ARTIFICIAL NEURAL NETWORK BASED HUMAN EAR RECOGNITION

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

By
MOHAMMED AL-SAYAM

In Partial Fulfilment of the Requirements for The Degree of Master of Science in Electrical and Electronic Engineering

NICOSIA, 2017

MOHAMMED AL-SAYAM

ARTIFICIAL NEURAL NETWORK BASED HUMAN
EAR RECOGNITION

NEU
2017

ARTIFICIAL NEURAL NETWORK BASED HUMAN EAR RECOGNITION

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

By
MOHAMMED AL-SAYAM

In Partial Fulfilment of the Requirements for
The Degree of Master of Science
in
Electrical and Electronic Engineering

NICOSIA, 2017

MOHAMMED AL-SAYAM: ARTIFICIAL NEURAL NETWORK BASED HUMAN EAR RECOGNITION

**Approval of Director of Graduate School
of
Applied
Sciences**

Prof. Dr. Nadire ÇAVUŞ

We certify this thesis is satisfactory for the award of the degree of Masters of Science in Electrical and Electronic Engineering

Examining Committee in Charge:

I hereby declare that all information in this document has 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: MOHAMMED AL-SAYAM

Signature:

Date:

ACKNOWLEDGMENTS

I would like to express my sincere appreciation and thanks to my supervisor, Assist. Prof. Dr. Kamil Dimililer , for his guidance and mentorship during my graduate studies. His impressive knowledge and creative thinking have been source of inspiration throughout this work. My deepest gratitude goes to my parents, my wife, my brothers, sisters, and my children, to whom I am most indebted. I thank them for constant love, prayers, patience and support while I was studying abroad. I know I can never come close to returning their favour upon me. I will always be thankful to my friends and colleagues for their unlimited support. I extend my thanks to all the Libyan community that gave me a second family away from home.

To my parents and family….

ABSTRACT

Ear recognition is a new and recent subject in the field of biometrics. Ear is one of different physical features of the humans that are unique and can't be identical for two individuals. It is recently drawing more much researcher's attention due to its stability during different stages of the life. Artificial neural networks have also drawn a very important development in the last decades. It has become of the first artificial intelligence approaches in the science. This work proposes the use of back propagation algorithm in the learning of artificial neural network for ear recognition. The data base composed of the ears' images of 99 persons will be processed and used in the training of the network. Different image processing techniques will be applied on the images before being fed to the network in order to reduce the noise and size of images while keeping main features of the original images. The results of the proposed work will be then tabulated and discussed.

***Keywords*:** Artificial neural networks; back propagation; biometrics; database; ear recognition

ÖZET

Kulak Tanıma, biyometrinin, yeni ve güncel bir konusudur. Kulak, insanın benzersiz bir fiziksel özelliği olup, iki farkli kişide ayni şekilde bulunması mumkun olmadığından ve İnsan hayatının farklı evrelerinde, kulak şekli değişmediğinden dolayı son yıllarda araştırmacıların ilgisini çekmektedir. Yapay Sinir Ağları son on yılda ciddi gelişme göstermiş ve yapay zekadaki ilk yaklasımlardan biri haline gelmiştir. Bu tezde kulak tanımlamak için yapay sinir ağlarında geri yayılım algoritması önerilmiştir. Veri tabani 99 farklı kişiden oluşup ağın eğitilmesi için kullanılmıştır. Ağa iletilmeden önce farklı resim işleme teknikleri uygulanıp, resimin boyutu küçültülmüş ve gürültü azaltılmıştır. Bunun yanında orjinal resimin ana karakteristikleri korunmuştur. Önerilen işin sonuçları tablolaştırılıp tartışılmıştır

Anahtar Kelimeler: Yapay sinir ağları; geri yayılım; biometri; veri tabanı; kulak tanıma

TABLE OF CONTENTS

CHAPTER 4: ARTIFICIAL NEURAL NETWORKS

CHAPTER 5: RESULTS AND DISCUSSIONS

CHAPTER 6: CONCLUSIONS AND FUTURE WORKS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ANN: Artificial Neural Networks

AI: Artificial Intelligence

BP: Back Propagation

MLP: Multi-Layer Perceptron

RGB: Red, Green, and Blue

SLP: Single Layer Perceptron

CHAPTER 1

INTRODUCTION

1.1 Overview

Ear is one of the main biometric features of human body besides face, fingerprint, eye print. Biometrics is the science of studying different means to find physical or activities traits of a person that can be used to recognize or identify him. Different biometrics using face identification, eye print, fingertips print, in addition to vocal recognition is commonly used. These features are easier to measure than behavioural features such as handwriting and printing and signature features.

