methods of communication are generally inappropriate for communication within the deaf community. Therefore the hands have become the primary means of communication within these communities[2].

### 1.2.3 History of Sign Language

Sign language, specially formulated for the deaf people, makes use of finger spellings, body language, lip pattern and manual communication, to convey the meaning. It mainly involves the use of orientation and movement of hands. The language can be taught only by a person who is specially trained in it. Today, the differently-abled people can communicate to the rest of the world as easily and effectively as the able bodied. The credit goes to the sign language, which was developed many years ago. following lines and get some interesting information on the background, origin and history of sign language following [3]. The origin of sign language can be traced back to the beginning of the Christian era, when illustrations of hand and finger positions were used to convey the meaning of different words. It is believed that sign language was used in Latin Bibles of the 10th century. However, the recorded history of the language dates back to the 17th century only. In 1620, Juan Pablo Bonet, a Spanish priest, published a book named "Reduction of Letters and Art for Teaching Mute People to Speak". This was regarded as the first modern treatise of Phonetics and Logo podia. It established oral education for the deaf and the dumb [4].The manual signs, mentioned in Bonet's book, came to be used for teaching the deaf and dumb people. Inspired by the sign language of Bonet, Charles-Michel de l'Épée published a book, containing the alphabets, in the 18th century. The first public school for deaf children was established in 1755, in Paris, by Abbé de l'Épée. Through his organization, Abbé intended to help the deaf communicate with the rest of the world, by using gestures, hand signs and finger spellings. After learning the signs that were already prevalent in Paris, Abbé developed his own sign system[5].

Another important development in the history of sign language came with the establishment of the first permanent American School for the deaf in Hartford, Connecticut. It was founded by Thomas Hopkins Gallaudet, in the year 1817. Forty years later, in 1857, Gallaudet's son, Edward Miner Gallaudet, founded a school for the deaf, in Washington, DC. The school came to be known as Gallaudet University in 1864. This further strengthened the education process of the deaf and dumb people[5].

Today, there are a number of sign languages prevalent in the world. The deaf communities across the world communicate differently, through their own version of sign language. One of the most popular sign languages is the American Sign Language (ASL), which is prevalent in the United States as well as some areas of Mexico and Canada. There also exists an International Sign Language, which, as the name suggests, is used at used at international meetings, such as the World Federation of the Deaf (WFD), as well as informally, when traveling and socializing[6].

The hands are also widely utilized during communication between the vocal community, with gestures often used to augment speech. However such gestures bear very little similarity to the signs that make up sign language. First these gestures serve only an auxiliary role, rather than being the primary focus of communication as they are in signing. Second such gestures have no defined meaning, but instead are interpreted in the context of the accompanying speech. In contrast the hand gestures used in sign language are highly formalized, with each gesture having a defined meaning, in much the same manner as the spoken or written word. This allows the construction of sign-language dictionaries in which each sign of the language is equated to one or more words in a spoken language (which are known as the gloss of that sign)[7].

Hence a sign language consists of a vocabulary of signs in exactly the same way as a spoken language consists of a vocabulary of words. No one signer will be familiar with all of the vocabulary, and there may be regional variations in the formation or meaning of signs, similar to the variations of dialect and accent found in spoken languages. In addition to this vocabulary of signs, a sign language usually provides a means of spelling words for which there is no equivalent sign in common usage, such as people's names or places. Signs exist for well known places (such as major countries and local cities), but may not exist (or not be known) for places less commonly referred to, such as cities in another country. In this case the signer will spell out the written version of the place name on their hands. Such finger spelling systems (or manual alphabets) consist of a distinct hand position or gesture for each letter of the alphabet. These systems are a much slower means of communication than regular signing and hence are used only when necessary, and are often accelerated by the deliberate omission of some letters from the word being spelled[6].

A common assumption of people unfamiliar with Deaf culture is that a single international sign language is used by all signers. In fact the manual languages are even more fragmented than vocal languages, with distinctly different sign languages being used in countries with the same spoken language. For example, English is the primary spoken language in Australia, the United Kingdom and the United States of America but three independent sign languages are in common use in these countries (Auslan, British Sign Language (BSL) and American Sign Language (ASL) respectively)[7].

As well as using different signs, different sign languages will often use alternative manual alphabets as well. For example the manual alphabet used in conjunction with ASL involves only a single hand, whereas the finger spelling system used in Australia is two-handed[5].

Even within a single country it is quite common for two or more manual languages to be in use. For example in Australia both Auslan and Signed English (SE) may be used. These languages are representatives of two different approaches to manual communication, and the distinction between them is of some interest when developing an automated sign recognition system.

Signed English is a manual representation of the English language. Each sign in SE corresponds to an English word, and standard English grammar is used. Hence SE has basically the same relationship to spoken English as does written English – it is a different representation of the same language.

The use of signing has many uses from air traffic control and sports to communicating with babies. Sign language as a replacement for oral language has developed in deaf communities, people who are hard of hearing and their and friends and families[8].

There is no common sign language and every deaf community has its own variation of the language, each language has complex grammar and is remarkably different from the grammars of spoken languages. Hundreds of sign languages are in use around the world.

Sign language has not directly descended from spoken language there is not a direct mapping between sign language and an object or meaning in an oral language also though there are some trivial examples[9].

There are such sign languages that have mappings between hand signals and words in spoken languages such as Signed English. The use of this type of sign language is not enough to be used as a communication tool and is often the first image people have when they think of sign language; sign language is far more complex.

### 1.2.4 Structural Components of Signs

Although primarily a manual language, signing also relies on non-manual features such as facial expression and body language to provide some of the subtle nuances which make human communication so expressive. Facial actions such as raising of the eyebrows, smiling or puffing out the cheeks can modify the meaning of the sign being performed, in much the same manner that variations in the voice or use of coverable gestures alter the meaning of words being spoken. Interpretation of subtle variations such as these was felt to be beyond the scope of this project, and therefore only the manual components of signing will be considered in this thesis[10].

Individual signs may involve the use of one or both hands. If only one hand is required, the same hand will consistently be used by a particular signer and this is referred to as their dominant hand. The majority of signers are right-hand dominant and therefore all illustrations and examples used in this thesis will also use the right hand. Similarly the choice of sensing device used for this research restricts the SLR system to recognition of signs performed by right-handed signers. However the system could easily be adapted to a left-handed user if the appropriate sensing technology was available by the addition of some simple pre-processing of the input data to the SLR system.

The signs depicted illustrate the possible combinations of one and two hands that may be used in signing. Water uses only a single hand. Upward demonstrates both hands performing the same action, whilst shelf illustrates the hands making symmetrically-opposite movements. In iceskating both hands perform the same motion, but at alternating times. In bully the subordinate (left) hand serves as a base for the dominant (right) hand[9].

Analysis of individual signs indicates that their manual component can be described in terms of four features the hand shape, orientation, place of articulation and motion.

Hand shape refers to the position of the joints of the fingers and wrist. Each sign language uses its own distinct set of hand shapes, although the physical structure of the hand means that the majority of hand shapes are common to most languages[10].

Orientation is the angle of the hand with respect to some fixed plane. Generally the different orientation possibilities are described by defining the relationship between the signer's hand, and the body (eg weigh has the palm facing upwards and fingers pointing away from the body)[11].

The place of articulation (or location) of a sign refers to its spatial location with respect to the signer's body. Signs can be made on or near particular parts of the body, or in the space in front of the chest which is referred to as neutral space.

Motion is the most complex aspect of a sign to describe, as it can consist of variation over time in any of the other three aspects. A sign may involve a transition from one hand shape to another (eg ten), or wiggling of the fingers while maintaining the same basic hand shape (eg piano). The orientation may change either through a single twisting of the wrist (eg air) or through repeated twisting (eg helicopter). Movement from one place of articulation to another, or through neutral space is a common component of signs (eg elephant). Signs may also incorporate more than one of these motions (eg the sign for involves a change in both hand shape and orientation).

### 1.2.5 Natural Language

In order to understand and appreciate what Sign Language has become we first need to look at language in general. Sign language as with other languages e.g. English, Spanish are natural languages, natural languages are languages that have evolved over time rather than being formally constructed for a purpose.

Oral and Sign language can be thought of as natural phenomena where the use of the language has evolved through a common need in a culture. The act of oral, written or hand gestured language is both part of the language and the medium to which it is conveyed.

Oral language has been around for tens of thousands of years and the first form of written language was cuneiform[12].

Language is a dynamic set of visual, auditory, or tactile symbols of communication and the elements used to manipulate them. Language can also refer to the use of such systems as a general phenomenon. Language is considered to be an exclusively human mode of communication; although other animals make use of quite sophisticated communicative systems, none of these are known to make use of all of the properties that linguists use to define language[13].

### 1.2.6 Properties of Language

A set of commonly accepted symbols is only one feature of language, all languages must define the structural relationships between these symbols in a system of grammar. Rules of grammar are what distinguish language from other forms of communication. They allow a finite set of symbols to be manipulated to create a potentially infinite number of grammatical utterances[14].

### 1.2.7 Sign Language in Different Communities

In this section we will look at Sign Languages from different communities, in the hope to discover common characteristics that can be exploited.

The scope of this investigation will give a description of each language and its Finger spelling counterpart it will not investigate the actual signing language being used as this is not needed for the project scope.

#### 1.2.7.1 American Sign Language (ASL)

American Sign Language (ASL) is the dominant sign language of the Deaf community in the United States, in the English-speaking parts of Canada, and in parts of Mexico. Although the United Kingdom and the United States share English as a spoken and written language, British Sign Language (BSL) is quite different from ASL, and the two sign languages are not mutually intelligible[15].

ASL is a natural language as proved to the satisfaction of the linguistic community by William Stokoe, and contains phonology, morphology, semantics, syntax and pragmatics just like spoken languages. It is a manual language or visual language, meaning that the information is expressed not with combinations of sounds but with combinations of hand shapes, palm orientations, movements of the hands, arms and body, location in relation to the body, and facial expressions. While spoken languages are produced by the vocal cords only, and can thus be easily written in linear patterns, ASL uses the hands, head and body, with constantly changing movements and orientations. Like other natural sign languages, it is three dimensional" in this sense. ASL is used natively and predominantly by the deaf and hard-of-hearing of the United States and Canada.

**1.2.7.1.1 Finger Spelling**

ASL includes both finger spelling borrowings from English, as well as the incorporation of alphabetic letters from English words into ASL signs to distinguish related meanings of what would otherwise be covered by a single sign in ASL. For example, two hands trace a circle to mean a group of people. Several kinds of groups can be specified by hand shape: When made with C hands, the sign means class, when made with F hands, it means 'family'. Such signs are often referred to as initialized signs because they substitute the first initial an English word as the hand shape in order to provide a more specific meaning [16].

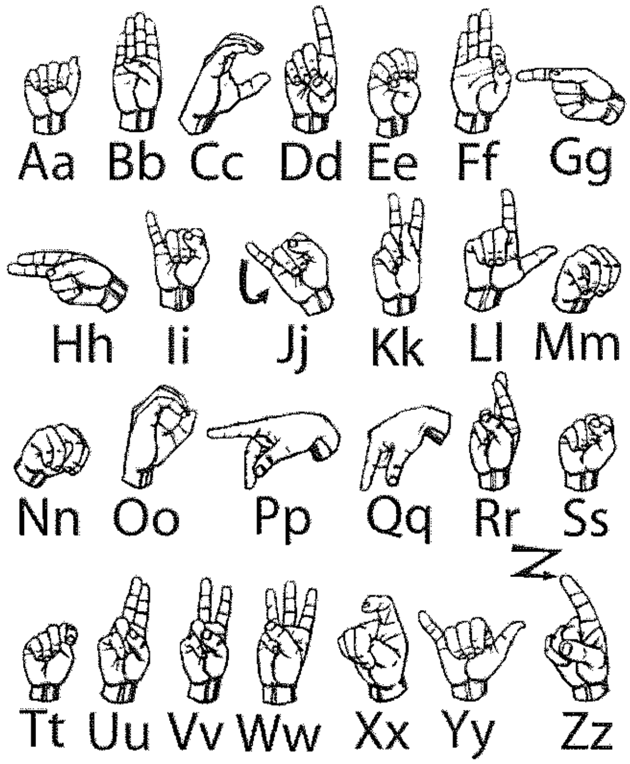



Figure 1.1 ASL Finger Spelling Hand Shapes

When using alphabetic letters in these ways, several otherwise non-phonemic hand shapes become distinctive. For example, outside finger spelling there is but a single fist hand shape, with the placement of the thumb irrelevant, but within finger spelling the position of the thumb on the fist distinguishes the letters A, S, and T. Letter-incorporated signs which rely on such minor distinctions tend not to be stable in the long run, but they may eventually create new distinctions in the language.

Finger spelling has also given way to a class of signs known as loan signs or borrowed signs. Sometimes defined as lexicalized finger spelling, loan signs are somewhat frequent and represent an English word which has, over time, developed a unique movement and shape. Sometimes loan signs are not even recognized as such because they are so frequently used and their movement has become so specialized. Loan signs are sometimes used for emphasis (like the loan sign #YES substituted for the sign YES), but sometimes represent the only form of the sign (e.g., #NO). Probably the most commonly used example of a loan sign is the sign for NO. In this sign, the first two fingers are fused, held out straight, and then tapped against the thumb in a repeated motion. When broken down, it can be seen that this movement is an abbreviated way of finger spelling N-O-N-O. Loan signs are usually glossed as the

English word in all capital letters preceded by the pound sign(#).Other commonly known loan signs include #CAR, #JOB, #BACK, #YES, and #EARLY .

#### 1.2.7.2 British Sign Language (BSL)

British Sign Language (BSL) is the sign language used in the United Kingdom (UK), and is the first or preferred language of deaf people in the UK; the number of signers has been put at 30,000 to 70,000. The language makes use of space and involves movement of the hands, body, face and head. Many thousands of people who are not Deaf also use BSL, as hearing relatives of Deaf people, sign language interpreters or as a result of other contact with the British Deaf community[17].

##### 1.2.7.2.1 Relationships with Other Sign Languages

Although the United Kingdom and the United States share English as the predominant spoken language, British Sign Language is quite distinct from American Sign Language (ASL). BSL finger spelling is also different from ASL, as it uses two hands whereas ASL uses one. BSL is also distinct from Irish Sign Language (ISL) (ISG in the ISO system) which is more closely related to French Sign Language (LSF) and ASL. It is also distinct from Signed English, a manually coded method expressed to represent the English language.

The sign languages used in Australia and New Zealand, Auslan and New Zealand Sign Language, respectively, evolved largely from 19th century BSL, and all retain the same manual alphabet, grammar, and similar lexicon. BSL, Auslan and NZSL together may be called BANZSL. Makaton, a communication system for people with cognitive impairments or other communication difficulties, was originally developed with signs borrowed from British Sign Language. The sign language used in Sri Lanka is also closely related to BSL despite the spoken language not being English, demonstrating the distance between sign languages and spoken ones[18].

BSL users campaigned to have BSL recognized on a similar level to Welsh, Scottish Gaelic, and Irish. BSL was recognized as a language in its own right by the UK government on 18 March 2003, but it has no legal protection, so therefore is not an official language of the United Kingdom[17].

**Usage**

BSL has many regional dialects. Signs used in Scotland, for example, may not always be understood in southern England, and vice versa. Some signs are even more local, occurring only in certain towns or cities (such as the Manchester system of number signs). Likewise, some may go in or out of fashion, or evolve over time, just as terms in spoken languages do [18].

### 1.2.7.2.2 Finger Spelling

BSL finger spelling is also different from ASL, as it uses two hands whereas ASL uses one [17].

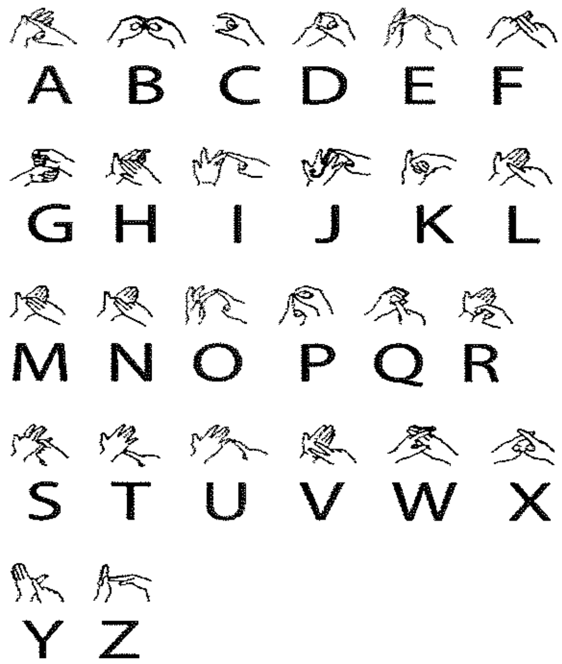



Figure 1.2 British Sign Language Finger Spelling Hand Shapes

### 1.2.7.3 Sign Language vs Oral Language

Most signers use the hands, shoulder and face simultaneously though the shoulders are not involved in complex movement and the facial expressions though extremely complex are a trivial matter to categories, they will be discussed later in Facial recognition and FACS. Why is the spoken language different to sign language?

The English spoken language like most oral languages is a linear language, sentences are structured using words that donate sounds and these are chained together[19].

Speech is then controlled using grammar much like written grammar used in written languages, then speech is broken down into VESPA (Volume, Echo, Speed, Pitch, Attenuation) changing the level of these can change the meaning how the words should be interpreted.

Using the sentence below as an example, The road was very wet and narrow this sentence has the road and 2 adjectives wet and narrow both those words describe the road and come after one another linearly.

When spoken they also have to be pronounced one after the other though the VESPA changes depending on the context, as seen there is a direct relationship between writing and speaking that sentence.

When the following sentence wants to be signed the adjectives can be communicated at the same time, this is non linear and allows for communication of certain words to occur in parallel.

The words wet and narrow are communicated at the same time not one after the other. This allows sign language to be one of the most efficient methods for communication it is far more efficient then English and other oral languages.

It can be said that a new word meaning wet and narrow could be made in English and used to improve efficiency but that is not the point, the point is that sign language has no direct translation to a spoken language (More evidence of this will be given in Washoe) [20].

They are 2 completely different forms of communication, the one thing they do have in common is they are both used in the same environment ,they both convey the same meaning, be it the environment, feels or things.

This is why an interpreter can be used to bridge the gap between sign and spoken languages, sign language should be seen as something much more than a language used by the deaf and hard of hearing [19].

### 1.2.8 Facial Action Coding System (FACS)

We can make a facial expression of an emotion without even thinking about it, facial expressions help compose Sign Language.

If most facial expressions are emotions (The face can make roughly 10,000 facial expressions about 3,000 of these are meaningful ) then is their some way to catalogue these emotions, to pick out what emotions occur to complement which signs being made?

In the research I have carried out, Not come across any studies or publications which label or number the facial expressions as used specifically in sign language. The first step to creating a mapping between facial expressions that are emotions to signs would be to catalogue emotional facial expressions [21].

The Facial Action Coding System (FACS) was developed by Paul Ekman and Silvan Tomkins in the1960's, they developed a way to read people's faces by reading their expressions. They could tell if somebody was telling the truth just by looking at their face. They then created a taxonomy of facial expressions, there are over 10 thousand different facial configurations (some facial expressions cannot be made violently) most of the faces done make sense and are just made by children. Out of the 10000 facial configurations about 3000 mean something. It can then easily been seen that a sign could be separated into the timing and sequence of the faces action unit [21].

**Is Sign Language Distinct to Humans?**

During the study of Sign Language we have assumed that it was practiced purely by Humans, could a broader look at this widen our understanding?. Is the problem of sign language through human interpretation too large a problem space?

**Washoe**

Washoe was the first non-human in history to acquire a human language. Washoe (c. September 1965 – October 30, 2007) was a chimpanzee who was the first non-human to learn to use a human language, that of American Sign Language. She also passed on some of her knowledge to her adopted son Louis [20].

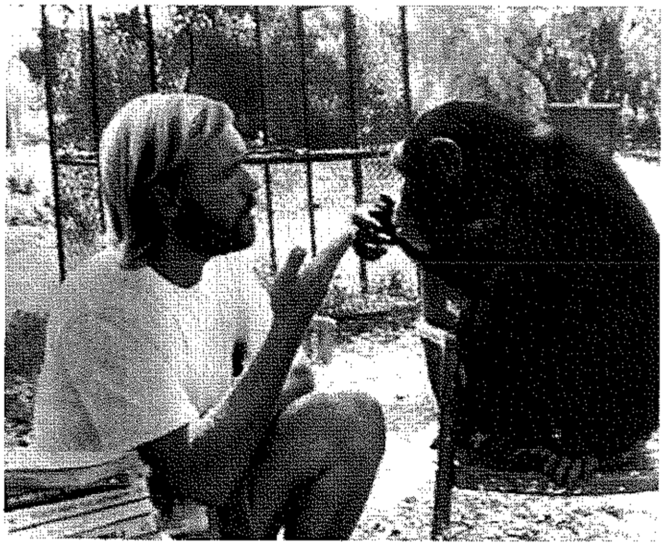


Figure 1.3 Washoe the First Chimpanzee Learn Sign Language

It is reported that Washoe could reliably use 250 signs. For Washoe to be considered reliable on a sign, it had to be seen by 3 different observers in 3 separate spontaneous instances in the correct context and used appropriately. Following those observations, it had to be seen 30 days in a row to be added to her sign list.

In addition to individual signs, Washoe displayed the ability to combine signs in novel and meaningful ways. For example, she referred to her toilet as dirty and the refrigerator as open food drink, even though the scientists around her always called them potty chair and cold box, this type of linguistic modification is similar to tool modification of wild chimpanzees.

Even though there are differences between chimpanzees and humans they are still capable of at least imitating Sign Language which is what we want to model in software [20].