The interest in biometrics has recently become extremely high due to the development in digital processing systems and the increasing needs for security in different fields. The researches on biometrics are occupying a more and more of researcher's attention; scientists are also concentrating and focussing on biometrics due to its high importance. Biometrics is a feature key in security and safety systems and it is in the heart of the increasing requirements for secure and effortless acknowledgment processes. Different types of biometrics can be grouped into two main categories; these are physical biometric features such as the shape and features of the face, eye, ear, finger print etc. The second category is the behaviour based biometric features such as writing method, voice tunes, walking rhythm, and other much behaviour of humans. These different biometric categories are most used in security systems, medical sections, military services, in addition to presence systems in huge companies.

The use of long identification words with passwords and special cards is a great dispute for people with memory difficulties or those people whose work implies the interaction with dozens of passwords for different systems. Bank accounts and credit cards identification also are of the important needs for biometric identification systems for higher security levels. Modern computers are also equipped with special recognition systems based on fingerprint and eye print biometrics to avoid being misused by the wrong person. The development in digital technology has promoted scientist to carry on advanced steps in biometrics. The use of digital biometric recognition systems is nowadays an everyday fact.

Ear is one of the most important features of the human biometrics that has its own characteristics. The prospective for using the manifestation of the human ear as a person recognition method was known and promoted early in the 19th century by the French crime expert Alphonse Bertillon (Saleh, 2007). Since that time, ear identification has been presented and implemented in hundreds of scientific articles and papers. The human ear is known for its stability over the long period of time. Changes in the exterior features of the human ear are unexpected for long periods of time or even all over the life. Some scientists also claimed that ear is a unique feature of the human that can't be repeated and thus it can be used to identify exactly the identity of the person (Yang, 2006).

In order to use ear as biometric the system must start by collecting ear image data and process these images. The processing is yet done manually to localize and normalize and image of the ear. The recognition of the person or ear is the second phase of the process where persons are identified using the treated ear's images. The pre-processing of the images is very important just like the recognition actions; however, it is less studied in literature (Alastahir, 2009).

Artificial intelligence is a very famous science that is applied in different areas of science and lie. This type of intelligence uses mathematical models and relations to reproduce and imitate the brain structure and function. Artificial neural networks are being widely implemented in many industrial and scientific processes. They are widely used in IRIS recognition, face recognition, signature recognition, and finger print recognition. Many scientific papers have presented the use of neural networks in medical image processing also for disease identification or classification.

Neural networks are parallel computing elements grouped in a system. This system contains a huge number of simple interconnected processing units. These units are called weights and they are subject to changes during the process of training. The training is a process that is similar to the learning in human brain. The principle of neural networks is used to simulate a process by looking at examples and finding a pattern or common sense between these examples. The weights in a neural network are slowly updated and adjusted with respect to some criterions that ensure the convergence toward a given target. The desire is that when the neural network is given a new group of input variables, it will generate a correct output. The process of training a network to do a required job involves the investigation of sets of inputs and outputs.

The values and interconnections between weights of the network are updated in an iterative form till reaching an optimum functional point that satisfies the input output relationship.

This work studies the implementation of artificial neural networks and back propagation learning algorithm in biometric systems based on ear recognition. The ear was chosen for its stable structure and unique features to be used in this work. Ear's images of 99 persons will be used and treated in this work to experiment the power of neural networks in person identification and classification.

1.2 Literature Review

Ear recognition is no more a new subject in the field of science and person identification. It has been introduced and implemented many decades ago. Various features extraction and identification algorithms have been studied and implemented in papers and articles based on ear features. Scientists are in the debate about the uniqueness of the human ear. However, the most popular opinion is in the support of this theory. The use of neural networks for recognition based on ear images has been introduced in (El-bakri, 2007). The author has presented the basics of using ear for person identification. Three-dimensional image recognition using special methods for analyzing images was introduced. The investment of ear features for person authentication using geometrical structures was presented in (Rahman, 2007). A new ear pre-processing algorithm based on the image ray conversion was presented in (Alastair, 2009). This conversion is able to focus on some tubular features like the helix of the ear and spectacle frames; they can be used as features of preparing images for recognition and pre-processing. In (Singh and Singla, 2013), the author has discussed the idea of biometrics with focus on ear recognition as biometric. Segmentation of ear images preparing for the application of different recognition techniques was proposed in (Saleh, 2007). A multistage geometric conversion of ear images was presented in (Shailaja, 2006). This conversion prepares for easier recognition of ear as biometric. In (Daishi et al., 2008), a new multi modal biometrics based on face and ear recognition approach was presented and discussed. This implies more confident and secure system recognition as it is based on multi modal instead of using one feature in person recognition. Assessment of ear identification methods has been presented in (Victor, Bowyer, and Sarkar, 2000). Chora (2007) has introduced a survey about the extraction methods of ear features for recognition purposes.

The implementation of wavelet conversion algorithm for ear identification was the subject discussed in (Ibrahim and Ibrahim, 2012). A new called “banana wavelet” transform was mentioned and applied. The performance of this method was discussed and experimented. Purkait, (2007) has presented a short discussion about the subject of ear recognition in person identification. Yang, (2006) and (Prakash, 2013) have presented a technique for processing two and three dimensional images that are going to be used in ear recognition.

Multimodal recognition was also repeated in different papers and discussed as a high performance recognition method. Fingerprint and ear recognition for personal recognition was studied in (Kasprowski, 2005). A technique of edge interaction point detection was presented to resolve the features of the ear; while the line connected components were used to extract the finger print features. Finally, a neural network was used for identity recognition system that is based on back propagation algorithm.

The subject of ear recognition and the use of ear as biometric feature have been discussed in thousands of researches that can’t be covered in this work. Person identification systems that use ear images separately or in multimodal structures are proposed widely in this field.

The use of neural networks for recognition purposes has been proposed in a lot of literature references. Its simple arrangement and performance in addition to its ease of use compared to the other methods that need a lot of manual processing have put it in the first of recognition methods. This work discusses the implementation of neural networks in ear recognition that can be used for different purposes in the future.

CHAPTER 2

PROPOSED WORK

This thesis work was intended for the study of the efficiency of the use of artificial neural networks in the human recognition and biometrics. The biometric that was proposed in this thesis is the ear. The proposed work relies on the assumption that the ear of a human is stable all over long periods of his age. Another assumption that was accepted in this work is that the human ear shape is unique and can’t be repeated for two different persons. The proposed work will use the human ear shape and features in the training and test of the back propagation learning algorithm of ANN. Different samples of images will be collected from different people and arranged such that they can be fed to the ANN structure. The images need to be treated in an early stage prior to the use of neural networks such that they can be easily fed to the neural structure. The treatment of images will include capturing, sizing, filtering, scale conversion, pattern averaging, and features detection of the images. All these processes will be applied using MATLAB software and will be adjusted automatically using functions of MATLAB.

2.1 Contribution of the Proposed Work

This thesis work is a small contribution in the world researches on biometrics and in the field of neural networks. It is a part of continuous researches on the subject of ear recognition and biometric signs. It aims to increase the security options available in different fields of life and science. The work is prepared and done in two stages of processing. The first stage is the image processing before the use of artificial neural networks; the second phase is the training and recognition using the artificial neural network systems.

2.2 Aim of the Thesis

The aim of the thesis is to contribute in the continuous race of international security. It aims to find and improve new biometric security methods to be involved in the future security applications in military and civil life.

2.3 Plan of the Proposed Work

The work in this thesis is basically pointed toward the study of the structure of the neural network and its strength in the biometrics. The ear biometric is the main feature that is going to be used. In order to be able to apply the required study, different steps must be followed before starting the neural network functions. Figure 2.1 presents a work plan that summarizes the proposed work in brief description.