These differences will be explored Later and exploited if possible, the idea behind this is that simulating human vision known as computer vision (computer vision is the field in which computers can use vision to solve problems) is a large problem space, but if an animal with less intelligence and vision capabilities can understand sign language then that is possibly a smaller problem space.

Attacking this problem space or vision model might be considerably less then tackling the human vision problem.

### 1.2.9 Object Recognition

#### 1.2.9.1 Large Object Tracking

In some interactive applications, the computer needs to track the position or orientation of a hand that is prominent in the image. Relevant applications might be computer games, or interactive machine control. In such cases, a description of the overall properties of the image ,may be adequate. Image moments, which are fast to compute, provide a very coarse summary of global averages of orientation and position. If the hand is on a uniform background, this method can distinguish hand positions and simple pointing gestures [22].

The large-object-tracking method makes use of a low-cost detector/processor to quickly calculate moments. This is called the artificial retina chip. This chip combines image detection with some low-level image processing (named artificial retina by analogy with those combined abilities of the human retina). The chip can compute various functions useful in the fast algorithms for interactive graphics applications.

#### 1.2.9.2 Shape Recognition

Most applications, such as recognizing particular static hand signal, require a richer description of the shape of the input object than image moments provide.

If the hand signals fell in a predetermined set, and the camera views a close-up of the hand, we may use an example-based approach, combined with a simple method top analyze hand signals called orientation histograms. These example-based applications involve two phases, training and running. In the training phase, the user shows the system one or more examples of a specific hand shape [23].

The computer forms and stores the corresponding orientation histograms. In the run phase, the computer compares the orientation histogram of the current image with each of the stored templates and selects the category of the closest match, or interpolates between templates, as appropriate. This method should be robust against small differences in the size of the hand but probably would be sensitive to changes in hand orientation.

**Goals**

The scope of this project is to create a method to recognize hand gestures, based on a pattern recognition technique developed by McConnell; employing histograms of local orientation. The orientation histogram will be used as a feature vector for gesture classification and interpolation. High priority for the system is to be simple without making use of any special hardware.

All the computation should occur on a workstation or PC. Special hardware would be used only to digitize the image (scanner or digital camera).

**Summary**

In this chapter we explained of history of sign language. The term of the sign language and its importance, the problems that are faced in this kind of language, the areas that could be used in. In this chapter we also explained the kind of sign language and Structural components of signs. Also the natural language and its representation in sign, sign language in different communities, American and British sign language. Facial action coding system was explained and one of the most common story of learning sign language to the animal, finally object recognition.

# CHAPTER TWO

# ARTIFICIAL NEURAL NETWORKS

## 2.1 Overview

The idea of pattern recognition comes from real life. The human brain can memorize and recognize the patterns. The neural networks model the human brain.

Neural network (NN) algorithms for pattern recognition work by applying the input patterns after preprocessing to the back propagation neural network.

The basic concepts and the algorithms which are used in artificial neural networks will be presented in this chapter. Back propagation algorithm will be explained in detail since the algorithm will be used in the developed pattern recognition system[26].

## 2.2 Introduction to Artificial Neural Networks

An artificial neural network (ANN) is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing element or nodes. Neural network architecture is inspired by the architecture of biological nervous systems, which use many simple processing elements operating in parallel to obtain high computation rates[27].

An artificial neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

- Knowledge is acquired by the network through a learning process.
- Interneuron connection strengths known as synaptic weights are used to store the knowledge.

The neuron is a “many inputs one output” unit. The output can be excited or not excited, just two possible choices. The signals from other neurons are summed together and compared against a threshold to determine if the neuron shall excite. The input signals are subject to attenuation in the synapses which are junction parts of the neuron[28].

ANN draws much of their inspiration from the biological nervous system. It is therefore very useful to have some knowledge of the way this system is organized. Most living creatures, which have the ability to adapt to a changing environment, need a controlling unit which is able to learn. Higher developed animals and humans use very complex networks of highly specialized neurons to perform this task. The control unit – the brain - can be divided in different anatomic and functional sub-units, each having certain tasks like vision, hearing, motor and sensor control[29].

The brain is connected by nerves to the sensors and actors in the rest of the body. The brain consists of a very large number of neurons, about $10^{11}$ in average. These can be seen as the basic building bricks for the central nervous system. The neurons are interconnected at points called synapses. The complexity of the brain is due to the massive number of highly interconnected simple units working in parallel, with an individual neuron receiving input from up to 10000 others [26].

Structurally the neuron can be divided in three major parts: the cell body (soma), the dendrites, and the axon. The cell body contains the organelles of the neuron and also the "dendrites" are originating there. These are thin and widely branching fibers, reaching out in different directions to make connections to a larger number of cells within the cluster. Input connections are made from the axons of other cells to the dendrites or directly to the body of the cell. These are known as axondentrititic and axonsomatic synapses[29].

There is only one axon per neuron. It is a single and long fiber, which transports the output signal of the cell as electrical impulses (action potential) along its length. The end of the axon may divide in many branches, which are then connected to other cells. The branches have the function to fan out the signal to many other inputs [28].

A single-input neuron artificial network is shown in Figure 2.1. The scalar input p is multiplied by the scalar weight w to form wp, one of the terms that is sent to the summer. The other input, 1, is multiplied by a bias b and then passed to the summer. The summer output net, often referred to network input, goes into an activation function *f*, which produces the scalar neuron output. This is the simplest form of the artificial neuron and is known as a perception [30].

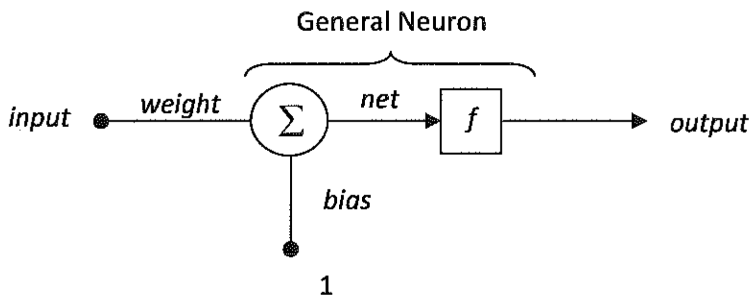



Figure 2.1 Single - Input Artificial Neuron

The neuron output is calculated by equation 2.1:

$$output = f(wp + b) \tag{2.1}$$

The simple model for artificial neuron in the Figure 3.1 can indicate the same way of the biological neuron. The weight w corresponds to the strength of a synapse, the cell body is represented by the summation and the activation function, and the neuron output represents the signal in the axon [31].

## 2.3 Teaching an Artificial Neural Network

Artificial neural networks learning algorithms can be divided into two main groups that are supervised (Associative learning) and unsupervised (Self-Organization).

### 2.3.1 Supervised Learning

The vast majority of artificial neural network solutions have been trained with supervision. In this mode, (In the former) the actual output of a neural network is compared to the desired output. Weights, which are usually randomly set to begin with, are then adjusted by the network so that the next iteration, or cycle, will produce a closer match between the desired and the actual output. The learning method tries to minimize the current errors of all processing elements. This global error reduction is created over time by continuously modifying the input weights until acceptable network accuracy is reached[32].

The supervised artificial neural network needs teacher to describe what the network should have given as response. The difference between target (desired output) and the actual output, the error is determined and back propagated through the network to adjust the network. The basic architecture of supervised artificial neural network is shown in Figure 2.2.

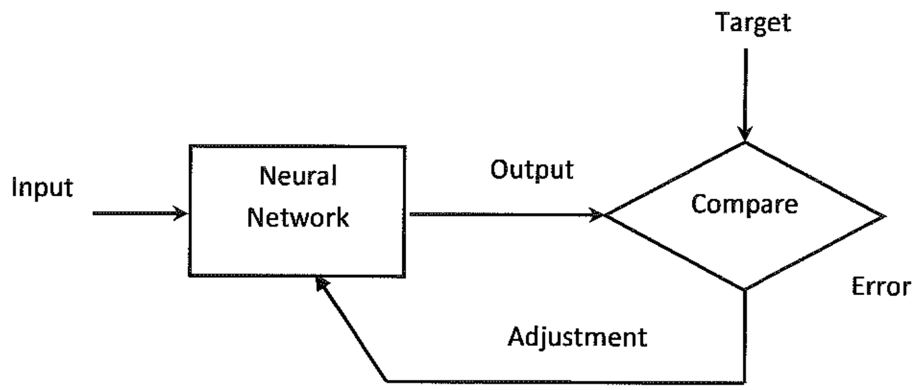



Figure 2.2 Architecture of Supervised Artificial Neural Network

With supervised learning, the artificial neural network must be trained before it becomes useful. Training consists of presenting input and output data to the network. This data is often referred to as the training set. That is, for each input set provided to the system, the corresponding desired output set is provided as well. In most applications, actual data must be used. This training phase can consume a lot of time. In prototype systems, with inadequate processing power, learning can take weeks. This training is considered complete when the neural network reaches a user defined performance level. This level signifies that the network has achieved the desired statistical accuracy as it produces the required outputs for a given sequence of inputs. When no further learning is necessary, the weights are typically frozen for the application. Some network types allow continual training, at a much slower rate, while in operation. This helps a network to adapt to gradually changing conditions[33].

Training sets need to be fairly large to contain all the needed information if the network is to learn the features and relationships that are important. Not only do the sets have to be large but the training sessions must include a wide variety of data. If the network is trained just one example at a time, all the weights set so meticulously for one fact could be drastically altered in learning the next fact. The previous facts could be forgotten in learning something new. As a result, the system has to learn everything together, finding the best weight settings for the total set of facts[26].

For example, in teaching a system to recognize pixel patterns for the ten digits, if there were twenty examples of each digit, all the examples of the digit seven should not be presented at the same time[34].

How the input and output data is represented, or encoded, is a major component to successfully instructing a network. Artificial networks only deal with numeric input data. Therefore, the raw data must often be converted from the external environment. Additionally, it is usually necessary to scale the data, or normalize it to the network's paradigm. This pre-processing of real-world stimuli, be they cameras or sensors, into machine readable format is already common for standard computers. Many conditioning techniques which directly apply to artificial neural network implementations are readily available. It is then up to the network designer to find the best data format and matching network architecture for a given application[35].

After a supervised network performs well on the training data, then it is important to see what it can do with data it has not seen before. If a system does not give reasonable outputs for this test set, the training period is not over. Indeed, this testing is critical to insure that the network has not simply memorized a given set of data but has learned the general patterns involved within an application[34].

One of the most commonly used supervised neural network model is back propagation network that uses back propagation learning algorithm. Back propagation algorithm is one of the well-known algorithms in neural networks [36].

### 2.3.2 Unsupervised Learning

Unsupervised learning is the great promise of the future. Currently, (the latter) this learning method ,(the latter)is limited to networks known as self-organizing maps. These kinds of networks are not in widespread use. They are basically an academic novelty. Yet, they have shown they can provide a solution in a few instances, proving that their promise is not groundless. They have been proven to be more effective than many algorithmic techniques for numerical aerodynamic flow calculations. They are also being used in the lab where they are split into a front-end network that recognizes short, phoneme-like fragments of speech which are then passed on to a back-end network. The second artificial network recognizes these strings of fragments as words[37].

For an unsupervised learning rule, the training set consists of input training patterns only. Therefore, the network is trained without benefit of any teacher. The network learns to adapt based on the experiences collected through the previous training patterns. The basic architecture of an unsupervised system is shown in Figure 2.3.

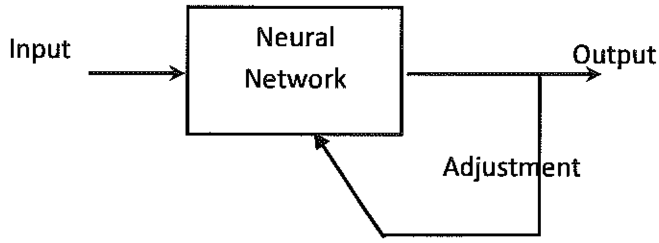



Figure 2.3 Architecture of Unsupervised Artificial Neural Network

This promising field of unsupervised learning is sometimes called self-supervised learning. These networks use no external influences to adjust their weights. Instead, they internally monitor their performance. These networks look for regularities or trends in the input signals, and makes adaptations according to the function of the network. Even without being told whether it's right or wrong, the network still must have some information about how to organize itself. This information is built into the network topology and learning rules. An unsupervised learning algorithm might emphasize cooperation among clusters of processing elements. In such a scheme, the clusters would work together. If some external input activated any node in the cluster, the cluster's activity as a whole could be increased. Likewise, if external input to nodes in the cluster was decreased, that could have an inhibitory effect on the entire cluster[38].

Competition between processing elements could also form a basis for learning. Training of competitive clusters could amplify the responses of specific groups to specific stimuli. As such, it would associate those groups with each other and with a specific appropriate response. Normally, when competition for learning is in effect, only the weights belonging to the winning processing element will be updated [34].

### 2.3.3 Learning Laws

Many learning laws are in common use. Most of these laws are some sort of variation of the best known and oldest learning law, Hebb's Rule. Research into different learning functions continues as new ideas routinely show up in trade publications. Some researchers have the modeling of biological learning as their main objective. Others are experimenting with adaptations of their perceptions of how nature handles learning. Either way, man's understanding of how neural processing actually works is very limited. Learning is certainly more complex than the simplifications represented by the learning laws currently developed. A few of the major laws are presented as examples [39].

**Hebb's Rule:**

The first, and undoubtedly the best known, learning rule were introduced by Donald Hebb. The description appeared in his book The Organization of Behavior in 1949. His basic rule is: If a neuron receives an input from another neuron and if both are highly active (mathematically have the same sign), the weight between the neurons should be strengthened [40].

**Hopfield Law:**

It is similar to Hebb's rule with the exception that it specifies the magnitude of the strengthening or weakening. It states, "if the desired output and the input are both active or both inactive, increment the connection weight by the learning rate, otherwise decrement the weight by the learning rate [41].

**The Delta Rule:**

This rule is a further variation of Hebb's Rule. It is one of the most commonly used. This rule is based on the simple idea of continuously modifying the strengths of the input connections to reduce the difference (the delta) between the desired output value and the actual output of a processing element. This rule changes the synaptic weights in the way that minimizes the mean squared error of the network. This rule is also referred to as the Widrow-Hoff Learning Rule and the Least Mean Square (LMS) Learning Rule. The way that the Delta Rule works is that the delta error in the output layer is transformed by the derivative of the transfer function and is then used in the previous neural layer to adjust input connection weights. In other words, this error is back-propagated into previous layers one layer at a time[42]. The process of back-propagating the network errors continues until the first layer is reached. The network type called Feedforward, Back-propagation derives its name from this method of computing the error term. When using the delta rule, it is important to ensure that the input data set is well randomized. Well ordered or structured presentation of the training set can lead to a network which cannot converge to the desired accuracy. If that happens, then the network is incapable of learning the problem [40].

**The Gradient Descent Rule:**

This rule is similar to the Delta Rule in that the derivative of the transfer function is still used to modify the delta error before it is applied to the connection weights. Here, however, an additional proportional constant tied to the learning rate is appended to the final modifying factor acting upon the weight. This rule is commonly used, even though it converges to a point of stability very slowly. It has been shown that different learning rates for different layers of a network help the learning process converge faster. In these tests, the learning rates for those layers close to the output were set lower than those layers near the input [43].

**Kohonen's Learning Law:**

This procedure, developed by Teuvo Kohonen, was inspired by learning in biological systems. In this procedure, the processing elements compete for the opportunity to learn, or update their weights. The processing element with the largest output is declared the winner and has the capability of inhibiting its competitors as well as exciting its neighbors. Only the winner is permitted an output, and only the winner plus its neighbors are allowed to adjust their connection weights[44].

Further, the size of the neighborhood can vary during the training period. The usual paradigm is to start with a larger definition of the neighborhood, and narrow in as the training process proceeds. Because the winning element is defined as the one that has the closest match to the input pattern, Kohonen networks model the distribution of the inputs. This is good for statistical or topological modeling of the data and is sometimes referred to as self-organizing maps or self-organizing topologies [45].

## 2.4 Multilayer Perceptron

The multilayer perceptron (MLP) is a hierarchical structure of several perceptrons. A single perceptron is not very useful because of its limited mapping ability. No matter what activation function is used, the perceptron is only able to represent an oriented ridge-like function. The perceptrons can, however, be used as building blocks of a larger, much more practical structure. A typical multilayer perceptron (MLP) network consists of a set of source nodes forming the input layer, one or more hidden layers of computation nodes, and an output layer of nodes. The input signal propagates through the network layer-by-layer [46]. The signal-flow of such a network with one hidden layer is shown in Figure 2.4.

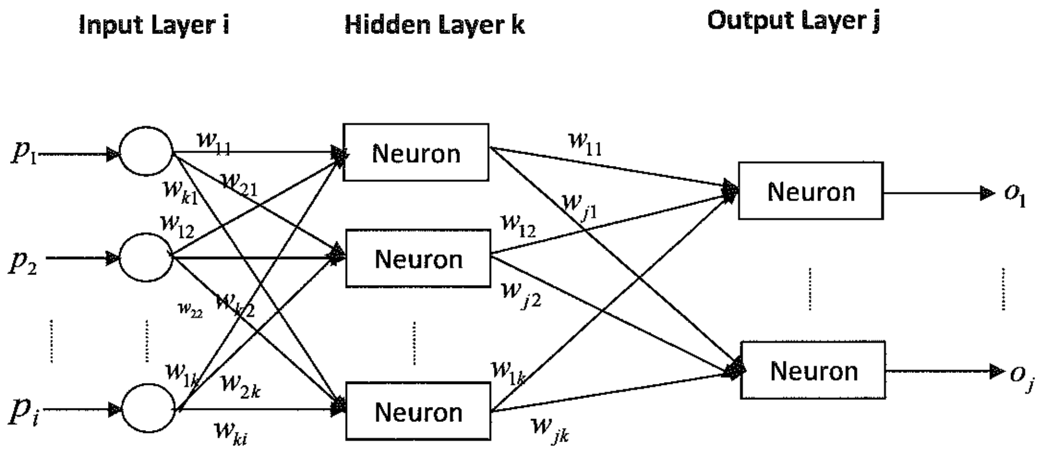



Figure 2.4 Architecture of Multilayer Perceptron

The supervised learning problem of the multilayer perceptron can be solved with the back-propagation algorithm. The algorithm consists of two steps. In the forward pass, the predicted outputs corresponding to the given inputs. In the backward pass, partial derivatives of the cost function with respect to the different parameters are propagated back through the network. The chain rule of differentiation gives very similar computational rules for the backward pass as the ones in the forward pass. The network weights can then be adapted using any gradient-based optimization algorithm. The whole process is iterated until the weights have converged. The multilayer perceptron network can also be used for unsupervised learning by using the so called auto-associative structure. This is done by setting the same values for both the inputs and the outputs of the network. The extracted sources emerge from the values of the hidden neurons. This approach is computationally rather intensive. The multilayer perceptron network has to have at least three hidden layers for any reasonable representation and training such a network is a time consuming process [47].

## 2.5 Back Propagation Neural Network

In the artificial neural networks, several network architectures and training algorithms are available, the back-propagation algorithm is the most popular algorithm. Back propagation neural network architecture is very popular because it can be applied to realize many different procedures [48].

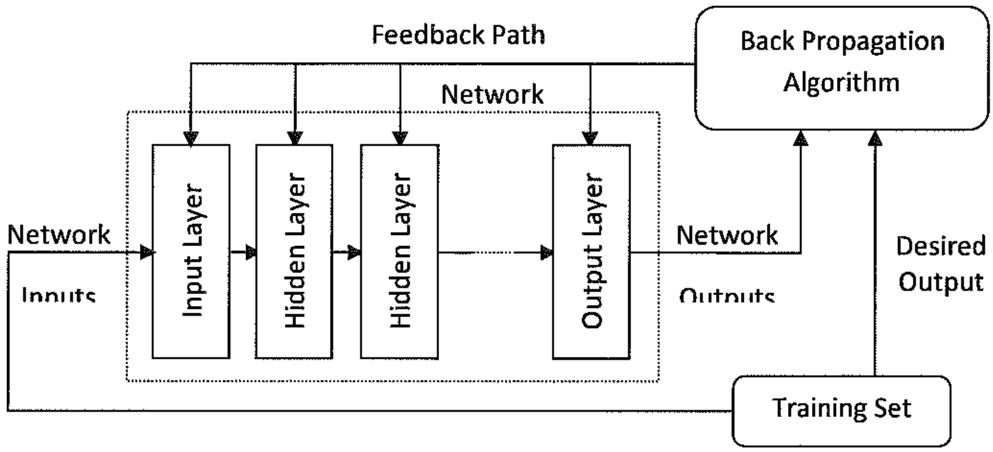



Figure 2.5 Block Diagram of Back Propagation Network

In Figure 2.5 the block diagram of back propagation network is shown. The back-propagation algorithm is a supervised learning algorithm for artificial neural networks. It extends the weight update rule used in the simple perceptron learning algorithm to multilayer feed forward artificial neural network[49].

The name "back-propagation" derives from the manner in which information is propagated across the network on each pass of the algorithm, the errors at the output nodes are passed back to the hidden nodes using the network connections, and the resulting information is used to update the connection weights [26,50].

### 2.5.1 Structure of Back propagation Network

Feed-forward neural networks trained by back propagation consist of several layers of simple processing elements called neurons, interconnections, and weights that are assigned to those interconnections. Each neuron contains the weighted sum of its inputs filtered by a sigmoid transfer function[48]. The neurons are interconnected in such a way that information relevant to the I/O mapping is stored in the weights. The various layers of neurons in back propagation networks receive, process, and transmit information on the relationships between the input parameters and corresponding responses. Aside from the input and output layers, these networks incorporate one or more hidden [51]. Architecture of back propagation network is shown in Figure 2.6.