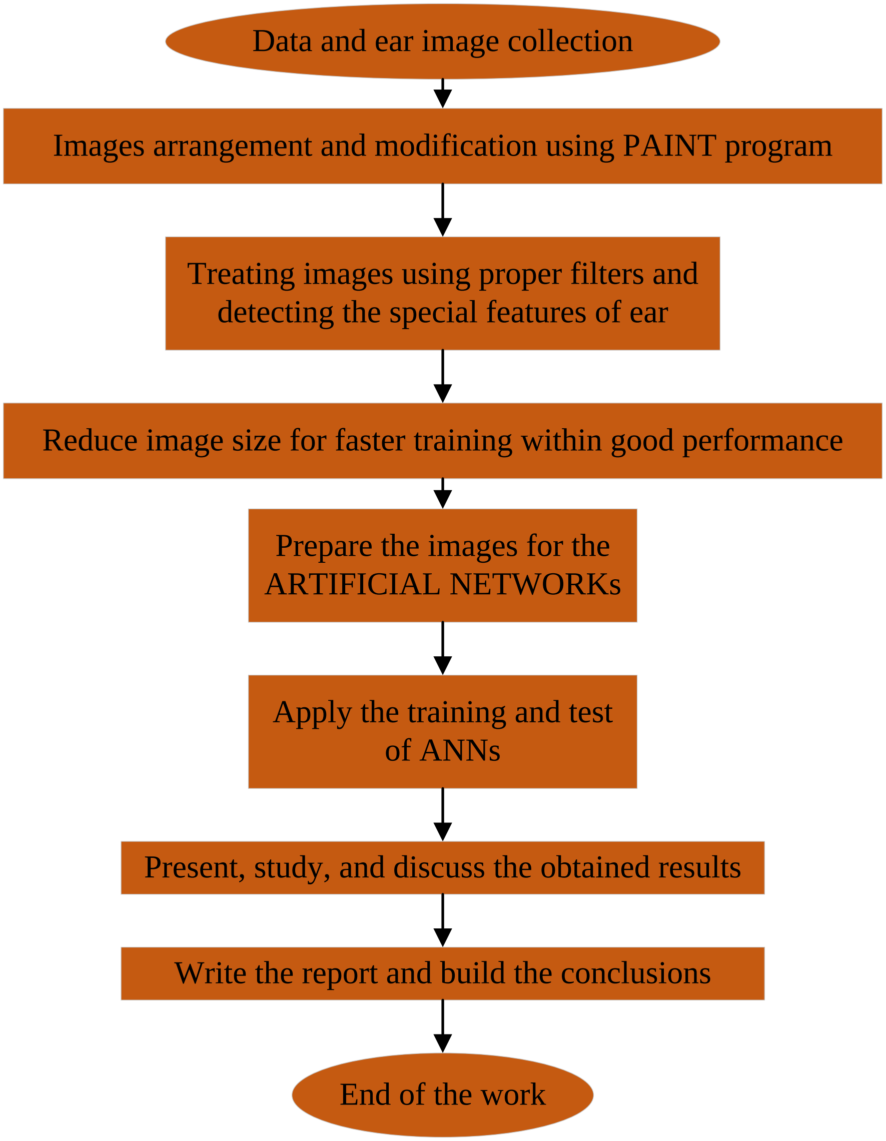

Data and ear image collection
Images arrangement and modification using PAINT program
Treating images using proper filters and detecting the special features of ear
Reduce image size for faster training within good performance
Prepare the images for the ARTIFICIAL NETWORKs
Apply the training and test of ANNs
Present, study, and discuss the obtained results
Write the report and build the conclusions
End of the work

Figure 2.1: Plan of the proposed work

CHAPTER 3

IMAGE PROCESSING TECHNIQUES

3.1 Image Processing

The image processing is the process of converting an image to digital form and to apply different mathematical operations to modify the structure of images. Image processing is a form of signal processing that deals with multi-dimensional arrays of images. The input for the digital image processor can be an image or a frame in video streaming. Generally, image processing includes treating images in three separate steps: (Image Processing Applications , 2011):

1- Import the images using scanner or camera.

2- Analyse and treat the image using different techniques like image compression or image enhancement.

3- Generate the output image at the last step of the image processing.

The aim for implementing image processing on images can be actually resumed in the next few reasons. It helps also in the visualization and observation of images that are not clear or to make visible some objects that are not visible. Image processing in this case is useful to increase the clearness of the images. Sharpening and restoring images to make them clear and increase the details in the image. It creates better images than the originals.

Pattern measurement and recognition where some patterns in the images can be found and extracted to be used later for different purposes. It can be also used to identify or distinguish images in different applications. In addition to retrieval of images; where an image can be retrieved from a group of images to be used later.

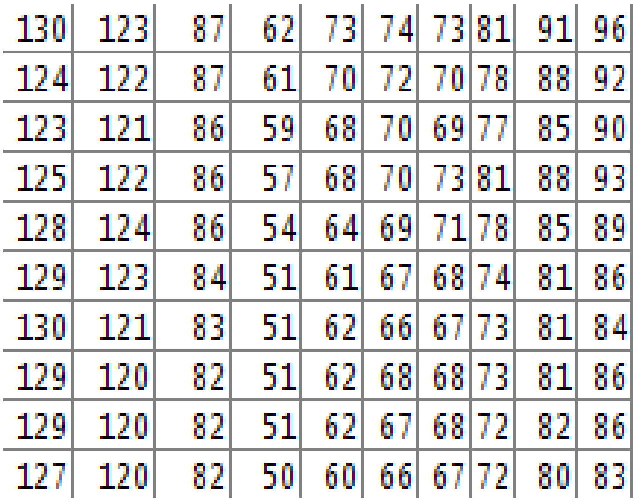
Generally there are two types of image processing. These are analogue image processing and digital image processing. Analogue image processing or it can also be called visual image processing is used with hardcopies of images such as photographs or printed papers. The analyser implements different techniques to analyse an image based on his knowledge in addition to the experience. Analyser uses association also while doing analogue image processing. In the digital image processing, the images are easily and very widely manipulated and interpreted using digital techniques using computer software. Raw image acquired from digital cameras or through communication systems are mostly defected and visually degraded. The use of digital processing techniques is compulsory to obtain better quality of the raw images after being acquired or transmitted through transmission channels. These processing techniques are able to get the original information of the image and to increase the visibility of that original image.

The development in personal computers and processors technology has increased the use of image processing software. Nowadays, image processing software are widely implemented and one can do any required modification on any image with simple click.

3.2 Image and Pixel

The image can be mathematically seen as a matrix of numbers that contains a number of lines and columns. Figure 3.1 shows an image that has 30x30 pixels in it in addition to the numerical values associated to each one of the pixels.

50x50 pixel



130	123	87	62	73	74	73	81	91	96
124	122	87	61	70	72	70	78	88	92
123	121	86	59	68	70	69	77	85	90
125	122	86	57	68	70	73	81	88	93
128	124	86	54	64	69	71	78	85	89
129	123	84	51	61	67	68	74	81	86
130	121	83	51	62	66	67	73	81	84
129	120	82	51	62	68	68	73	81	86
129	120	82	51	62	67	68	72	82	86
127	120	82	50	60	66	67	72	80	83

First 10x10 pixel values

Figure 3.1: Image versus pixel representation of the image

The pixel is the tinniest particle in an image. Any image is constructed of a set of pixels arranged to construct the image. The pixel in an image represents the colour density at that point represented in mathematical number. This number can be 8bits or 16bits or even more. In an 8 bits gray images; the pixel values can range between 0 and 255. Each number specifies a value of the concentration of the black colour in the image. It specifies the energy of the light photon that hits the point of the pixel at that time.

3.2.1 Total pixel's number

An image is generally specified as a two dimensional framework or in some images three connected two dimensional frames. The image is composed of a number of rows and number of columns. The total number of pixels is the equal to the number of rows multiplies the number of columns.