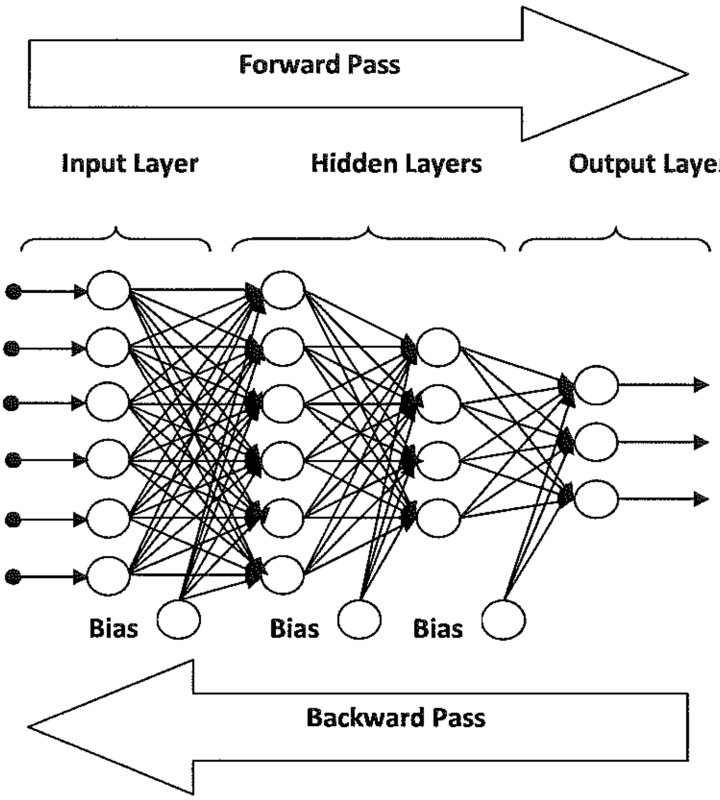



Figure 2.6 Back Propagation Network Architecture

### 2.5.2 Back Propagation Network Algorithm

In the back propagation learning algorithm, the network begins with a random set of weights. An input vector is fed forward through the network, and the output values are calculated using this initial weight set. Next, the calculated output is compared with the measured output data, and the squared difference between this pair of vectors determines the overall system error. The network attempts to minimize this error using the gradient descent approach, in which the network weights are adjusted in the direction of decreasing error[52].

The steps of back propagation algorithm can be listed as the following:

**Step 1**: Initialize hidden and output weights to small random values.

**Step 2**: Input training vector.

**Step 3**: Calculate outputs of hidden neurons.

**Step 4**: Calculate outputs of output neurons.

**Step 5**: Calculate the differences between the results of outputs of output neurons and targets.

**Step 6**: Back propagate the error to update the hidden and the output weights.

**Step 7**: Repeat the steps 3, 4, 5, and 6 until reaching the goal error.

**Step 8:** Upon conversion save hidden and output weights for use in feed forward calculations [53].

#### 2.5.2.1 Feed Forward Calculation

When a back propagation network is cycled, the activations of the input units are passed forward to the output layer through the connecting weights. The starting point for most neural networks is a model neuron, as in Figure 2.7.

This neuron consists of multiple inputs and a single output. Each input is modified by a weight, which multiplies with the input value. The neuron will combine these weighted inputs and, with reference to activation function determine its output [54].

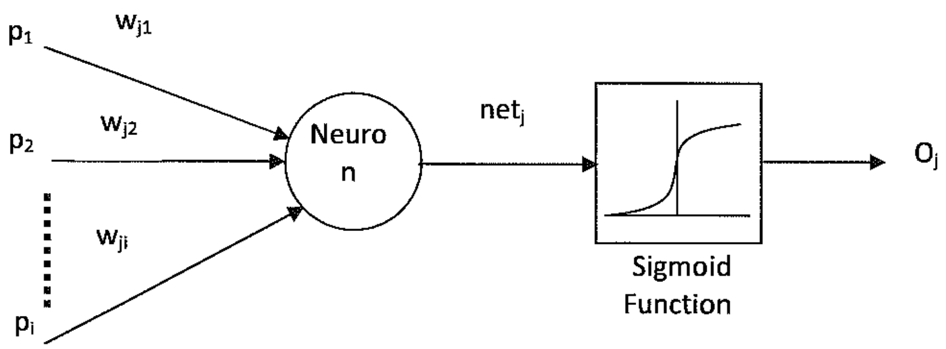



Figure 2.7 A model Neuron Structure

The main idea in feed forward calculation is passing inputs forward and all outputs are computed through sigmoid function. The output of each neuron is a function of its inputs [55].

In particular, the output of the *j*th neuron in any layer is described by two sets of equations 2.2 and 2.3:

$$net_j = \sum p_i w_{ji} \qquad (2.2)$$

$$O_j = f_{th}(net_j) \qquad (2.3)$$

For every neuron, j, in a layer, each of the i inputs, pj, to that layer is multiplied by a previously established weight, wij. These are all summed together, resulting in the internal value of this operation, netj. This value is then sent through an activation function, $f_{th}$.

The activation function is usually the sigmoid function, which has an input to output mapping as shown in Figure 2.8. The resulting output, $O_j$, is an input to the next layer or it is a response of the neural network if it is the last layer [56].

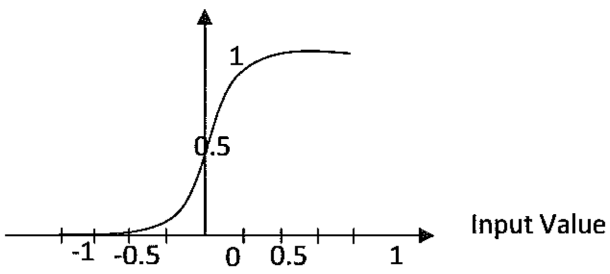



Figure 2.8 Sigmoid Activation Function

The output of the neuron with sigmoid activation function is given by equation 3.4:

$$O_j = f(net_j) = \frac{1}{(1+\exp(-net_j))} \tag{2.4}$$

The derivative of the sigmoid function can be obtained as follows equation 2.5:

$$\frac{\partial f(net_j)}{\partial net_j} = O_j * (1 - O_j) = f(net_j) * (1 - f(net_j)) \tag{2.5}$$

**2.5.2.2 Error Back Propagation Calculation**

The error back propagation calculations are applied only during the training of the neural network. The vital elements in these calculations are the error signal, learning rate, momentum factor, and weight adjustment [57].

- **Signal Error:**

During the network training, the feed forward output state calculation is combined with backward error propagation and weight adjustment calculations that represent the network's learning. Central to the concept of training a neural network is the definition of network error.

Rumelhart and McClelland define an error term that depends on the difference between the output values an output neuron is supposed to have, called the target value Tj, and the value it actually has as a result of the feed forward calculations, Oj. The error term represents a measure of how well a network is training on a particular training set [58].

A method called gradient descent is used to minimize the total error on the patterns in the training set. In gradient descent, weights are changed in proportion to the negative of an error derivative with respect to each weight given by equation 2.6:

$$\Delta w_{ji} = -\eta \left[ \frac{\partial E}{\partial w_{ji}} \right] \tag{2.6}$$

where $\eta$ is the learning rate and $E$ is the average over all training instances of the sum overall output neurons (total error).

Weights move in the direction of steepest descent on the error surface defined by the total error (summed across patterns) given by equation 2.7:

$$E = \sum_p \sum_j (T_{Pj} - O_{Pj})^2 \tag{2.7}$$

where $O_{pj}$ the actual output response to pattern p and $T_{pj}$ is the target output value.

Figure 2.9 illustrates the concept of gradient descent using a single weight. After the error on each pattern is computed, each weight is adjusted in proportion to the calculated error gradient back propagated from the outputs to the inputs. The changes in the weights reduce the overall error in the network [57].

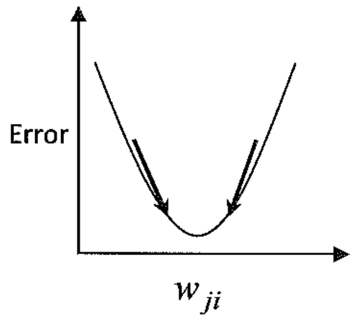



Figure 2.9 Typical Curve between Overall Error and A single Weight

The aim of the training process is to minimize this error over all training patterns. From equation 2.3, it can be seen that the output of a neuron in the output layer is a function of its input, or $O_j = f_{th}(net_j)$. The first derivative of this function $O_j = f'_{th}(net_j)$ is an important element in error back propagation. For output layer neurons, a quantity called the error signal is represented by $\Delta_{pj}$ which is defined in equation 2.8:

$$\Delta_{pj} = f'_{th}(net_{pj}) * (T_{pj} - O_{pj}) = (T_{pj} - O_{pj}) * O_{pj} * (1 - O_{pj}) \qquad (2.8)$$

This error value is propagated back and appropriate weight adjustments are performed. This is done by accumulating the $\Delta$'s for each neuron for the entire training set, add them, and propagate back the error based on the grand total $\Delta$. This is called batch (epoch) training [58].

- **Learning Rate and Momentum Factor**

There are two essential parameters that do affect the learning capability of the neural network. First the learning rate coefficient $\eta$ which defines the learning power of a neural network. Second the momentum factor $\alpha$ which defines the speed at which the neural network learns. This can be adjusted to a certain value in order to prevent the neural network from getting caught in what is called local energy minima. Both rates can have a value between 0 and 1[59].

The larger the learning rate $\eta$ the larger the weight changes on each epoch, and the quicker the network learns. However, the size of the learning rate can also influence whether the network achieves a stable solution. If the learning rate gets too large, then the weight changes no longer approximate a gradient descent procedure. Oscillation of the weights is often the result.

The ideal case is using the largest learning rate possible without triggering oscillation. This would offer the most rapid learning and the least amount of time spent waiting at the computer for the network to train. One method that has been proposed is a slight modification of the back propagation algorithm so that it includes a momentum term [60].

- **Weight Adjustment**

Each weight has to be set to an initial value. Random initialization is usually performed. Weight adjustment is performed in stages, starting at the end of the feed forward phase, and going backward to the inputs of the hidden layer.

The weights that feed the output layer and the hidden layer are updated using equation 2.9. This also includes the bias weights at the output layer neurons. However, in order to avoid the risk of the neural network getting caught in local minima[61].

The momentum term can be added as in equation 2.10.

$$w_{ji}(n+1) = w_{ji}(n) - \eta \Delta_{pj} O_{pi}^{T} \tag{2.9}$$

$$w_{ji}(n+1) = w_{ji}(n) - (1-\alpha)\eta \Delta_{pj} O_{pi}^{T} + \alpha[\delta w_{ji}(n)] \tag{2.10}$$

where the subscript n is the learning epoch and $\delta w_{ji}(n)$ stands for the previous weight change. The bias weights at the output and hidden layer neurons are updated, similarly .

### 2.5.3 Discussion Some Important Issues

There are some issues, which may cause some problem in neural networks, if totally ignored. For example input normalization before feeding data into a neural network is crucial. Moreover, appropriate weights initialization is needed. Also the other issues are very important in neural networks like training conversion criteria, various techniques/problems and generalization.

#### 2.5.3.1 Input Normalization and Weights Initialization

The contribution of an input will depend heavily on its variability relative to other inputs. If for example one of the inputs has range of 0 to 1 and another has a range of 0 to 1000, then the contribution of the first input will be swamped by the second input. So it is essential to rescale the inputs so their variability reflects their importance. For lack of any prior information (regarding the importance of each input), it is common to normalize each input to the same range or the same standard deviation [62].

Typically inputs are normalized to same small ranges, like [0,1] or [-1,1]. In particular any scaling that gathers input values around zero works better. So instead of a [-1,1] scale, it might be preferable to normalize the inputs so as to have mean value of 0 and standard deviation of 1.

Weights initialization follows nearly the same path as input normalization. The main emphasis in the neural network literature on initial values has been on the avoidance of saturation, hence the desire to use small random values. Symmetry breaking in the weight space is needed in order to make neurons compute different functions. If all nodes have identical weights then they would respond identically. Therefore the gradient, which updates the weights, would be the same for each neuron. This way the weights would remain identical even after the update and this means no learning. A special case is to initialize all weights of every neuron to 0. Then in every neuron the gradient of a zero function would be zero and thus weights would remain zero until training is terminated[63].

Small weights (as well as small inputs) are needed to avoid immediate saturation because large weights could amplify a moderate input to produce an extremely large weighted sum at the inputs of the next layer. This would put the nodes into the flat regions of their nonlinearities and learning would be very slow because of the very small derivatives [62].

#### 2.5.3.2 Training Conversion Criteria

Stopping of training when the back propagation network is trained has to be known. Since various "learning rate - momentum factor- number hidden neurons"-schemes are being tested to adapt the stopping criteria to each case in the network to get a good efficient learning[64].

Four basic termination conditions when training an artificial neural network:

- Fixed number of iterations: Iterations, also called epochs, refer to the number of times the total training set is being presented in the neural network.
- Use threshold for the error: Empirically estimate a certain value for the error, which considered being acceptable.
- Early stopping: Divide the available data into training and validation sets. Commonly use a large number of hidden units and very small initial values. Compute the validation error rate periodically during training. Finally, stop training when the validation errors rate "start to go up". However, it is important to stress that the validation error is not a good estimate of the generalization error. The most common method for getting an unbiased

estimate of the generalization error is to run the ANN on a third set of data, that is not used at all during the training process.

#### 2.5.3.3 Techniques and Arising Problems

Multilayer Neural Networks have error surfaces with multiple local minima. The complexity of these surfaces increases as the number of weights (and so neurons) increases. Therefore, there is only one deepest global minimum among many shallow or deep local minimums. This means that the training procedure might get trapped into the latter small minima. In fact this is the case but there are two perspectives in relative bibliography that try to explain why artificial neural networks are still so much efficient and powerful tool.

- Many weights' means that error surfaces exist in high multidimensional spaces (one dimension for each weight). Someone would say that during back propagation one of the weights might fall in local minimum. But, other weights would not! Intuitively, the more the weights, the more dimensions exist, which provide "escape roots" from local minimums[65].
- Another perspective is the one, based on which sigmoid function behaves as linear when the weights are close to zero. This is the case during the first iterations of the neural network training. So in first steps the network simulates a smooth function. By the time the weights are "heavily" updated and the simulated function has much more complex error surface.

Back propagation's main problem is that it is sensitive to the so-called overfitting of the training data at the cost of decreasing generalization accuracy over other unseen examples. It is said that when the overfitting case is faced the artificial neural network adopts the idiosyncrasies of the training data. This means that the performance over unseen examples decreases. Especially when the training set is not representative of the general distribution of all possible examples, the performance drops dramatically. In order to avoid over fitting, caused by the repetitive feed of the same group of training examples onto the artificial neural networks, the early stopping technique is used[66].

It is necessary to remind that in early stopping the total number of iterations of the training procedure is such that produces the lowest error over the validation set, since this is the best indicator over unseen examples. In other words, the number of iterations that yields the best performance over the validation set is needed. Another, potentially useful technique is called weight decay or commonly regularization. This way, weights are kept small and the error surface smooth.

#### 2.5.3.4 Generalization

Generalization is the ability of capturing the underlying function, during the training phase, and hence producing correct outputs in response to novel patterns (patterns that has not seen before). A system then is said to generalize well. If performance in new patterns is poor then poor is the generalization as well.

Minimizing the generalization error is not equivalent to selecting a model where the bias is zero. This is because the model variance penalty may be too high. This is called the bias/variance trade-off. Variance and bias are well-understood issues when it comes to regression problems (function approximation using Neural Networks). However, in classification there is a correspondence but it is surely more complex subject [67].

There are a few conditions that are typically necessary–although not sufficient for good generalization:

- In order to generalize well, a system needs to be sufficiently powerful to approximate the target function. If it is too simple to fit even the training data then generalization to new data is also likely to be poor.
- The inputs contain sufficient information pertaining to the target, so that really existence a concept (unknown and complex mathematical function) that relates inputs with corrects outputs.
- In general, the training set must be a representative subset of the theoretical population. A poor set of training data may contain misleading regularities not found in the underlying function/classifier[68].

## 2.6 Summary

This chapter presented a general overview of artificial neural networks (ANN). The back propagation algorithm was also presented in detail since the algorithm is to be used in our pattern recognition system as a classifier.

This chapter gave a background to understand the developed method for pattern recognition system which will be explained in the next chapter.

# CHAPTER THREE
# SIGN LANGUAGE RECOGNITION SYSTEM

## 3.1 Overview

An automatic system for the feature based sign language recognition must deal with two basic problems: extraction of the essential features of images and try to recognition this images.

This chapter presents discussion of thesis topic about sign language recognition using back propagation neural networks. The method describes sign recognition using back propagation neural networks approach implementation on different sign images (number from zero to nine) by take 4 images for training and 4 images for testing with added noise and rotations on some of this images.

This chapter explains the research methodology and how has the process of collecting the database and how it was preprocessing. How it was organized to be feeding to the neural network system. How has the process of training of the neural network system. Testing process also explained in this chapter and the system designed to identify dumb and deaf language through the hand language images using artificial neural networks approach.

It is useful to have a machine that can carry out to pattern recognition process quickly and with high perform. For example, the security services that rely on fingerprints as an entry permit, if they depend on humans for the process of checking the fingerprints this would take long time If there is an application that can do this process quickly and with high accuracy, it will be saving money and time, as well as there are many real life applications that occur repeatedly and significantly such as reading checks in banks, where the existence of such an application eliminates the requirement that a human skills in such a repetitive task.

A neural network system for the image identification depends on shape of the gesture images must deals with two basic problems detection of the features of this images and extraction of the essential features of this images.

## 3.2 The Sign Languages System

The hand languages it is the one way to communication between the dumb and deaf people and with the other people, it is almost complete like our language, therefore this system try to recognition the numbers between (0 - 9). In the future work try to make this system recognize the characters and the some gestures.

A Neural Network (NN) is to be designed and trained to recognize Gestures through their signal language images.

The hand languages system simply involves the following series of steps:

1. Collect the signal or gesture images to build the database.
2. Preprocessing.
3. Training (ANN) using the database.
4. Test (ANN) using the database.
5. Results.

### 3.2.1 Database Collection

The starting point of working on the system was the collection of database with all images that would be used for training and testing the system. The images that are used in this work are .jpg format.

Using adobe Photoshop software tool to resize the signal images from the Different size to 45*45 pixels for each image, and added noise (Camouflage, Gaussian Blur, Crystal Noises and Smart Blur) and made rotation for some images in the 2nd experiment.

The database that was used in this work is made up of signal language images which are obtained from the internet[69,70]. The size of the images the size images are changed to (45x45) pixels.

The original database contains 10 images for numbers signal (0-9) for each gesture image 4 images for training and 4 images for testing for both experiments. Figures 3.1, 3.2 and 3.3 show an example of the original gesture images and the database images.

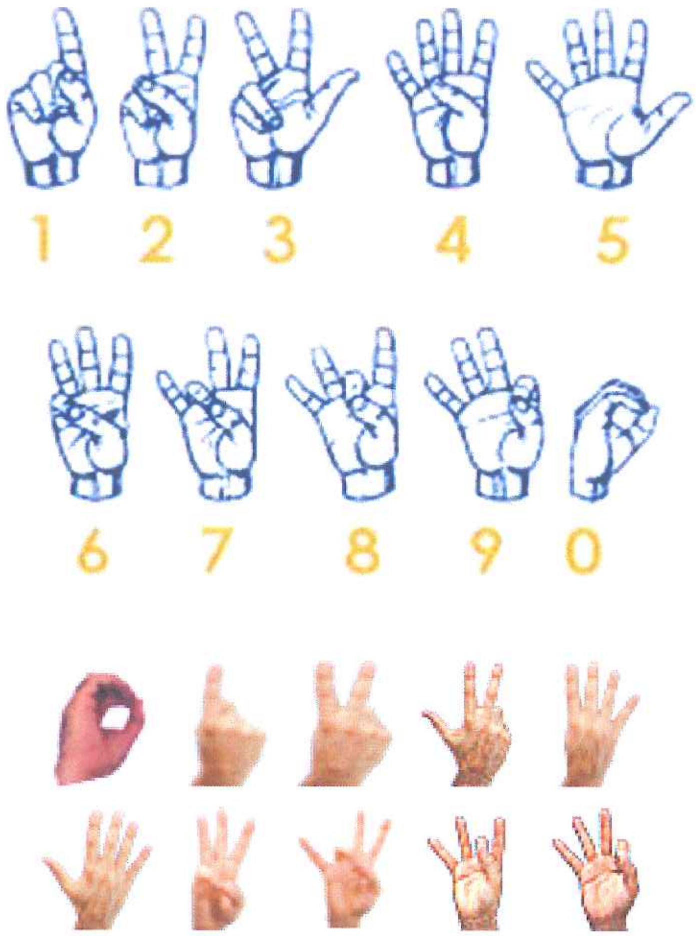



Figure 3.1 Example of original gesture Images.

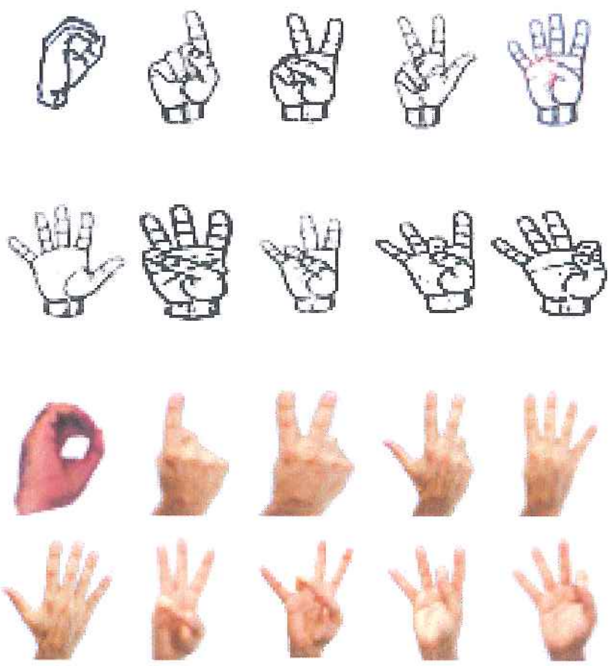


Figure 3.2 Example of database training Images.

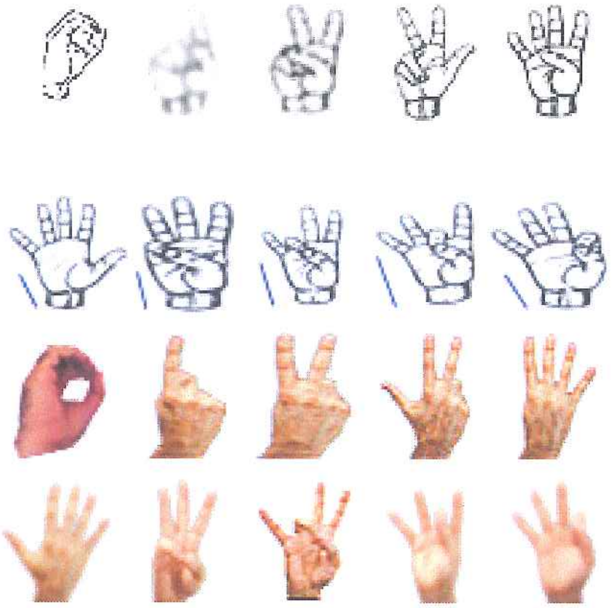


Figure 3.3 Example of database testing Images.

We can quote the difference between images of the training and test images in the above.

### 3.2.2 Preprocessing

The preprocessing stage in gesture image recognition system is a very important stage.

There are many operations in this stage:

1- Crop gesture images because they are big pictures.

2- Change the images size to stander size (45*45) pixels.

This operations are done by use adobe Photoshop program.

A digital text image that is containing dumb and deaf signals are generally an RGB image. The figures below showing two types of image containing digital dumb and deaf signal.

**RGB to Grayscale Image conversion**

In the pre-processing 1st stage is to converting the input RGB image into gray scale image. Here we considering the Othu's algorithm for RGB to gray scale conversion. The figure 3.4 and 3.5 below are show an RGB image and grayscale converted image.

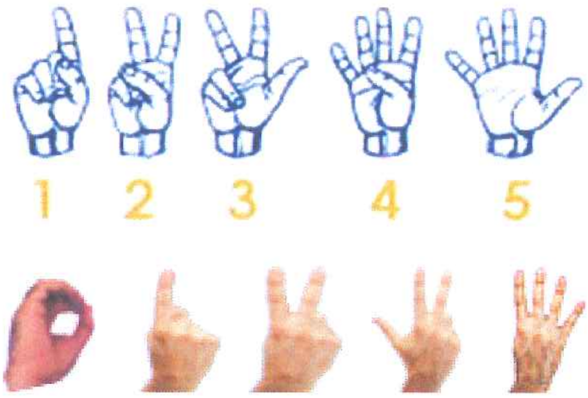



Figure 3.4 explain the RGB image

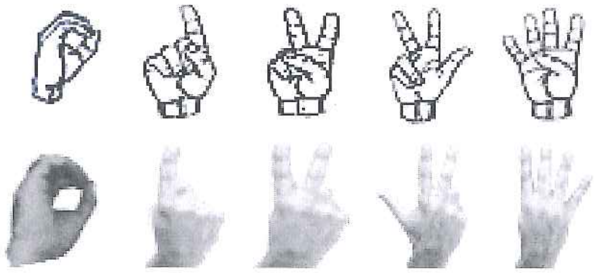


Figure 3.5 explain the Grayscale image

**Grayscale to Binary Image conversion**

In the pre-processing 2nd stage, converting the gray scale image into binary image. In a grayscale image there are 256 pixels of monochrome where 0 means pure black and 255 means pure white. This image is converted to binary image by checking whether or not each pixel value is greater than 255 level, (level, found by Otsu's Method)[71]. If the pixel value is greater than or equal to 255 level then the value is set to 1 i.e. white otherwise 0 i.e. black. The figure 3.6 is showing a Black and Wight images.

Figure 3.6 Explain the Black and Wight Images.

## 3.3 Image Reading

A digital image is a numeric representation (normally binary) of a two-dimensional image. Depending on whether the image resolution is fixed, it may be of vector or raster type. Without qualifications, the term "digital image" usually refers to raster images also called bitmap images[72].

Gray scale image is the image we will mostly work in this system. It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The first way it name Double Class or (Data Type), this assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The second way is called (UNIT 8) which assigns an integer between 0 and 255 to represent the brightness of a pixel. The value 0 corresponds to black and 255 to white. The class uint8 only requires roughly 1/8 of the storage compared to the class double. On the other hand, many mathematical functions can only be applied in the double class[73].

Through what previously mentioned above the images that used in order to train and test the signal language system are grayscale Double Class because the range of each pixel between (0..1).

### 3.3.1 Image Vectorization

A vector is defined by placing a sequence of numbers within square braces. Matlab is a software package that makes it easier to enter matrices and vectors, and manipulate them. To prepare the training set patterns to be fed in to the neural network must be arranged to input matrix, this process is called vectorization.

The dimension of the vector matrix for the 10 signals it will be different depend on the experiment that carried out and depending on the values and averaging and down sampling parameters that used in the experiment.

### 3.3.2 Using Matlab

Matlab is a simple and useful high-level language for matrix manipulation. Since images are matrices of numbers, many vision algorithms are naturally implemented in Matlab. It is often convenient to use Matlab even for programs for which this language is not the ideal choice in terms of data structures.

In fact, Matlab is an interpreted language, which makes program development very easy, and includes extensive tools for displaying matrices and functions, printing them into several different formats like Postscript, debugging, and creating graphical user interfaces. In addition, the Matlab package provides a huge amount of predefined functions. Matlab has an artificial neural networks toolbox, through this toolbox we can programming and design the system in easy way.

## 3.4 Implementation of the Neural Network

There are a few issue that have to be taken into consideration before the implementing the neural network. The following subtopics will be discussed.

### 3.4.1 Define BPNN Architecture and Design

Back propagation network architecture involves the selecting of an appropriate number of layers and the number of nodes in each layer based on the size and type of the application and the problem involved. As stated earlier, the neural network architecture consists of three layers, which are the input layer, hidden layer and also the output layer. The required number of nodes in each layer also differs from each dataset based on the classification problem. Thus, the number of nodes in the input and output layers are determined by the input and output variables based on the dataset.

Nevertheless, the essentials number of hidden nodes did not present. These hidden nodes are required for the computing difficult functions recognized as the non-separable functions. The number of nodes in the hidden layer determines the network's learning capabilities. It's been crucial of selecting the appropriate number of hidden layer for the optimal network design. The hidden layer size may affect the complexity and the required time for training but, out of all, it could influence its competence to generalize. However, there is no an appropriate standard rule or theory to determine the optimal number of hidden nodes.

### 3.4.2 Formulation of Weight Adjustment

The main focus in weight adjustment process is the activation function. The conventional sigmoid function will be applied to the standard back propagation.

### 3.4.3 Define the Learning Rate and Momentum Factor

The core parameters for neural networks are the learning rate and momentum factor, as these values will have an effect on the learning performance.

### 3.4.4 Define the Maximum Error

Maximum error is another parameter that should be taken into the consideration. This maximum error should be the stopping criteria for the back propagation training. As for this system, the maximum error is set different values. Besides that, the training process of the back propagation is been set to a maximum of 3000 iterations, or until the error reaches the maximum error. It is adequate for the network to train the dataset within 3000 iterations and converge to the solution. Since the main focus of this system is the faster convergence rate, thus the minimum iteration is important.

## 3.5 Training Using Back Propagation

An interesting aspect of Back Propagation Neural Networks (BPNN) in the Multi-Layer Perceptron (MLP) is that during the learning process, the hidden layers build an internal representation of the inputs that are useful to produce the output.

The adjustment of neural network parameters using back propagating of observed error at the network output is the most famous technique for supervised training of neural networks. Back Propagation depends on a sophisticated system of training contains a layers of neurons, begin with the first input layer which received the value according to their sequence in from the images vector matrix, followed by one or more of hidden layers of then the output layer.

The size of the input layer must be identical to the size of the image pixels and the size of output layer must be equal to the number of the signal's images that take to be trained in this case the output layer was equal to 10.

A series of experiments have been carried out with different number of raining and testing images and different values of the training parameters as shown in tables in the next chapter. The figure below represents the Neural Network Design.

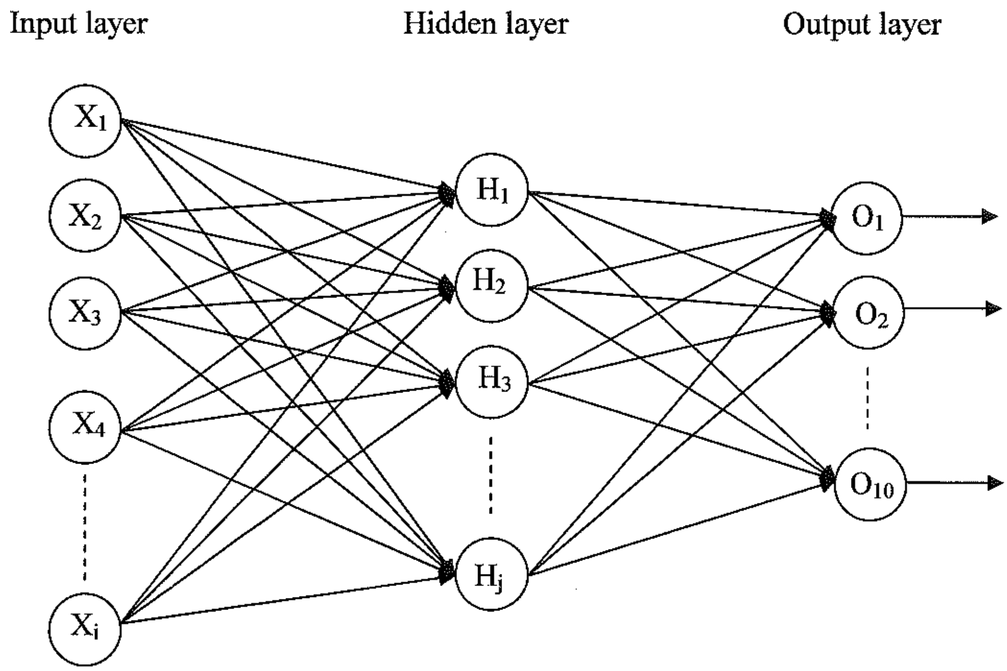



Figure 3.7 Neural Network Design.

Figure 4.8 shows the block diagram of the signal language recognition system, represented the three stage of the system. Starting with database collection, which is obtained from internet.

The second stage was to pre-processing these images to prepare it to the neural network stage. The preprocessing stage include adding noise to the images to extended the database images number and normalizing the pixel value.

The neural network stage involved training and testing processes. The training process was carried out using training set form database, the testing process was carried out using different images set equipped to the testing process.

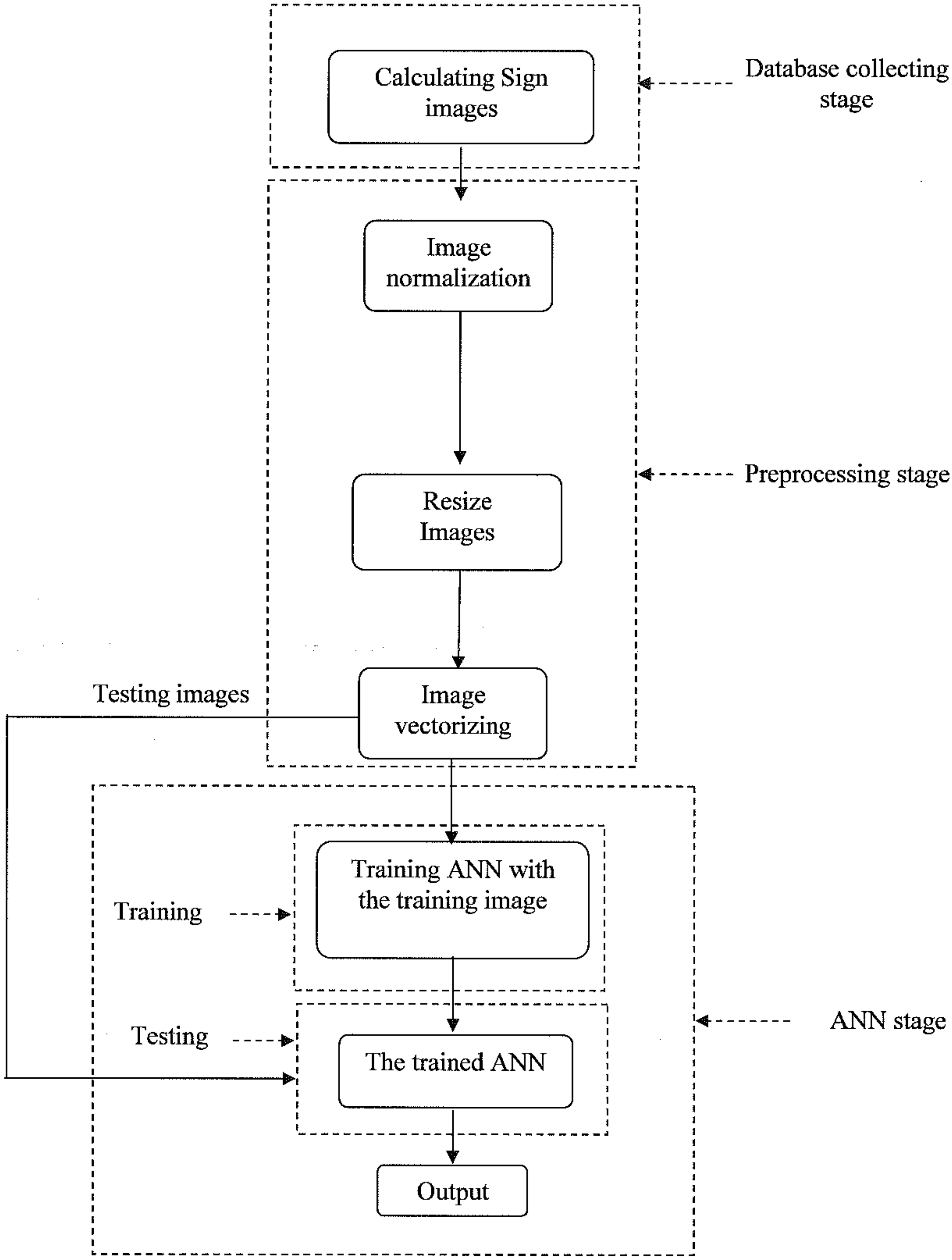



Figure 3.8 Block Diagram of SLR.

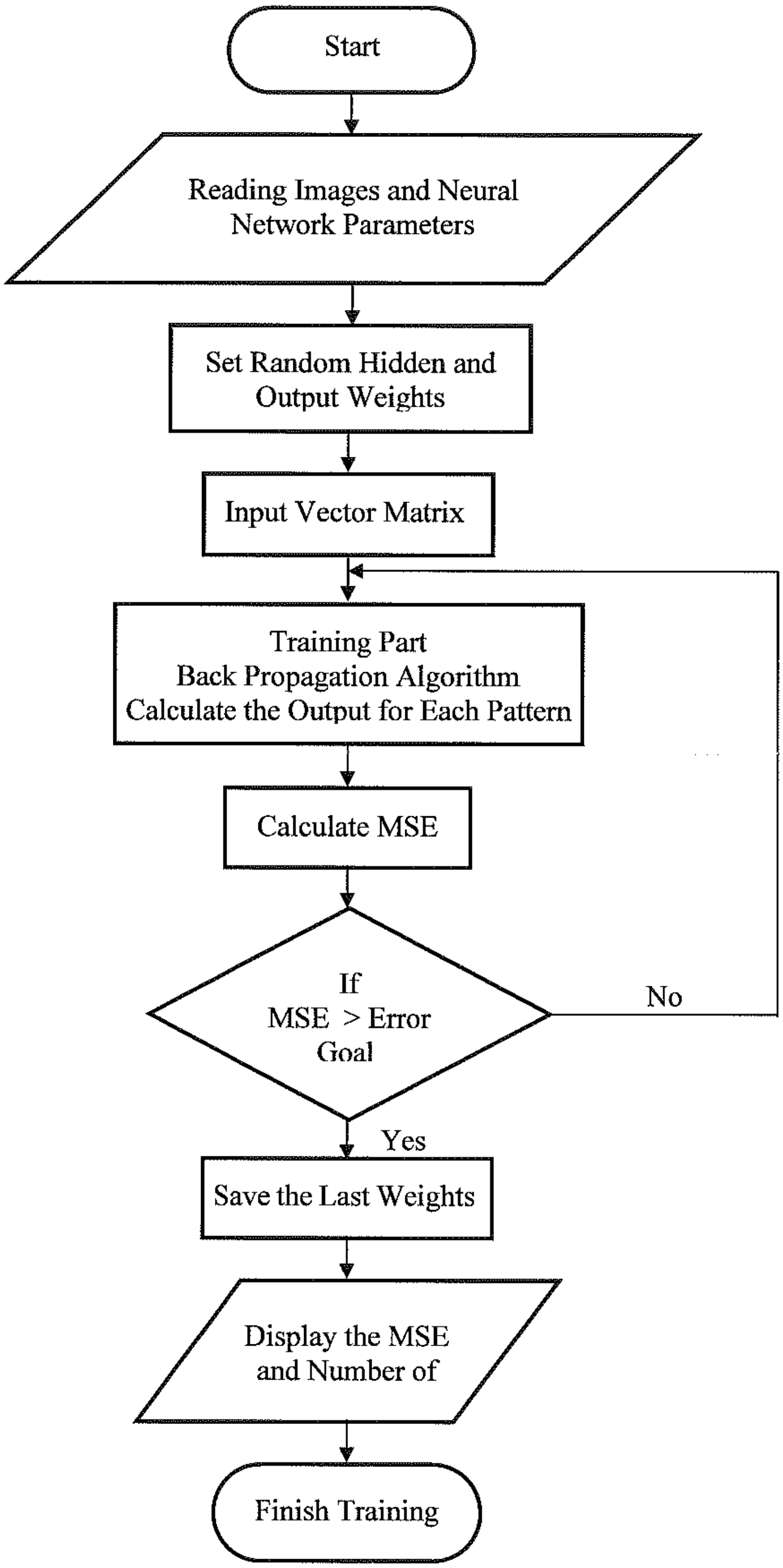



Figure 3.9 Flow Chart Diagram of the System (Training Process).

## 3.6 Testing SLR

After training is completed, in order to test the performance of the signal language recognition system, the testing process carried out by using a different set of signals images signals images of the same gestures that used their signals images for the training process, by adding some noise to the images to make it different from the original signal language images. Figure 3.10 shows an example of testing image.

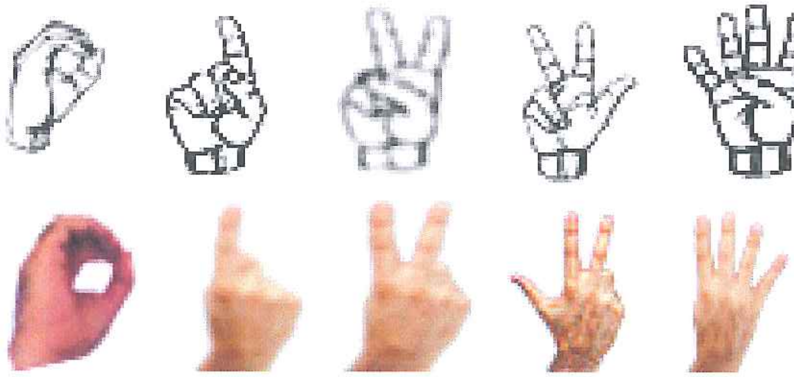


Figure 3.10 Example of Testing Image.

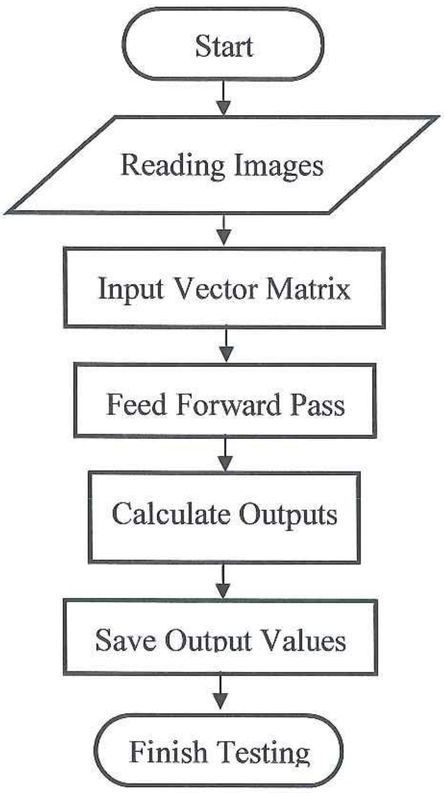



Figure 3.11 Flow Chart Diagram of the System (Testing Process).

## 3.7 Summary

This chapter discusses the methodology of the signal language recognition system in detail. Database collecting, normalized, resize the images. How it was organized to be feeding to the neural network system, and training and testing the system. Next chapter will contain the results that obtained from the training and testing this system.