3.2.2 Grey level of image

The value of the pixel at a moment is a measure of the power of the image at the area of that pixel. Grey level shows the brightness of the pixel in the image. The minimum value of the grey level is set 0 while the maximum value is dependent on the bits of digitization of the image. More simply, in an 8 bit image the maximum grey level is equal to 2^8-1=255. In binary images, the value of the bit can be either 0 or 255. In colour image, the value of grey level can be found based on the formula:

$$Grey\,level = 0.299*R + 0.587*G + 0.114*B \tag{3.1}$$

Where; the components R, G, and B represent the concentration of red, green, and blue colours respectively. This formula is set with careful consideration of the human eye's sensitivity of colours. This makes the grey level value totally independent of colour and based only on the way the human eye can see it (Zollitch, 2016).

3.3 Methodology of the Work

As for the moment, the proposed system consists of two phases. These are the processing phase and the identification phase based on the processed images. The processed images are being identified and separated based on the extracted patterns by using the neural network identification system. In the first phase, images are going to be processed using different image processing techniques. All these techniques will be explained and presented in details throughout this chapter work. The used techniques will include but not limited to conversion from RGB images to grey scale images, Canny edge segmentation, filtering the noise using median and Weiner filtering methods, averaging the image pixels to reduce the image size without feature losses. All the used techniques are used to achieve at least one of three main goals. These goals are: simplification of ANN process by size reduction, image quality enhancement, and extracting special features from the images. At the end of the processing phase of the images, these processed images are ready to be submitted to the next stage of neural network where they are going to be treated and identified in other different phases.

3.4 Used Image Processing Techniques

The image processing techniques that were used in this work can be resumed in the next few steps:

1- Reading RGB images from the folder.
2- Converting RGB images to grey images.
3- Images smoothening using simple median filter.
4- Filtering the images using Weiner filter.
5- Segmenting the images using canny edge detector.
6- Extracting the patterns and specific features by averaging image pixels.

Next to all these processing steps comes the step of image identification in the neural network identifier. This step starts by arranging the images in a way that simplifies the treatment of the images in the neural networks processor. The process is divided into two basic stages that are the training stage and the test stage. In the training stage, some of the images are used as examples for the network to be used and the network is being trained to recognize perfectly these images. The neural network is able to extract and store the common patterns in the images in order to use them in future comparisons with other images. After the training of the neural network, some images that were not shown before to the network are presented to verify whether the network has learned or not. Figure 3.2 illustrates the flow chart of the two stages of the work.