# CHAPTER FOUR

# EXPERIMENTAL RESULTS AND DISCUSSION

## 4.1 Overview

This chapter presents the training and testing results of the neural network sign language and gesture recognition system. There are three experiments in this system. The first and second experiments used sketches, and the third experiment was used real images. The experiments were carried out for the training and testing stage. The results for each experiment are discussed in the next section.

## 4.2 General Experimental Setup

The comparison of the results was based on three criteria:

1. Experiment time cost
2. Number of epochs
3. Recognition performance

Many different experiments were carried out on each image. The number of training and testing images was four images for each signal for training and four different images for testing. The values of the training parameters are also different in each training process.

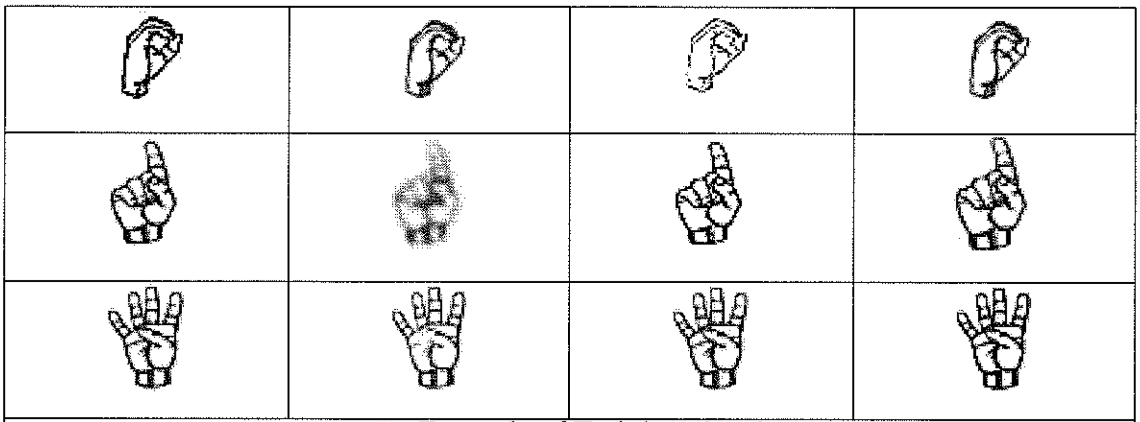
Training and testing the neural networks was implemented using the following system configuration: 2.2 GHz PC with 1 GB of RAM using Windows 7 32-bit operating system, and Matlab software tool.

The threshold value used to differentiate between the identified and not identified pattern was 60%.

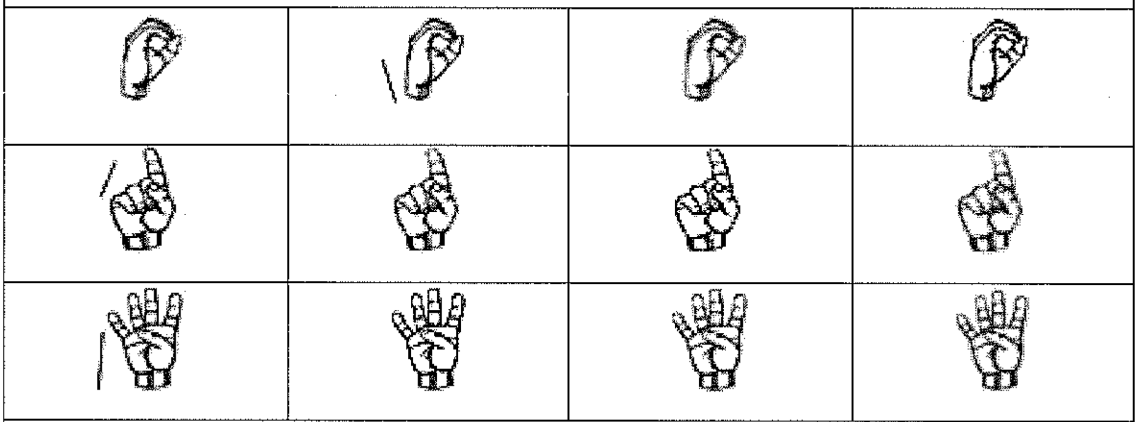
## 4.3 First Experiment

As shown in the tables below, the training and testing accuracy for the first experiment best than accuracy of second experiment that depends on the method of training the neural networks and the algorithm that used in the training phase and that depend on how images was coding and feed to the neural networks. In this experiment used sketches gesture, it represent images (element) of data base.

The number of sketches that used in the first experiment is four sketches of the training process and four sketches of the testing process. Figure 4.1 shows an example of training and testing sketches that used in first experiment. Appendix I shows the training and testing sketches.



a. An Example of Training sketches



b. An Example of Testing sketches

Figure 4.1 An Example of Training and Testing sketches for Experiment One[69,70]

Table 4.1 shows the training and testing time and the final training parameters during experiment one.

Table 4.1 Training Parameters and Training and Testing Time First Experiment

| | |
|---|---|
| Number of Input Neurons | 2025 |
| Number of Hidden Neurons | 70 |
| Number of Output Neurons | 10 |
| Weights Values Range | -0,35 and 0,35 |
| Learning Rate | 0.07 |

| | |
|---|---|
| Momentum Factor | 0.03 |
| Error | 0,00003 |
| Number Of Iteration | 1937 |
| Maximum Iteration | 3000 |
| Training Time | 126 Sec |
| Testing Time | 0.031 Sec |

Figure 4.2 shows the training performance curve of the artificial neural network system.

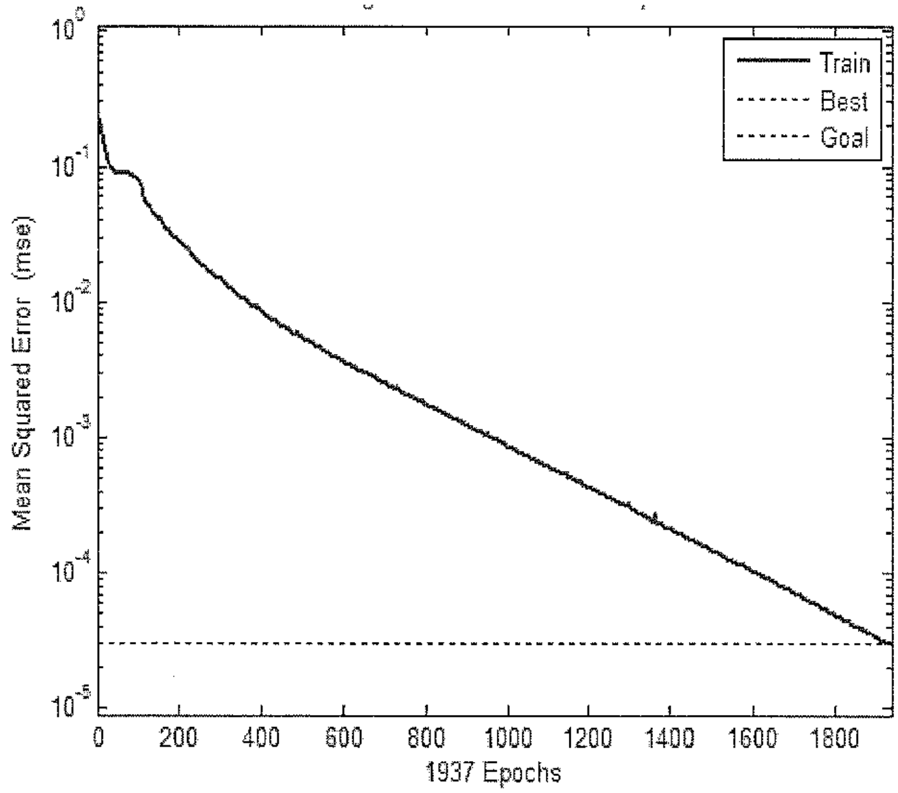



Figure 4.2 First Experiment training performance Curve

After training and testing process was finished, the recognition rate and accuracy of the training and testing of the 40 training sketches and 40 testing sketches for each sign sketches, the training and testing recognition rate and accuracy was calculated as shown in table 4.2 and 4.3.

Table 4.2 Training Recognition Rate and Accuracy for First Experiment

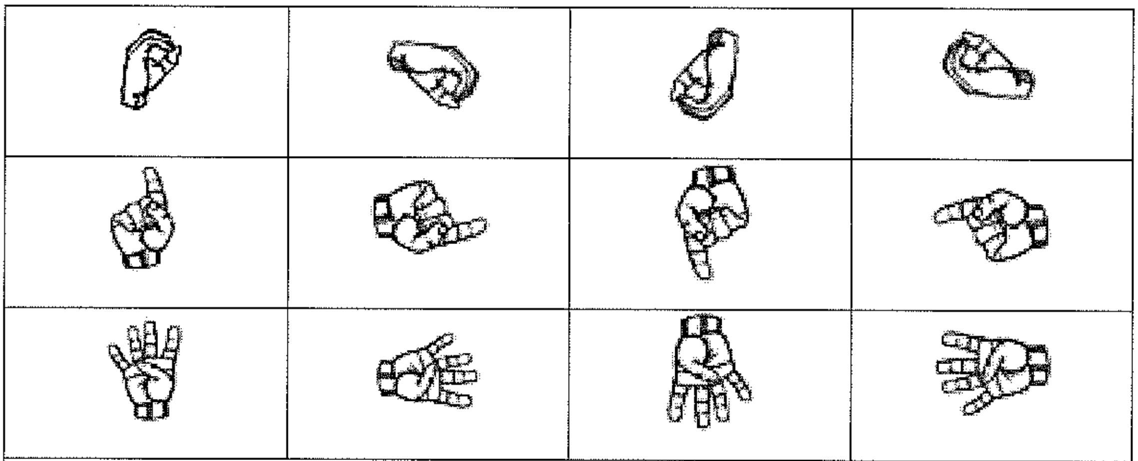
| Number | The Sign of Number | Accuracy |
|---|---|---|
| ZERO |  | **0.987825** |
| ONE |  | **0.987725** |
| TWO |  | **0.9882** |
| THREE |  | **0.9885** |
| FOUR |  | **0.987875** |
| FIVE |  | **0.990025** |
| SIX |  | **0.9878** |
| SEVEN |  | **0.98985** |
| EIGHT |  | **0.987725** |
| NINE |  | **0.9897** |
| **Total Accuracy** | | **0.988523** |
| **Neural Networks Recognition Rate** | | **(10/10) 100 %** |

Table 4.3 Testing Recognition Rate and Accuracy for First Experiment

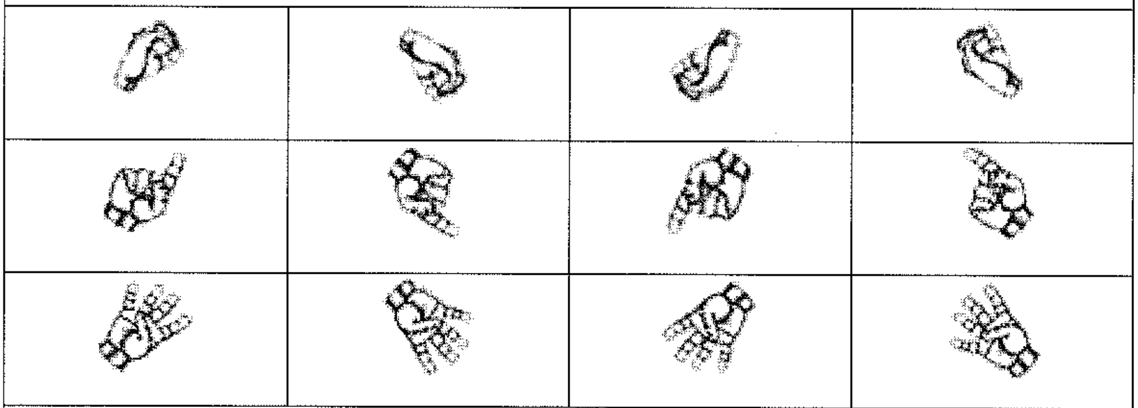
| Number | The Sign of Number | Accuracy |
|---|---|---|
| ZERO |  | 0.608975 |
| ONE |  | 0.776725 |
| TWO |  | 0.732175 |
| THREE |  | 0.775525 |
| FOUR |  | 0.987025 |
| FIVE |  | 0.728325 |
| SIX |  | 0.85435 |
| SEVEN |  | 0.619825 |
| EIGHT |  | 0.729075 |
| NINE |  | 0.59691 |
| **Total Accuracy** | | **0.74089** |
| **Neural Networks Recognition Rate** | | **(9/10) 100 %** |

## 4.4 Second Experiment

The second experiment was carried out by the same way and mechanism of the first experiment with the same number of training and testing sketches, but with the different training parameters values, and deferent sketches because added noise and rotation this sketches, this is what led to the difference in the accuracy and performance of the neural network and different recognition rate, tables below shows the training parameters values that used in this experiment, as well as training and testing time of the experiment, and the recognition rate, accuracy and performance of the neural network system, figure 4.3 below shows the training performance of the network. Appendix I shows the training and testing sketches.



a. An Example of Training sketches



b. An Example of Testing sketches

Figure 4.3 An Example of Training and Testing sketches for Experiment Two[69,70]

Table 4.4 shows the training and testing time and the final training parameters during experiment Two.

Table 4.4 Training Parameters and Training and Testing Timing Second Experiment

| | |
|---|---|
| Number of Input Neurons | 2025 |
| Number of Hidden Neurons | 20 |
| Number of Output Neurons | 10 |
| Weights Values Range | -0,35 and 0,35 |
| Learning Rate | 0.055 |
| Momentum Factor | 0.35 |
| Error | 0,0029 |
| Number Of Iteration | 462 |
| Maximum Iteration | 3000 |
| Training Time | 32.98 Sec |
| Testing Time | 0.0431 Sec |

Figure 4.4 shows the training performance curve of the artificial neural network system for the second experiment.

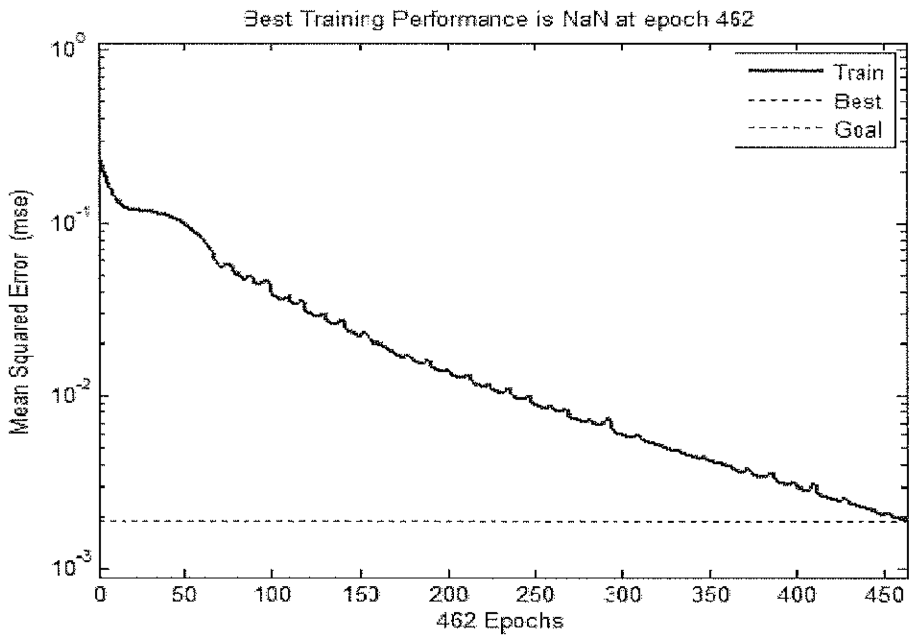



Figure 4.4 Second Experiment training performance Curve

Table 4.5 and table 4.6 shows the neural network training and testing recognition rate and accuracy that calculated from the second experiment training and testing sketches.

Table 4.5 Training Recognition Rate and Accuracy for Second Experiment

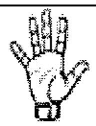
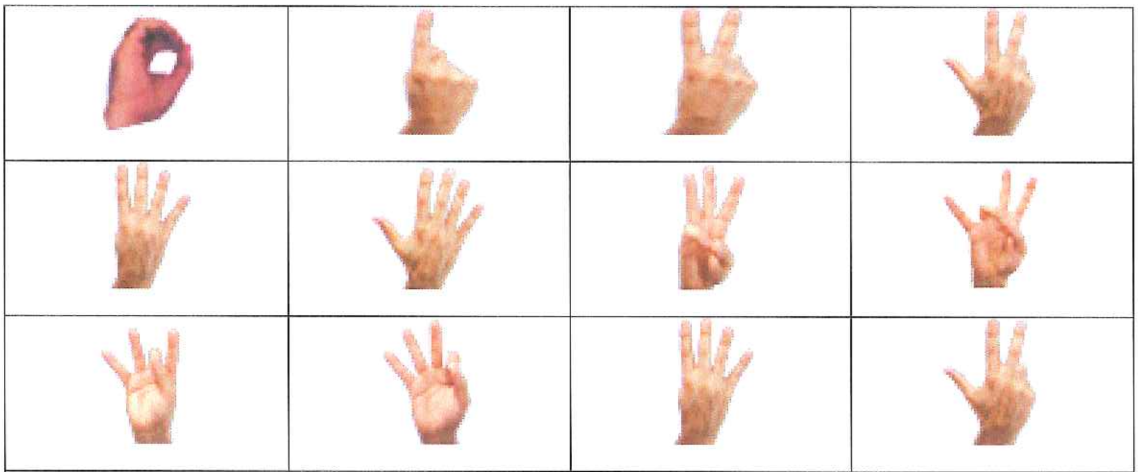
| Number | The Sign of Number | Accuracy |
|---|---|---|
| ZERO |  | **0.9988** |
| ONE |  | **0.99885** |
| TWO |  | **0.99882** |
| THREE |  | **0.99883** |
| FOUR |  | **0.99885** |
| FIVE |  | **0.99887** |
| SIX |  | **0.99892** |
| SEVEN |  | **0.9988** |
| EIGHT |  | **0.9988** |
| NINE |  | **0.99882** |
| **Total Accuracy** | | **0.99883** |
| **Neural Networks Recognition Rate** | | **(10/10) 100 %** |

Table 4.6 Testing Recognition Rate and Accuracy for Second Experiment

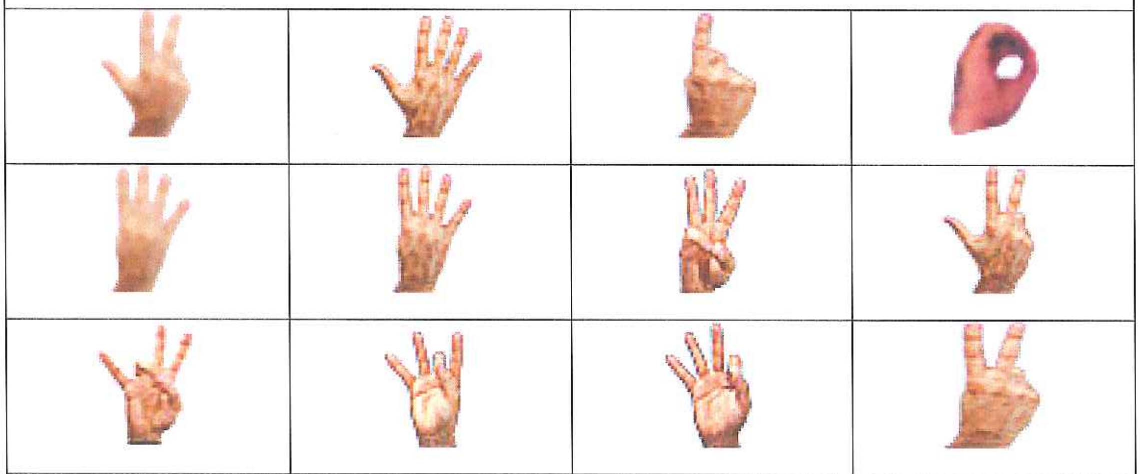
| Number | The Sign of Number | Accuracy |
|---|---|---|
| ZERO |  | 0.9032 |
| ONE |  | 0.5868 |
| TWO |  | 0.6477 |
| THREE |  | 0.4024 |
| FOUR |  | 0.638 |
| FIVE |  | 0.686 |
| SIX |  | 0.4458 |
| SEVEN |  | 0.5316 |
| EIGHT |  | 0.5615 |
| NINE |  | 0.3184 |
| **Total Accuracy** | | **0.514** |
| **Neural Networks Recognition Rate** | | **(6/10) 100 %** |

## 4.5 Third Experiment

This experiment was used real images, it mean the database consist of real images not sketches, this experiment was carried out by the same way and mechanism of the first and second experiments with the same number of training and testing real images, but with the different training parameters values, this is what led to the difference in the accuracy and performance of the neural network and different recognition rate.



a. An Example of Training Images



b. An Example of Testing Images

Figure 4.5 An Example of Training and Testing sketches for Experiment Three[69,70]

Table 4.7shows the training and testing time and the final training parameters during experiment three.

Table 4.7 Training Parameters and Training and Testing Timing Third Experiment

| | |
|---|---|
| Number of Input Neurons | 2025 |
| Number of Hidden Neurons | 90 |
| Number of Output Neurons | 10 |
| Weights Values Range | -0,35 and 0,35 |
| Learning Rate | 0.03 |
| Momentum Factor | 0.003 |
| Error | 0,003 |
| Number Of Iteration | 842 |
| Maximum Iteration | 10000 |
| Training Time | 35 Sec |
| Testing Time | 0.0431 Sec |

Figure 4.8 shows the training performance curve of the artificial neural network system for the third experiment.

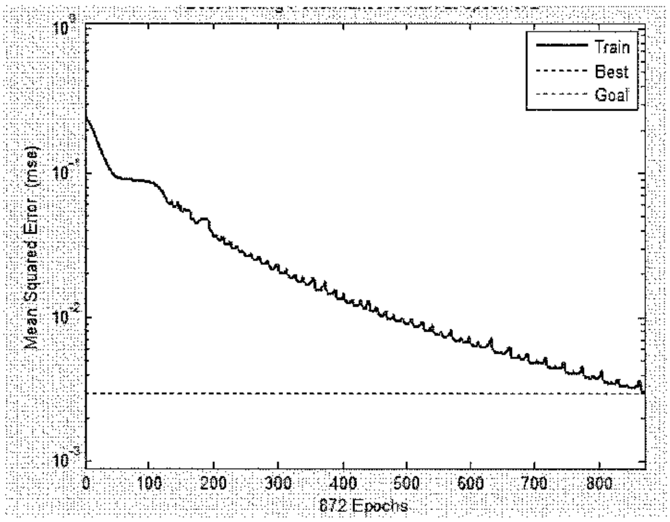



Figure 4.6 Third Experiment training performance Curve

Table 4.8 and table 4.9 shows the neural network training and testing recognition rate and accuracy that calculated from the third experiment training and testing images.

Table 4.8 Training Recognition Rate and Accuracy for Third Experiment

| **Number** | **The Sign of Number** | **Accuracy** |
|---|---|---|
| ZERO |  | **0.9988** |
| ONE |  | **0.99885** |
| TWO |  | **0.97882** |
| THREE |  | **0.99883** |
| FOUR |  | **0.99885** |
| FIVE |  | **0.97887** |
| SIX |  | **0.99892** |
| SEVEN |  | **0.9988** |
| EIGHT |  | **0.9898** |
| NINE |  | **0.95882** |
| **Total Accuracy** | | **0.97883** |
| **Neural Networks Recognition Rate** | | **(10/10) 100 %** |

Table 4.9 Testing Recognition Rate and Accuracy for Third Experiment

| Number | The Sign of Number | Accuracy |
|---|---|---|
| ZERO |  | **0.9032** |
| ONE |  | **0.8968** |
| TWO |  | **0.8797** |
| THREE |  | **0.7424** |
| FOUR |  | **0.8638** |
| FIVE |  | **0.7686** |
| SIX |  | **0.8458** |
| SEVEN |  | **0.8316** |
| EIGHT |  | **0.8615** |
| NINE |  | **0.8184** |
| **Total Accuracy** | | **0.8411** |
| **Neural Networks Recognition Rate** | | **(8/10) 100 %** |

## 4.6 Discussion

Several experiments were carried out on the sign language and gesture recognition system. The organization of the experiments was based on training and testing images which are used in training and testing process and training parameters which are used in training process. In the experiments, the number of training and testing images was forty images for training process and forty image for testing process. The best results were obtained through the first experiment with highest recognition rate and accuracy this compare between the first and second experiments because it was used sketches, the third experiment was used real images. After training and testing of sign recognition system it seems that the neural network is good tool for recognition task.

After looking to train and test results for the experiments, it seems that the neural network can learn with four training sets and recognizes other four sets with high accuracy.

## 4.7 Summary

This chapter explained the experimental result and performance analysis that has been carried out through training and testing the sign language and gesture recognition system. The results demonstrated the successful implementation of the system.

# CONCLUSION

The Sign Language of any culture has little connection with the oral language of that culture. There is no one to one relationship between sign languages and oral languages in most cultures. Sign language normally involves 2 hands, facial expressions and shoulder movement.

In this thesis was to identify some images signals deaf and dumb and gestures. To try to facilitate communication between the deaf and dumb themselves and among normal people. Because this language is not known everyone . This system will be similar to the dictionary in which some signals deaf and dumb and gestures and what meanings to facilitate the communication between people.

In this system used numbers between (0-9) were recognition by artificial neural networks and used back propagation function. There are three experiments were applied to these images. Database for the first and second experiments consisting of schemes, But the data base of the third experiment consists of real images.

Four images were taken for training and four images of the testing for each gesture in three experiments.

Were discussing the results we have obtained in the fourth chapter of this thesis.

## REFERENCES


1. Lynette P., 2009, The acquisition of New Zealand Sign Language as a second language for students in an interpreting programme: The learners' perspective, Master of Arts in Applied Language Studies, Auckland University of Technology.
2. http://slis.ie/images/uploads/Guidelines_for_Deaf_Clients.pdf June/2012.
3. http://en.wikipedia.org/wiki/American_Sign_Language June/2012.
4. Rod R. and Mickey F., 1995, The Perigee Visual Dictionary of Signing, Berkley Publishing Group.
5. Hallen, 24 February 1998, A History of Sign Language, Mid-Term Paper.
6. http://www.wfdeaf.org/ July/2012.
7. http://www.lifeprint.com/asl101/topics/gloss.htm July/2012.
8. http://www.cdss.ca.gov/cdssweb/entres/forms/English/pub391.pdf July/2012.
9. Wendy S., Sign Language Phonology, The Oxford International Encyclopedia of Linguistics.
10. Peter, W., 1990,Recognition of Sign Language Using Neural Networks, Flinders University of South Australia.
11. http://www.fluentin3months.com/asl-videos/ July/2012.
12. Wendy S. and Diane L., Natural Sign Languages, In Handbook of Linguistics, 2001.
13. http://publish.uwo.ca/~rmoir2/docs/CriticalThinking%20Tutorial%202.pdf July/2012.
14. Brian, M., Second Language Acquisition and the Competition Model, Carnegie Mellon University.
15. Bayley, R., and Mary, R., Phonological variation in American Sign Language The case of hand shape, University of Texas.
16. Sharon, L., July 1, 2011, Get a Grip on Fingerspelling, Professional Certification.
17. http://en.wikipedia.org/wiki/British_Sign_Language July/2012.
18. http://en.wikipedia.org/wiki/Sign_language July/2012.
19. Jim, C., The Relationship between American Sign Language Proficiency and English Academic Development, Toronto University.
20. http://en.wikipedia.org/wiki/Washoe_(chimpanzee) June/2012.
21. Ying, T., Takeo, K. and Jeffrey, F., Recognizing Action Units for Facial Expression Analysis, Robotics Institute, Carnegie Mellon University, Department of Psychology, University of Pittsburgh.

22. Barbara, L., Sudeep S., Ayush P. and Karshmer I., Progress in Automated Computer Recognition of Sign Language, Department of Special Education, Computer Science and Engineering and Department of Information Technology, University of Florida.
23. Peter Wray Vamplew, 1990, Recognition of Sign Language Using Neural Networks, B.A, B.Sc. (Hons.), Flinders University of South Australia.
24. http://en.wikipedia.org/wiki/Deafness#Sign_language may/2012.
25. YEGNANARAYANA, B., 8 September 1993, Artificial neural networks for pattern recognition, Department Computer Science and Engineering, Indian Institute.
26. Simon H, 1999, Neural Network A Comprehensive Foundation, Second Edition, Hamilton, Ontario, Canada.
27. Ivan G, U. MASS Lowell, Crash Introduction to Artificial Neural Networks, Materials for UML 91.531 Data Mining course.
28. http://en.wikipedia.org/wiki/Neuron may/2012.
29. http://en.wikipedia.org/wiki/Artificial_neural_network June/2012.
30. Eduardo, A., ARTIFICIAL NEURAL NETWORK, Syllabus.
31. http://www.mitpressjournals.org/doi/abs/10.1162/neco.1989.1.3.295 July/2012.
32. http://en.wikipedia.org/wiki/Supervised_learning June /2012.
33. Kshirsagar, A. and Rathod, M., Artificial Neural Network, K.B.P. College of Engg. & Poly, Satara-415002, K.B.P. College of Engg. & Poly, Satara-415002.
34. http://www.oocities.org/hjayathilake/an7.html July /2012.
35. GIRISH, K., Artificial Neural Network, Indian Agricultural Research Institute PUSA, New Delhi-110012.
36. Pete, Mc, C., An Introduction to Back-Propagation Neural Networks, Encoder.
37. http://www.oocities.org/hjayathilake.html July /2012.
38. Suzanna, B., 1992, An Information-theoretic Unsupervised Learning Algorithm for Neural Networks, Department of Computer Science, University of Toronto.
39. YEGNANARAYANA, B., 2006, Artificial Neural Networks, Department of Computer Science and Engineering Indian Institute of Technology Madras Chennai.
40. Khalid, I., Unsupervised Neural Network: Hebb Rule Implementation, Underwater Robotics Research Group Website.
41. Deepak, M., and Prem, K., 12 February 2007, Modified Hopfield Neural Network Approach for Solving Nonlinear Algebraic Equations, Advance online publication.
42. http://www.cs.indiana.edu/classes/b351-gass/Notes/backprop.html July /2012.

43. Tom, M., October 6 2011, Machine Learning 10-601, Machine Learning Department Carnegie Mellon University.
44. Wlodzislaw, D., and Antoine, N., Multi Dimensional Scaling and Self Organized Mapping, Department of Computer Methods, Nicholas Copernicus University Grudzi adzka-Torun, Poland.
45. Sivanandam, N., Sumathi and Deepa, 2006, Introduction to Neural Networks Using Matlab 6.0.
46. http://en.wikipedia.org/wiki/Multilayer_perceptron June /2012.
47. http://hagan.okstate.edu/2_Architectures.pdf July /2012.
48. Martin, R., and Heinrich, B., A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Institut fur Logik, Komplexitat und Deduktionssyteme University of Karlsruhe.
49. Kiri, F., February 1, 2008, ANN Backpropagation: Weight updates for hidden nodes.
50. Jason, B., January 2011 Clever Algorithms: Nature-Inspired Programming Recipes, First Edition, Clever Algorithms.com.
51. Tung-Chueng, C., Ru-Jen C., 19April 2006, Application of back-propagation networks in debris flow prediction, Department of Civil Engineering, Kao-Yuan University, China, Department of Information Management, Kao-Yuan University, Republic of China.
52. http://page.mi.fu-berlin.de/rojas/neural/chapter/K7.pdf June /2012.
53. Youssef, B., 2011 Neural Network Model for Path-Planning Of Robotic Rover Systems LACSC – Lebanese Association for Computational Sciences, Lebanon.
54. David, J., Montana and Lawrence, D., Training Feed forward Neural Networks Using Genetic Algorithms, BBN Systems and Technologies Corp. Cambridge.
55. David, L., A Basic Introduction to Feed forward Backpropagation Neural Networks, Associate Professor of Geosciences.
56. http://en.wikipedia.org/wiki/Feedforward_neural_network July/2012.
57. http://www4.rgu.ac.uk/files/chapter3%20-%20bp.pdf July /2012.
58. Russell, C. Eberhart and Yuhui, Sh., 2007, Computational Intelligence: Concepts to Implementations.
59. Moreira, M. and Fiesler, E., October 1995, Neural Networks Adaptive Learning Rate and Momentum Terms, IDAP Technical report.
60. http://www.willamette.edu/~gorr/classes/cs449/momrate.html June /2012.

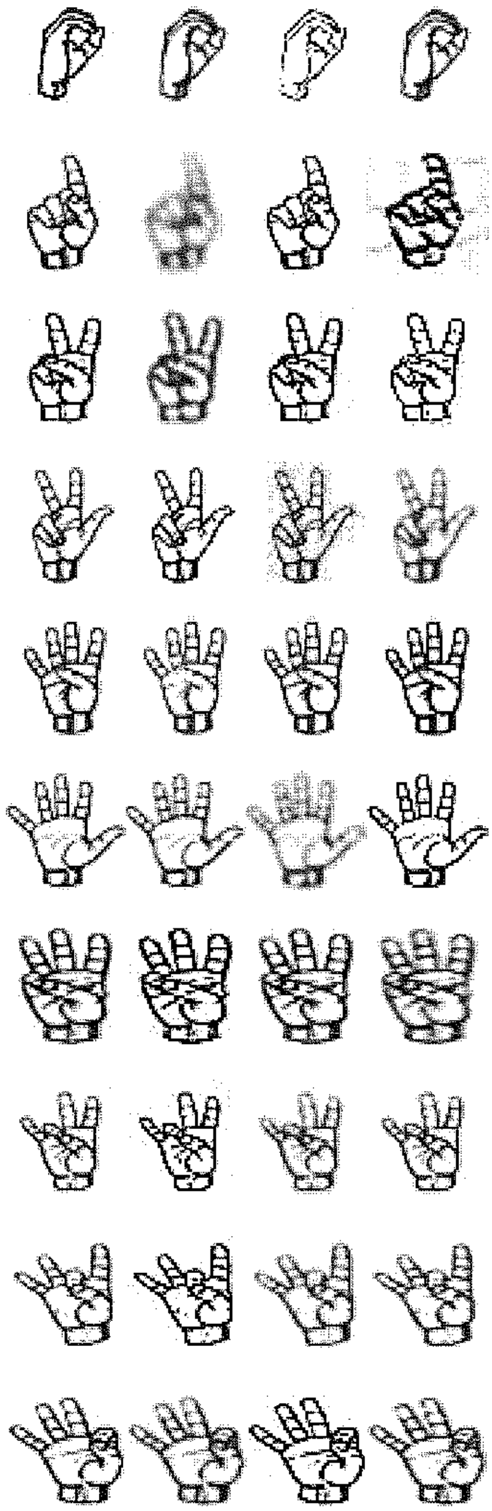
61. Carlos, G., Artificial Neural Networks for Beginners, Cognitive and Computing Science, University of Sussex.
62. Neil, A., Andrew, S. and Doug, T., A Three-Unit Network is All You Need to Discover Females, Department of Computer Science University of California, San Diego.
63. Ehud, D., 2.JUNE 1990, A Simple Procedure for Pruning Back-Propagation Trained Neural Networks, IEEE TRANSACTIONS ON NEURAL NETWORK.
64. Hinton, G., Srivastava, N., Krizhevsky, A. and Salakhutdinov, R., 3July 2012, Improving neural networks by preventing co-adaptation of feature dtectors, Department of Computer Science, University of Toronto.
65. Daivid, C., Steven, J. and Geoffrey, E., June 1986, Experiments on Learning by Back propagation, Computer Science Department.
66. Stefan, Z., December 2003, Data Mining for Prediction, The Royal Institute of Technology, Department of Computer and Systems Sciences.
67. Giustolisi, O. and Laucelli, D., June 2005, Improving generalization of artificial neural networks in rainfall-runoff modeling, Hydrological Sciences-Journal-des sciences Hydrologiques.
68. Warren, S., Sarle and Cary, N., 2002, Questions Part 3: Generalization, Usenet newsgroup comp. ai.neural-nets, USA.
69. http://people.howstuffworks.com/sign-language2.htm July /2012.
70. http://lifeprint.com/asl101/pages-signs/n/numbers.htm July /2012.
71. http://en.wikipedia.org/wiki/Otsu's_method July /2012.
72. http://en.wikipedia.org/wiki/Digital_image June /2012.
73. Kristian, S., Introduction to image processing in Matlab, Department of Applied Mathematics, University of Colorado at Boulder.