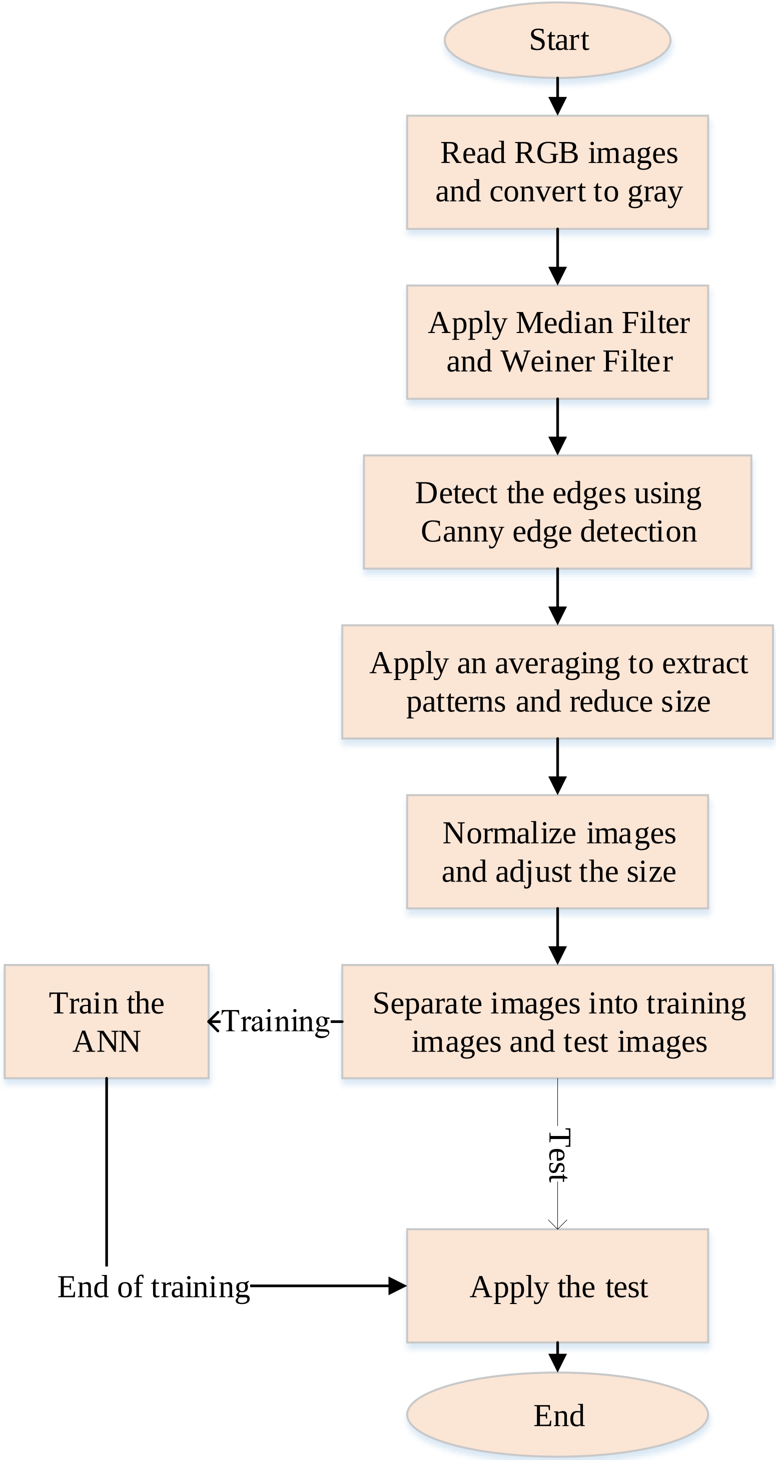

Start
Read RGB images and convert to gray
Apply Median Filter and Weiner Filter
Detect the edges using Canny edge detection
Apply an averaging to extract patterns and reduce size
Normalize images and adjust the size
Separate images into training images and test images
Training
Train the ANN
Test
End of training
Apply the test
End

Figure 3.2: Flow chart of the used recognition process

3.5 Image Processing Methods

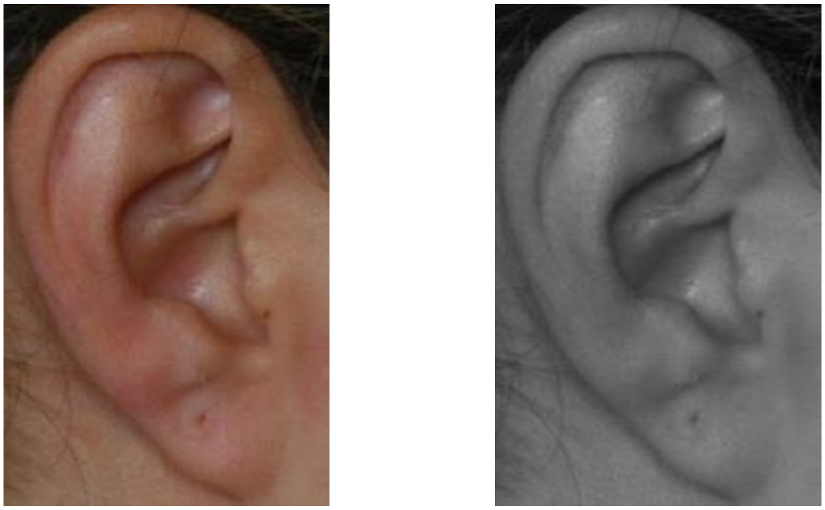
The used ear images were treated and ameliorated for in order to improve the efficiency of the identification process using different methods. The used methods construct the basis for high quality and low processing costs for the ear based identification system.