# APPENDIX I

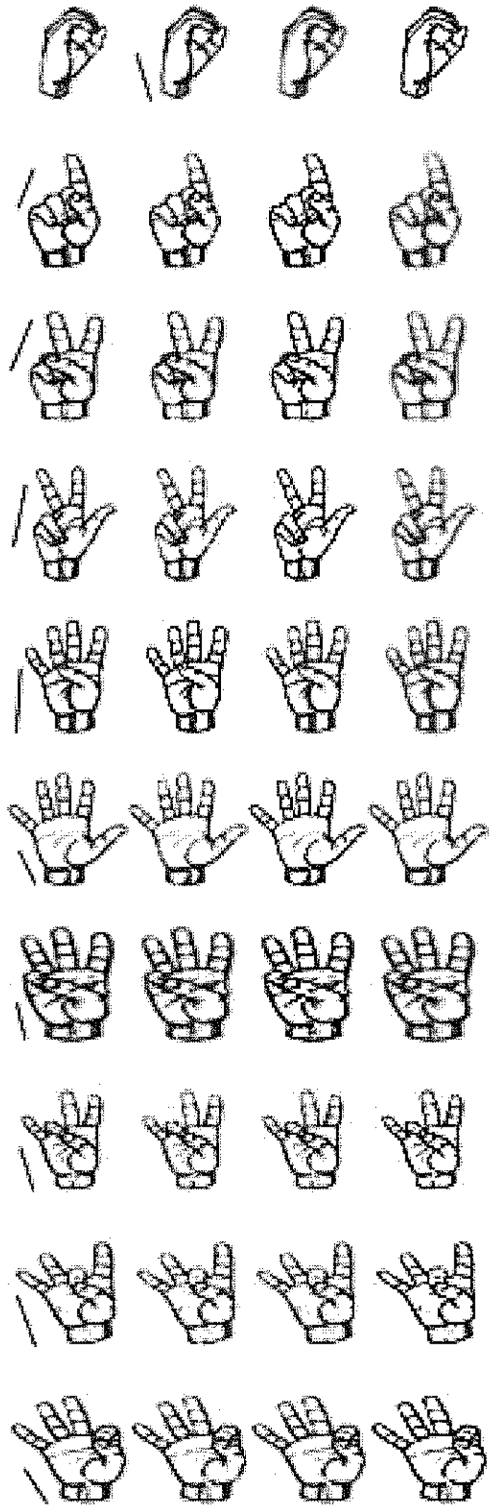
## Database

### 1. First Experiment

1.1 Training Database [69,70]

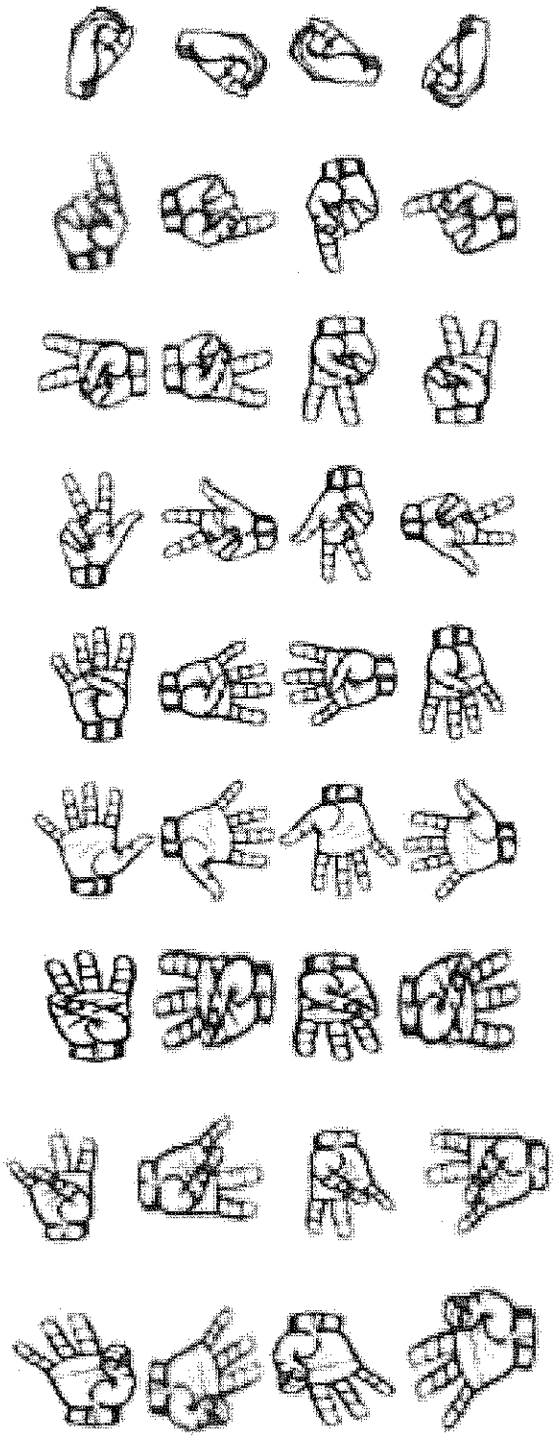


## 1.2 Testing Database

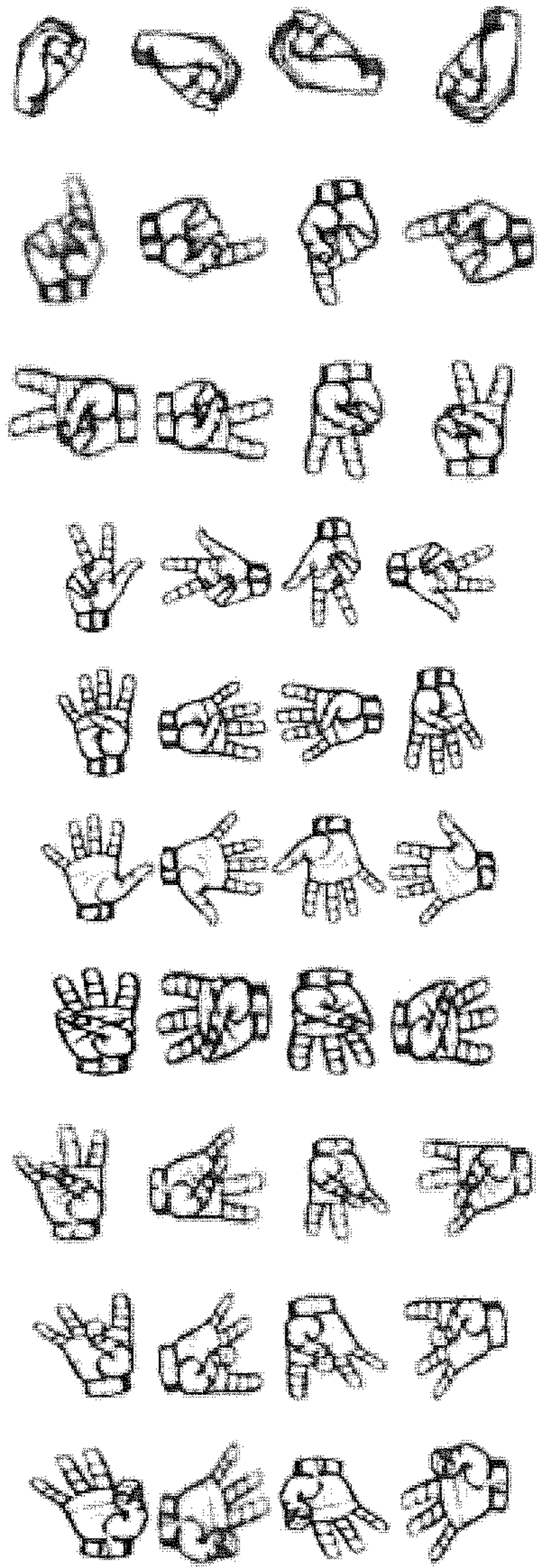


## 2. Second Experiment

2.1 Training Database [69,70]



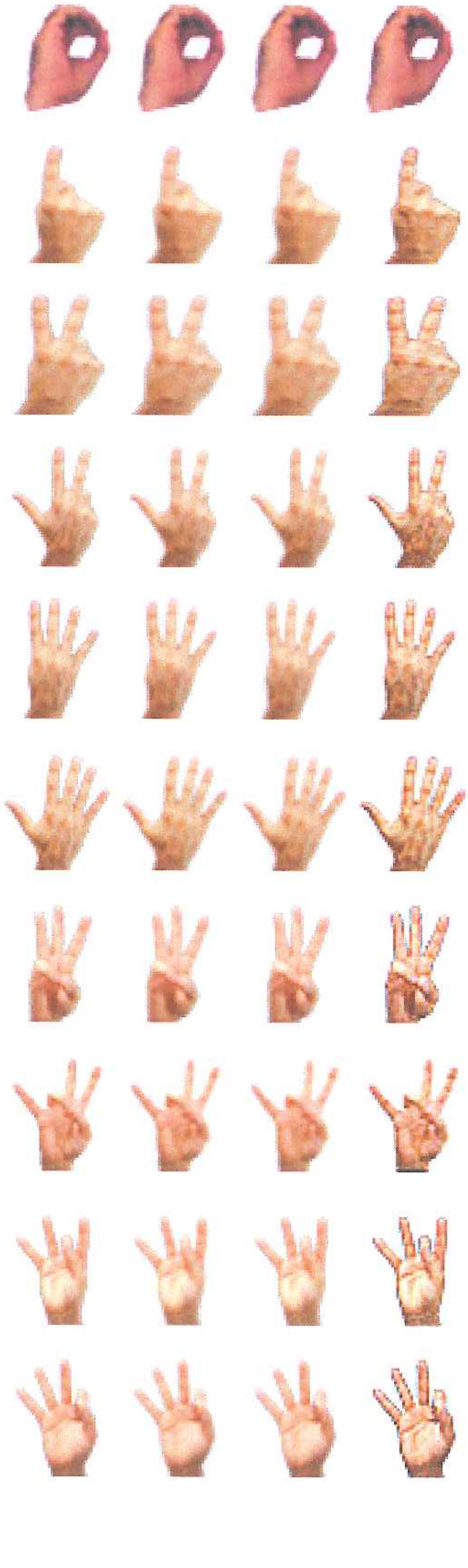
## 2.2 Testing Database



## 3. Third Experiment

3.1 Training Database [69,70]

3.2 Testing Database



# APPENDIX II

# MATLAB SOURSE CODE

## 1. First Experiment

```
clear all;
close all;
signal_number = 10;
signal_expression = 4;
PATTERNS = [];
cd('C:\Users\Dell\Desktop\database22\bm');
x1=imread('a1.jpg');x1=imresize(x1,[45 45],'bicubic');
x1=rgb2gray(x1);
x2=imread('a2.jpg'); x2=imresize(x2,[45
45],'bicubic');x2=rgb2gray(x2);
x3=imread('a3.jpg'); x3=imresize(x3,[45 45],'bicubic');
x3=rgb2gray(x3);
x4=imread('a4.jpg'); x4=imresize(x4,[45
45],'bicubic');x4=rgb2gray(x4);
x5=imread('a5.jpg'); x5=imresize(x5,[45 45],'bicubic');
x5=rgb2gray(x5);
x6=imread('a6.jpg'); x6=imresize(x6,[45 45],'bicubic');
x6=rgb2gray(x6);
x7=imread('a7.jpg'); x7=imresize(x7,[45 45],'bicubic');
x7=rgb2gray(x7);
x8=imread('a8.jpg'); x8=imresize(x8,[45 45],'bicubic');
x8=rgb2gray(x8);
x9=imread('a9.jpg'); x9=imresize(x9,[45 45],'bicubic');
x9=rgb2gray(x9);
x10=imread('a10.jpg'); x10=imresize(x10,[45
45],'bicubic');x10=rgb2gray(x10);
x11=imread('a11.jpg'); x11=imresize(x11,[45
45],'bicubic');x11=rgb2gray(x11);
x12=imread('a12.jpg');x12=imresize(x12,[45
45],'bicubic');x12=rgb2gray(x12);
x13=imread('a13.jpg'); x13=imresize(x13,[45 45],'bicubic');
x13=rgb2gray(x13);
x14=imread('a14.jpg'); x14=imresize(x14,[45
45],'bicubic');x14=rgb2gray(x14);
x15=imread('a15.jpg'); x15=imresize(x15,[45
45],'bicubic');x15=rgb2gray(x15);
x16=imread('a16.jpg'); x16=imresize(x16,[45 45],'bicubic');
x16=rgb2gray(x16);
x17=imread('a17.jpg'); x17=imresize(x17,[45 45],'bicubic');
x17=rgb2gray(x17);
x18=imread('a18.jpg'); x18=imresize(x18,[45
45],'bicubic');x18=rgb2gray(x18);
x19=imread('a19.jpg'); x19=imresize(x19,[45
45],'bicubic');x19=rgb2gray(x19);
x20=imread('a20.jpg'); x20=imresize(x20,[45
45],'bicubic');x20=rgb2gray(x20);
x21=imread('a21.jpg'); x21=imresize(x21,[45
45],'bicubic');x21=rgb2gray(x21);
x22=imread('a22.jpg'); x22=imresize(x22,[45
45],'bicubic');x22=rgb2gray(x22);
x23=imread('a23.jpg'); x23=imresize(x23,[45
45],'bicubic');x23=rgb2gray(x23);
x24=imread('a24.jpg'); x24=imresize(x24,[45
45],'bicubic');x24=rgb2gray(x24);
```

```
x25=imread('a25.jpg'); x25=imresize(x25,[45
45],'bicubic');x25=rgb2gray(x25);
x26=imread('a26.jpg'); x26=imresize(x26,[45
45],'bicubic');x26=rgb2gray(x26);
x27=imread('a27.jpg'); x27=imresize(x27,[45
45],'bicubic');x27=rgb2gray(x27);
x28=imread('a28.jpg'); x28=imresize(x28,[45
45],'bicubic');x28=rgb2gray(x28);
x29=imread('a29.jpg'); x29=imresize(x29,[45
45],'bicubic');x29=rgb2gray(x29);
x30=imread('a30.jpg'); x30=imresize(x30,[45
45],'bicubic');x30=rgb2gray(x30);
x31=imread('a31.jpg'); x31=imresize(x31,[45
45],'bicubic');x31=rgb2gray(x31);
x32=imread('a32.jpg'); x32=imresize(x32,[45
45],'bicubic');x32=rgb2gray(x32);
x33=imread('a33.jpg'); x33=imresize(x33,[45
45],'bicubic');x33=rgb2gray(x33);
x34=imread('a34.jpg'); x34=imresize(x34,[45
45],'bicubic');x34=rgb2gray(x34);
x35=imread('a35.jpg'); x35=imresize(x35,[45
45],'bicubic');x35=rgb2gray(x35);
x36=imread('a36.jpg'); x36=imresize(x36,[45
45],'bicubic');x36=rgb2gray(x36);
x37=imread('a37.jpg'); x37=imresize(x37,[45
45],'bicubic');x37=rgb2gray(x37);
x38=imread('a38.jpg'); x38=imresize(x38,[45
45],'bicubic');x38=rgb2gray(x38);
x39=imread('a39.jpg'); x39=imresize(x39,[45
45],'bicubic');x39=rgb2gray(x39);
x40=imread('a40.jpg'); x40=imresize(x40,[45
45],'bicubic');x40=rgb2gray(x40);

P1 = double([x1(:) x2(:) x3(:) x4(:)]/255);
P2 = double([ x5(:) x6(:) x7(:) x8(:)]/255);
P3 = double([x9(:) x10(:) x11(:) x12(:)]/255);
P4 = double([x13(:) x14(:) x15(:) x16(:)]/255);
P5 = double([x17(:) x18(:) x19(:) x20(:)]/255);
P6 = double([x21(:) x22(:) x23(:) x24(:)]/255);
P7 = double([x25(:) x26(:) x27(:) x28(:)]/255);
P8 = double([x29(:) x30(:) x31(:) x32(:)]/255);
P9 = double([x33(:) x34(:) x35(:) x36(:)]/255);
P10 = double([x37(:) x38(:) x39(:) x40(:)]/255);


patterns =[P1 P2 P3 P4 P5 P6 P7 P8 P9 P10];

%DATABASE FOR signal
 M = [1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0;...
      0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0;...
      0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0;...
      0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0;...
      0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0;...
      0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0;...
```

```
      0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0;...
      0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0;...
      0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0;...
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1];


 [g,h]=size(patterns);
[m,h]=size(M);
% CREATING AND INITIATING THE NETWORK
net = newff(minmax(patterns),[90 10],{'logsig','logsig'},'traingdx');
%( for change *************
net = init(net);
net.LW{2,1} = net.LW{2,1}*0.01;
net.b{2} = net.b{2}*0.01;



% TRAINING THE NETWORK
net.trainParam.goal = 0.003; % Sum-squared error goal.
net.trainParam.lr = 0.007;% Learning Rate ( for change *************
net.trainParam.show = 10; % Frequency of progress displays (in
epochs).
net.trainParam.epochs = 3000; % Maximum number of epochs to train.
net.trainParam.mc = 0.03; % Momentum Factor.( for change *************

[net,tr] = train(net,patterns,M);


% Train Results
for k=1:h
    patterns(:,k);
signal = sim(net,patterns(:,k))
end;

pause; pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
signal_number = 10;
signal_expression = 4;
test_patterns = [];
Thresh = 0.8;

y1=imread('t1.jpg'); y1=imresize(y1,[45 45],'bicubic');
y1=rgb2gray(y1);
y2=imread('t2.jpg'); y2=imresize(y2,[45
45],'bicubic');y2=rgb2gray(y2);
y3=imread('t3.jpg'); y3=imresize(y3,[45 45],'bicubic');
y3=rgb2gray(y3);
y4=imread('t4.jpg'); y4=imresize(y4,[45
45],'bicubic');y4=rgb2gray(y4);
y5=imread('t5.jpg'); y5=imresize(y5,[45 45],'bicubic');
y5=rgb2gray(y5);
y6=imread('t6.jpg'); y6=imresize(y6,[45 45],'bicubic');
y6=rgb2gray(y6);
```

```
y7=imread('t7.jpg'); y7=imresize(y7,[45 45],'bicubic');
y7=rgb2gray(y7);
y8=imread('t8.jpg'); y8=imresize(y8,[45 45],'bicubic');
y8=rgb2gray(y8);
y9=imread('t9.jpg'); y9=imresize(y9,[45 45],'bicubic');
y9=rgb2gray(y9);
y10=imread('t10.jpg'); y10=imresize(y10,[45
45],'bicubic');y10=rgb2gray(y10);
y11=imread('t11.jpg'); y11=imresize(y11,[45
45],'bicubic');y11=rgb2gray(y11);
y12=imread('t12.jpg');y12=imresize(y12,[45
45],'bicubic');y12=rgb2gray(y12);
y13=imread('t13.jpg'); y13=imresize(y13,[45 45],'bicubic');
y13=rgb2gray(y13);
y14=imread('t14.jpg'); y14=imresize(y14,[45
45],'bicubic');y14=rgb2gray(y14);
y15=imread('t15.jpg'); y15=imresize(y15,[45
45],'bicubic');y15=rgb2gray(y15);
y16=imread('t16.jpg'); y16=imresize(y16,[45 45],'bicubic');
y16=rgb2gray(y16);
y17=imread('t17.jpg'); y17=imresize(y17,[45 45],'bicubic');
y17=rgb2gray(y17);
y18=imread('t18.jpg'); y18=imresize(y18,[45
45],'bicubic');y18=rgb2gray(y18);
y19=imread('t19.jpg'); y19=imresize(y19,[45
45],'bicubic');y19=rgb2gray(y19);
y20=imread('t20.jpg'); y20=imresize(y20,[45
45],'bicubic');y20=rgb2gray(y20);
y21=imread('t21.jpg'); y21=imresize(y21,[45
45],'bicubic');y21=rgb2gray(y21);
y22=imread('t22.jpg'); y22=imresize(y22,[45
45],'bicubic');y22=rgb2gray(y22);
y23=imread('t23.jpg'); y23=imresize(y23,[45
45],'bicubic');y23=rgb2gray(y23);
y24=imread('t24.jpg'); y24=imresize(y24,[45
45],'bicubic');y24=rgb2gray(y24);
y25=imread('t25.jpg'); y25=imresize(y25,[45
45],'bicubic');y25=rgb2gray(y25);
y26=imread('t26.jpg'); y26=imresize(y26,[45
45],'bicubic');y26=rgb2gray(y26);
y27=imread('t27.jpg'); y27=imresize(y27,[45
45],'bicubic');y27=rgb2gray(y27);
y28=imread('t28.jpg'); y28=imresize(y28,[45
45],'bicubic');y28=rgb2gray(y28);
y29=imread('t29.jpg'); y29=imresize(y29,[45
45],'bicubic');y29=rgb2gray(y29);
y30=imread('t30.jpg'); y30=imresize(y30,[45
45],'bicubic');y30=rgb2gray(y30);
y31=imread('t31.jpg'); y31=imresize(y31,[45
45],'bicubic');y31=rgb2gray(y31);
y32=imread('t32.jpg'); y32=imresize(y32,[45
45],'bicubic');y32=rgb2gray(y32);
y33=imread('t33.jpg'); y33=imresize(y33,[45
45],'bicubic');y33=rgb2gray(y33);
y34=imread('t34.jpg'); y34=imresize(y34,[45
45],'bicubic');y34=rgb2gray(y34);
y35=imread('t35.jpg'); y35=imresize(y35,[45
45],'bicubic');y35=rgb2gray(y35);
y36=imread('t36.jpg'); y36=imresize(y36,[45
45],'bicubic');y36=rgb2gray(y36);
```

```
y37=imread('t37.jpg'); y37=imresize(y37,[45
45],'bicubic');y37=rgb2gray(y37);
y38=imread('t38.jpg'); y38=imresize(y38,[45
45],'bicubic');y38=rgb2gray(y38);
y39=imread('t39.jpg'); y39=imresize(y39,[45
45],'bicubic');y39=rgb2gray(y39);
y40=imread('t40.jpg'); y40=imresize(y40,[45
45],'bicubic');y40=rgb2gray(y40);




P1_test = double([y1(:) y2(:) y3(:) y4(:)]/255);
P2_test = double([ y5(:) y6(:) y7(:) y8(:)]/255);
P3_test = double([y9(:) y10(:) y11(:) y12(:)]/255);
P4_test = double([y13(:) y14(:) y15(:) y16(:)]/255);
P5_test = double([y17(:) y18(:) y19(:) y20(:)]/255);
P6_test = double([y21(:) y22(:) y23(:) y24(:)]/255);
P7_test = double([y25(:) y26(:) y27(:) y28(:)]/255);
P8_test = double([y29(:) y30(:) y31(:) y32(:)]/255);
P9_test = double([y33(:) y34(:) y35(:) y36(:)]/255);
P10_test = double([y37(:) y38(:) y39(:) y40(:)]/255);


test_pattern=[P1_test P2_test P3_test P4_test P5_test P6_test P7_test
P8_test P8_test P9_test P10_test];


[m,n]=size(test_pattern);
%s=0;
for k=1:n
    test_pattern(:,k);
   % s=s+1;
diseasestest = sim(net,test_pattern(:,k))
end

cd('C:\Users\Dell\Desktop\database22\bm');
```

## 2. The Second Experiment

```
clear all;
close all;
signal_number = 10;
signal_expression = 4;
PATTERNS = [];
%cd('F:\database2_correction\Database');
x1=imread('a1.jpg'); x1=imresize(x1,[45 45],'bicubic');
x1=rgb2gray(x1);
x2=imread('a2.jpg'); x2=imresize(x2,[45
45],'bicubic');x2=rgb2gray(x2);
x3=imread('a3.jpg'); x3=imresize(x3,[45 45],'bicubic');
x3=rgb2gray(x3);
x4=imread('a4.jpg'); x4=imresize(x4,[45
45],'bicubic');x4=rgb2gray(x4);
x5=imread('a5.jpg'); x5=imresize(x5,[45 45],'bicubic');
x5=rgb2gray(x5);
x6=imread('a6.jpg'); x6=imresize(x6,[45 45],'bicubic');
x6=rgb2gray(x6);
x7=imread('a7.jpg'); x7=imresize(x7,[45 45],'bicubic');
x7=rgb2gray(x7);
x8=imread('a8.jpg'); x8=imresize(x8,[45 45],'bicubic');
x8=rgb2gray(x8);
x9=imread('a9.jpg'); x9=imresize(x9,[45 45],'bicubic');
x9=rgb2gray(x9);
x10=imread('a10.jpg'); x10=imresize(x10,[45
45],'bicubic');x10=rgb2gray(x10);
x11=imread('a11.jpg'); x11=imresize(x11,[45
45],'bicubic');x11=rgb2gray(x11);
x12=imread('a12.jpg');x12=imresize(x12,[45
45],'bicubic');x12=rgb2gray(x12);
x13=imread('a13.jpg'); x13=imresize(x13,[45 45],'bicubic');
x13=rgb2gray(x13);
x14=imread('a14.jpg'); x14=imresize(x14,[45
45],'bicubic');x14=rgb2gray(x14);
x15=imread('a15.jpg'); x15=imresize(x15,[45
45],'bicubic');x15=rgb2gray(x15);
x16=imread('a16.jpg'); x16=imresize(x16,[45 45],'bicubic');
x16=rgb2gray(x16);
x17=imread('a17.jpg'); x17=imresize(x17,[45 45],'bicubic');
x17=rgb2gray(x17);
x18=imread('a18.jpg'); x18=imresize(x18,[45
45],'bicubic');x18=rgb2gray(x18);
x19=imread('a19.jpg'); x19=imresize(x19,[45
45],'bicubic');x19=rgb2gray(x19);
x20=imread('a20.jpg'); x20=imresize(x20,[45
45],'bicubic');x20=rgb2gray(x20);
x21=imread('a21.jpg'); x21=imresize(x21,[45
45],'bicubic');x21=rgb2gray(x21);
x22=imread('a22.jpg'); x22=imresize(x22,[45
45],'bicubic');x22=rgb2gray(x22);
x23=imread('a23.jpg'); x23=imresize(x23,[45
45],'bicubic');x23=rgb2gray(x23);
x24=imread('a24.jpg'); x24=imresize(x24,[45
45],'bicubic');x24=rgb2gray(x24);
x25=imread('a25.jpg'); x25=imresize(x25,[45
45],'bicubic');x25=rgb2gray(x25);
x26=imread('a26.jpg'); x26=imresize(x26,[45
45],'bicubic');x26=rgb2gray(x26);
```

```
x27=imread('a27.jpg'); x27=imresize(x27,[45
45],'bicubic');x27=rgb2gray(x27);
x28=imread('a28.jpg'); x28=imresize(x28,[45
45],'bicubic');x28=rgb2gray(x28);
x29=imread('a29.jpg'); x29=imresize(x29,[45
45],'bicubic');x29=rgb2gray(x29);
x30=imread('a30.jpg'); x30=imresize(x30,[45
45],'bicubic');x30=rgb2gray(x30);
x31=imread('a31.jpg'); x31=imresize(x31,[45
45],'bicubic');x31=rgb2gray(x31);
x32=imread('a32.jpg'); x32=imresize(x32,[45
45],'bicubic');x32=rgb2gray(x32);
x33=imread('a33.jpg'); x33=imresize(x33,[45
45],'bicubic');x33=rgb2gray(x33);
x34=imread('a34.jpg'); x34=imresize(x34,[45
45],'bicubic');x34=rgb2gray(x34);
x35=imread('a35.jpg'); x35=imresize(x35,[45
45],'bicubic');x35=rgb2gray(x35);
x36=imread('a36.jpg'); x36=imresize(x36,[45
45],'bicubic');x36=rgb2gray(x36);
x37=imread('a37.jpg'); x37=imresize(x37,[45
45],'bicubic');x37=rgb2gray(x37);
x38=imread('a38.jpg'); x38=imresize(x38,[45
45],'bicubic');x38=rgb2gray(x38);
x39=imread('a39.jpg'); x39=imresize(x39,[45
45],'bicubic');x39=rgb2gray(x39);
x40=imread('a40.jpg'); x40=imresize(x40,[45
45],'bicubic');x40=rgb2gray(x40);

a={x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,x12,x13,x14,x15,x16,x17,x18,x19,
x20,x21,x22,x23,x24,x25,x26,x27,x28,x29,x30,x31,x32,x33,x34,x35,x36,x3
7,x38,x39,x40};

for i=1:40
    b(:,i)=reshape(a{1,i(:,:)},[],1);
end
 patterns=[];
for j=1:4
    for i=j:4:36+j
        patterns=[patterns b(:,i)];
    end
end

patterns=double(patterns)./255;
% P1 = (double([x1(:) x2(:) x3(:) x4(:)])/255);
% P2 = (double([ x5(:) x6(:) x7(:) x8(:)])/255);
% P3 = (double([x9(:) x10(:) x11(:) x12(:)])/255);
% P4 = (double([x13(:) x14(:) x15(:) x16(:)])/255);
% P5 = (double([x17(:) x18(:) x19(:) x20(:)])/255);
% P6 = (double([x21(:) x22(:) x23(:) x24(:)])/255);
% P7 = (double([x25(:) x26(:) x27(:) x28(:)])/255);
% P8 = (double([x29(:) x30(:) x31(:) x32(:)])/255);
% P9 = (double([x33(:) x34(:) x35(:) x36(:)])/255);
% P10 =(double([x37(:) x38(:) x39(:) x40(:)])/255);
%
%
% patterns =[P1 P2 P3 P4 P5 P6 P7 P8 P9 P10];

%DATABASE FOR signal
```

```
 M = [1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0;...
      0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0;...
      0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0;...
      0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0;...
      0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0;...
      0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0;...
      0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0;...
      0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0;...
      0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0;...
      0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1];


 [g,h]=size(patterns);
[m,h]=size(M);
% CREATING AND INITIATING THE NETWORK
net = newff(minmax(patterns),[150 10],{'logsig','logsig'},'traingdx');
%( for change *************
net = init(net);
net.LW{2,1} = net.LW{2,1}*0.01;
net.b{2} = net.b{2}*0.01;



% TRAINING THE NETWORK
net.trainParam.goal = 0.003; % Sum-squared error goal.
net.trainParam.lr = 0.04;% Learning Rate ( for change *************
net.trainParam.show = 10; % Frequency of progress displays (in
epochs).
net.trainParam.epochs = 3000; % Maximum number of epochs to train.
net.trainParam.mc = 0.5; % Momentum Factor.( for change *************

[net,tr] = train(net,patterns,M);


% Train Results
for k=1:h
    patterns(:,k);
traning = sim(net,patterns(:,k))
end;

pause; pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
signal_number = 10;
signal_expression = 4;
test_patterns = [];
Thresh = 0.8;
```

```
y1=imread('t1.jpg'); y1=imresize(y1,[45 45],'bicubic');
y1=rgb2gray(y1);
y2=imread('t2.jpg'); y2=imresize(y2,[45
45],'bicubic');y2=rgb2gray(y2);
y3=imread('t3.jpg'); y3=imresize(y3,[45 45],'bicubic');
y3=rgb2gray(y3);
y4=imread('t4.jpg'); y4=imresize(y4,[45
45],'bicubic');y4=rgb2gray(y4);
y5=imread('t5.jpg'); y5=imresize(y5,[45 45],'bicubic');
y5=rgb2gray(y5);
y6=imread('t6.jpg'); y6=imresize(y6,[45 45],'bicubic');
y6=rgb2gray(y6);
y7=imread('t7.jpg'); y7=imresize(y7,[45 45],'bicubic');
y7=rgb2gray(y7);
y8=imread('t8.jpg'); y8=imresize(y8,[45 45],'bicubic');
y8=rgb2gray(y8);
y9=imread('t9.jpg'); y9=imresize(y9,[45 45],'bicubic');
y9=rgb2gray(y9);
y10=imread('t10.jpg'); y10=imresize(y10,[45
45],'bicubic');y10=rgb2gray(y10);
y11=imread('t11.jpg'); y11=imresize(y11,[45
45],'bicubic');y11=rgb2gray(y11);
y12=imread('t12.jpg');y12=imresize(y12,[45
45],'bicubic');y12=rgb2gray(y12);
y13=imread('t13.jpg'); y13=imresize(y13,[45 45],'bicubic');
y13=rgb2gray(y13);
y14=imread('t14.jpg'); y14=imresize(y14,[45
45],'bicubic');y14=rgb2gray(y14);
y15=imread('t15.jpg'); y15=imresize(y15,[45
45],'bicubic');y15=rgb2gray(y15);
y16=imread('t16.jpg'); y16=imresize(y16,[45 45],'bicubic');
y16=rgb2gray(y16);
y17=imread('t17.jpg'); y17=imresize(y17,[45 45],'bicubic');
y17=rgb2gray(y17);
y18=imread('t18.jpg'); y18=imresize(y18,[45
45],'bicubic');y18=rgb2gray(y18);
y19=imread('t19.jpg'); y19=imresize(y19,[45
45],'bicubic');y19=rgb2gray(y19);
y20=imread('t20.jpg'); y20=imresize(y20,[45
45],'bicubic');y20=rgb2gray(y20);
y21=imread('t21.jpg'); y21=imresize(y21,[45
45],'bicubic');y21=rgb2gray(y21);
y22=imread('t22.jpg'); y22=imresize(y22,[45
45],'bicubic');y22=rgb2gray(y22);
y23=imread('t23.jpg'); y23=imresize(y23,[45
45],'bicubic');y23=rgb2gray(y23);
y24=imread('t24.jpg'); y24=imresize(y24,[45
45],'bicubic');y24=rgb2gray(y24);
y25=imread('t25.jpg'); y25=imresize(y25,[45
45],'bicubic');y25=rgb2gray(y25);
y26=imread('t26.jpg'); y26=imresize(y26,[45
45],'bicubic');y26=rgb2gray(y26);
y27=imread('t27.jpg'); y27=imresize(y27,[45
45],'bicubic');y27=rgb2gray(y27);
y28=imread('t28.jpg'); y28=imresize(y28,[45
45],'bicubic');y28=rgb2gray(y28);
y29=imread('t29.jpg'); y29=imresize(y29,[45
45],'bicubic');y29=rgb2gray(y29);
y30=imread('t30.jpg'); y30=imresize(y30,[45
45],'bicubic');y30=rgb2gray(y30);
```

```
y31=imread('t31.jpg'); y31=imresize(y31,[45
45],'bicubic');y31=rgb2gray(y31);
y32=imread('t32.jpg'); y32=imresize(y32,[45
45],'bicubic');y32=rgb2gray(y32);
y33=imread('t33.jpg'); y33=imresize(y33,[45
45],'bicubic');y33=rgb2gray(y33);
y34=imread('t34.jpg'); y34=imresize(y34,[45
45],'bicubic');y34=rgb2gray(y34);
y35=imread('t35.jpg'); y35=imresize(y35,[45
45],'bicubic');y35=rgb2gray(y35);
y36=imread('t36.jpg'); y36=imresize(y36,[45
45],'bicubic');y36=rgb2gray(y36);
y37=imread('t37.jpg'); y37=imresize(y37,[45
45],'bicubic');y37=rgb2gray(y37);
y38=imread('t38.jpg'); y38=imresize(y38,[45
45],'bicubic');y38=rgb2gray(y38);
y39=imread('t39.jpg'); y39=imresize(y39,[45
45],'bicubic');y39=rgb2gray(y39);
y40=imread('t40.jpg'); y40=imresize(y40,[45
45],'bicubic');y40=rgb2gray(y40);


aa={y1,y2,y3,y4,y5,y6,y7,y8,y9,y10,y11,y12,y13,y14,y15,y16,y17,y18,y19
,y20,y21,y22,y23,y24,y25,y26,y27,y28,y29,y30,y31,y32,y33,y34,y35,y36,y
37,y38,y39,y40};

for i=1:40
    bb(:,i)=reshape(aa{1,i(:,:)},[],1);
end
 test_pattern=[];
for j=1:4
    for i=j:4:36+j
        test_pattern=[test_pattern bb(:,i)];
    end
end

test_pattern=double(test_pattern)./255;

% P1_test = double([y1(:) y2(:) y3(:) y4(:)]/255);
% P2_test = double([ y5(:) y6(:) y7(:) y8(:)]/255);
% P3_test = double([y9(:) y10(:) y11(:) y12(:)]/255);
% P4_test = double([y13(:) y14(:) y15(:) y16(:)]/255);
% P5_test = double([y17(:) y18(:) y19(:) y20(:)]/255);
% P6_test = double([y21(:) y22(:) y23(:) y24(:)]/255);
% P7_test = double([y25(:) y26(:) y27(:) y28(:)]/255);
% P8_test = double([y29(:) y30(:) y31(:) y32(:)]/255);
% P9_test = double([y33(:) y34(:) y35(:) y36(:)]/255);
% P10_test = double([y37(:) y38(:) y39(:) y40(:)]/255);
%
%
% test_pattern=[P1_test P2_test P3_test P4_test P5_test P6_test
P7_test P8_test P8_test P9_test P10_test];
```

```
[m,n]=size(test_pattern);
%s=0;
for k=1:n
    test_pattern(:,k);
   % s=s+1;
testing = sim(net,test_pattern(:,k))
end

%cd('F:\database2_correction\Database');
```

### 3. The Third Experiment

```
clear all;
close all;
signal_number = 10;
signal_expression = 4;
PATTERNS = [];
cd('F:\SLR\Program\3rd Experiment');
x1=imread('a1.jpg'); x1=imresize(x1,[45 45],'bicubic');
x1=rgb2gray(x1);
x2=imread('a2.jpg'); x2=imresize(x2,[45
45],'bicubic');x2=rgb2gray(x2);
x3=imread('a3.jpg'); x3=imresize(x3,[45 45],'bicubic');
x3=rgb2gray(x3);
x4=imread('a4.jpg'); x4=imresize(x4,[45
45],'bicubic');x4=rgb2gray(x4);
x5=imread('a5.jpg'); x5=imresize(x5,[45 45],'bicubic');
x5=rgb2gray(x5);
x6=imread('a6.jpg'); x6=imresize(x6,[45 45],'bicubic');
x6=rgb2gray(x6);
x7=imread('a7.jpg'); x7=imresize(x7,[45 45],'bicubic');
x7=rgb2gray(x7);
x8=imread('a8.jpg'); x8=imresize(x8,[45 45],'bicubic');
x8=rgb2gray(x8);
x9=imread('a9.jpg'); x9=imresize(x9,[45 45],'bicubic');
x9=rgb2gray(x9);
x10=imread('a10.jpg'); x10=imresize(x10,[45
45],'bicubic');x10=rgb2gray(x10);
x11=imread('a11.jpg'); x11=imresize(x11,[45
45],'bicubic');x11=rgb2gray(x11);
x12=imread('a12.jpg');x12=imresize(x12,[45
45],'bicubic');x12=rgb2gray(x12);
x13=imread('a13.jpg'); x13=imresize(x13,[45 45],'bicubic');
x13=rgb2gray(x13);
x14=imread('a14.jpg'); x14=imresize(x14,[45
45],'bicubic');x14=rgb2gray(x14);
x15=imread('a15.jpg'); x15=imresize(x15,[45
45],'bicubic');x15=rgb2gray(x15);
x16=imread('a16.jpg'); x16=imresize(x16,[45 45],'bicubic');
x16=rgb2gray(x16);
x17=imread('a17.jpg'); x17=imresize(x17,[45 45],'bicubic');
x17=rgb2gray(x17);
x18=imread('a18.jpg'); x18=imresize(x18,[45
45],'bicubic');x18=rgb2gray(x18);
x19=imread('a19.jpg'); x19=imresize(x19,[45
45],'bicubic');x19=rgb2gray(x19);
x20=imread('a20.jpg'); x20=imresize(x20,[45
45],'bicubic');x20=rgb2gray(x20);
```

```
x21=imread('a21.jpg'); x21=imresize(x21,[45
45],'bicubic');x21=rgb2gray(x21);
x22=imread('a22.jpg'); x22=imresize(x22,[45
45],'bicubic');x22=rgb2gray(x22);
x23=imread('a23.jpg'); x23=imresize(x23,[45
45],'bicubic');x23=rgb2gray(x23);
x24=imread('a24.jpg'); x24=imresize(x24,[45
45],'bicubic');x24=rgb2gray(x24);
x25=imread('a25.jpg'); x25=imresize(x25,[45
45],'bicubic');x25=rgb2gray(x25);
x26=imread('a26.jpg'); x26=imresize(x26,[45
45],'bicubic');x26=rgb2gray(x26);
x27=imread('a27.jpg'); x27=imresize(x27,[45
45],'bicubic');x27=rgb2gray(x27);
x28=imread('a28.jpg'); x28=imresize(x28,[45
45],'bicubic');x28=rgb2gray(x28);
x29=imread('a29.jpg'); x29=imresize(x29,[45
45],'bicubic');x29=rgb2gray(x29);
x30=imread('a30.jpg'); x30=imresize(x30,[45
45],'bicubic');x30=rgb2gray(x30);
x31=imread('a31.jpg'); x31=imresize(x31,[45
45],'bicubic');x31=rgb2gray(x31);
x32=imread('a32.jpg'); x32=imresize(x32,[45
45],'bicubic');x32=rgb2gray(x32);
x33=imread('a33.jpg'); x33=imresize(x33,[45
45],'bicubic');x33=rgb2gray(x33);
x34=imread('a34.jpg'); x34=imresize(x34,[45
45],'bicubic');x34=rgb2gray(x34);
x35=imread('a35.jpg'); x35=imresize(x35,[45
45],'bicubic');x35=rgb2gray(x35);
x36=imread('a36.jpg'); x36=imresize(x36,[45
45],'bicubic');x36=rgb2gray(x36);
x37=imread('a37.jpg'); x37=imresize(x37,[45
45],'bicubic');x37=rgb2gray(x37);
x38=imread('a38.jpg'); x38=imresize(x38,[45
45],'bicubic');x38=rgb2gray(x38);
x39=imread('a39.jpg'); x39=imresize(x39,[45
45],'bicubic');x39=rgb2gray(x39);
x40=imread('a40.jpg'); x40=imresize(x40,[45
45],'bicubic');x40=rgb2gray(x40);

P1 = double([x1(:) x2(:) x3(:) x4(:)])/255;
P2 = double([ x5(:) x6(:) x7(:) x8(:)])/255;
P3 = double([x9(:) x10(:) x11(:) x12(:)])/255;
P4 = double([x13(:) x14(:) x15(:) x16(:)])/255;
P5 = double([x17(:) x18(:) x19(:) x20(:)])/255;
P6 = double([x21(:) x22(:) x23(:) x24(:)])/255;
P7 = double([x25(:) x26(:) x27(:) x28(:)])/255;
P8 = double([x29(:) x30(:) x31(:) x32(:)])/255;
P9 = double([x33(:) x34(:) x35(:) x36(:)])/255;
P10 = double([x37(:) x38(:) x39(:) x40(:)])/255;


patterns =[P1 P2 P3 P4 P5 P6 P7 P8 P9 P10];

%DATABASE FOR signal
 M = [1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0;...
      0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0;...
```

```
        0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0
1 0 0 0 0 0 0 0;...
        0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 1 0 0 0 0 0 0;...
        0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0;...
        0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0;...
        0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 1 0 0 0;...
        0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0;...
        0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 1 0;...
        0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1];


 [g,h]=size(patterns);
[m,h]=size(M);
% CREATING AND INITIATING THE NETWORK
net = newff(minmax(patterns),[150 10],{'tansig','logsig'},'traingdx');
%( for change *************
net = init(net);
net.LW{2,1} = net.LW{2,1}*0.1;
net.b{2} = net.b{2}*0.01;



% TRAINING THE NETWORK
net.trainParam.goal = 0.0003; % Sum-squared error goal.
net.trainParam.lr = 0.3;% Learning Rate ( for change *************
net.trainParam.show = 10; % Frequency of progress displays (in
epochs).
net.trainParam.epochs = 100000; % Maximum number of epochs to train.
net.trainParam.mc = 0.5; % Momentum Factor.( for change *************

[net,tr] = train(net,patterns,M);


% Train Results
for k=1:h
    patterns(:,k);
signal = sim(net,patterns(:,k))
end;

pause; pause;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%
signal_number = 10;
signal_expression = 4;
test_patterns = [];
Thresh = 0.8;

y1=imread('t1.jpg'); y1=imresize(y1,[45 45],'bicubic');
y1=rgb2gray(y1);
y2=imread('t2.jpg'); y2=imresize(y2,[45
45],'bicubic');y2=rgb2gray(y2);
```

```
y3=imread('t3.jpg'); y3=imresize(y3,[45 45],'bicubic');
y3=rgb2gray(y3);
y4=imread('t4.jpg'); y4=imresize(y4,[45
45],'bicubic');y4=rgb2gray(y4);
y5=imread('t5.jpg'); y5=imresize(y5,[45 45],'bicubic');
y5=rgb2gray(y5);
y6=imread('t6.jpg'); y6=imresize(y6,[45 45],'bicubic');
y6=rgb2gray(y6);
y7=imread('t7.jpg'); y7=imresize(y7,[45 45],'bicubic');
y7=rgb2gray(y7);
y8=imread('t8.jpg'); y8=imresize(y8,[45 45],'bicubic');
y8=rgb2gray(y8);
y9=imread('t9.jpg'); y9=imresize(y9,[45 45],'bicubic');
y9=rgb2gray(y9);
y10=imread('t10.jpg'); y10=imresize(y10,[45
45],'bicubic');y10=rgb2gray(y10);
y11=imread('t11.jpg'); y11=imresize(y11,[45
45],'bicubic');y11=rgb2gray(y11);
y12=imread('t12.jpg');y12=imresize(y12,[45
45],'bicubic');y12=rgb2gray(y12);
y13=imread('t13.jpg'); y13=imresize(y13,[45 45],'bicubic');
y13=rgb2gray(y13);
y14=imread('t14.jpg'); y14=imresize(y14,[45
45],'bicubic');y14=rgb2gray(y14);
y15=imread('t15.jpg'); y15=imresize(y15,[45
45],'bicubic');y15=rgb2gray(y15);
y16=imread('t16.jpg'); y16=imresize(y16,[45 45],'bicubic');
y16=rgb2gray(y16);
y17=imread('t17.jpg'); y17=imresize(y17,[45 45],'bicubic');
y17=rgb2gray(y17);
y18=imread('t18.jpg'); y18=imresize(y18,[45
45],'bicubic');y18=rgb2gray(y18);
y19=imread('t19.jpg'); y19=imresize(y19,[45
45],'bicubic');y19=rgb2gray(y19);
y20=imread('t20.jpg'); y20=imresize(y20,[45
45],'bicubic');y20=rgb2gray(y20);
y21=imread('t21.jpg'); y21=imresize(y21,[45
45],'bicubic');y21=rgb2gray(y21);
y22=imread('t22.jpg'); y22=imresize(y22,[45
45],'bicubic');y22=rgb2gray(y22);
y23=imread('t23.jpg'); y23=imresize(y23,[45
45],'bicubic');y23=rgb2gray(y23);
y24=imread('t24.jpg'); y24=imresize(y24,[45
45],'bicubic');y24=rgb2gray(y24);
y25=imread('t25.jpg'); y25=imresize(y25,[45
45],'bicubic');y25=rgb2gray(y25);
y26=imread('t26.jpg'); y26=imresize(y26,[45
45],'bicubic');y26=rgb2gray(y26);
y27=imread('t27.jpg'); y27=imresize(y27,[45
45],'bicubic');y27=rgb2gray(y27);
y28=imread('t28.jpg'); y28=imresize(y28,[45
45],'bicubic');y28=rgb2gray(y28);
y29=imread('t29.jpg'); y29=imresize(y29,[45
45],'bicubic');y29=rgb2gray(y29);
y30=imread('t30.jpg'); y30=imresize(y30,[45
45],'bicubic');y30=rgb2gray(y30);
y31=imread('t31.jpg'); y31=imresize(y31,[45
45],'bicubic');y31=rgb2gray(y31);
y32=imread('t32.jpg'); y32=imresize(y32,[45
45],'bicubic');y32=rgb2gray(y32);
```

```
y33=imread('t33.jpg'); y33=imresize(y33,[45
45],'bicubic');y33=rgb2gray(y33);
y34=imread('t34.jpg'); y34=imresize(y34,[45
45],'bicubic');y34=rgb2gray(y34);
y35=imread('t35.jpg'); y35=imresize(y35,[45
45],'bicubic');y35=rgb2gray(y35);
y36=imread('t36.jpg'); y36=imresize(y36,[45
45],'bicubic');y36=rgb2gray(y36);
y37=imread('t37.jpg'); y37=imresize(y37,[45
45],'bicubic');y37=rgb2gray(y37);
y38=imread('t38.jpg'); y38=imresize(y38,[45
45],'bicubic');y38=rgb2gray(y38);
y39=imread('t39.jpg'); y39=imresize(y39,[45
45],'bicubic');y39=rgb2gray(y39);
y40=imread('t40.jpg'); y40=imresize(y40,[45
45],'bicubic');y40=rgb2gray(y40);



P1_test = double([y1(:) y2(:) y3(:) y4(:)])/255;
P2_test = double([ y5(:) y6(:) y7(:) y8(:)])/255;
P3_test = double([y9(:) y10(:) y11(:) y12(:)])/255;
P4_test = double([y13(:) y14(:) y15(:) y16(:)])/255;
P5_test = double([y17(:) y18(:) y19(:) y20(:)])/255;
P6_test = double([y21(:) y22(:) y23(:) y24(:)])/255;
P7_test = double([y25(:) y26(:) y27(:) y28(:)])/255;
P8_test = double([y29(:) y30(:) y31(:) y32(:)])/255;
P9_test = double([y33(:) y34(:) y35(:) y36(:)])/255;
P10_test = double([y37(:) y38(:) y39(:) y40(:)])/255;


test_pattern=[P1_test P2_test P3_test P4_test P5_test P6_test P7_test
P8_test P8_test P9_test P10_test];


[m,n]=size(test_pattern);
%s=0;
for k=1:n
    test_pattern(:,k);
   % s=s+1;
diseasestest = sim(net,test_pattern(:,k))
end

cd('F:\SLR\Program\3rd Experiment');
```