3.5.1 RGB to gray scale image conversion

The coloured images are constructed out of three basic colours; these are the Red, Green, and blue. Each RGB image is described based on a three matrices containing the light intensity of each one of the colours. The treatment of an RGB image implies the treatment of the three matrices during all the processes. This implies increasing the processing and neural training costs three times more than the case of a simple gray scale image. In order to reduce the processing stresses from the system, it is considered a good idea to convert the RGB images into gray scale images. Gray scale images contain the same information of the RGB images except from the colour details that are more useful for human vision. This process is like an evaluation of the image framework. The conversion is based on the sensitivity of the human eye that subtle to green colour more than the other colours (Zollitch, 2016). For that reason, in the conversion equation the green colour is equipped with higher weight than the other two colours as shown below:

$$Grey\,level = 0.299 * R + 0.587 * G + 0.114 * B \tag{3.2}$$

The given weight is assumed based on the contribution of each one of the three colours in the human vision. Based on this assumption, the gray scale image is bright and clearer as it is based on the colour contribution rather than a simple averaging process. Figure 3.3 presents a comparison between the RGB image and its corresponding gray scale converted image.



(a) RGB image (b) Gray scale image

Figure 3.3: The RGB image vs. the gray scale converted image

3.5.2 Median filtering of the image

Median filtering is an image processing method implemented to diminish and eliminate the unwanted noise an image. It creates fewer pixels concentrated and clear image. The idea here is to eliminate all the pixels that have very variable values from a group of pixels. Most smoothing techniques are based on low pass linear filters. It relies mostly on an averaging process of the image or in other cases on the median value process. In order to make smooth our image, a filter shall be applied on the treated image. The most common and easy to use filter type is the median filter that we propose in our work. It is implemented to eliminate any impulsive type noise of the image while keeping the main features of the image unchanged.

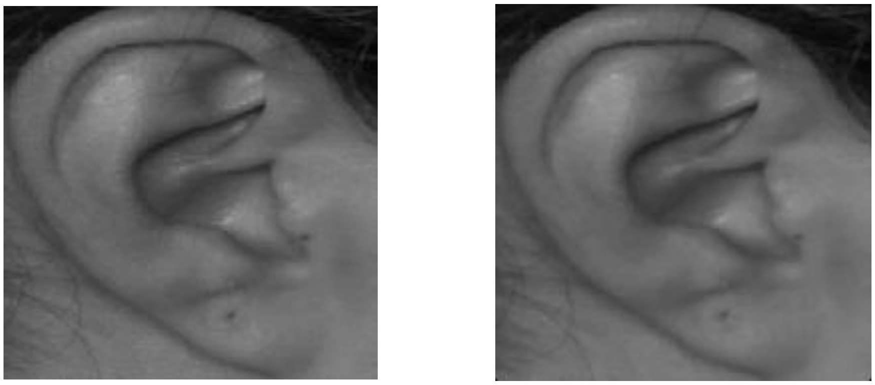
The process of median filtering is a simple process in which the median value of a chosen window around the studied pixel is found. The found value then replaces the studied pixel to eliminate any expected pulses in the image. This way, each pixel of the image is treated carefully and replaced by the median of the surrounding elements located in a square kernel. This way, median filter detects the noise and eliminates it while keeping the important sharp details of the image. Figure 3.4 shows the idea of applying median filter on a small matrix of numbers. It is obvious that the value 40 in the original matrix disappeared after the median filtering because it is very far from the neighbouring values. Figure 3.5 presents the median filtering of a 256x256 gray scale ear image.

24	23	21	20	18	17	16	17
40	24	24	23	21	19	18	18
21	23	35	24	22	20	18	20
18	20	22	22	20	18	16	21
17	18	19	18	17	16	15	20
18	17	17	16	16	15	15	17
19	18	16	15	15	15	16	14
17	16	16	15	14	13	13	11

median →

24	24	21	20	18	17	17	17
23	24	23	22	20	18	18	18
21	23	23	22	21	19	18	19
18	20	22	22	20	18	18	20
18	18	18	18	17	16	16	17
18	18	17	16	16	15	15	16
18	17	16	16	15	15	15	14
19	18	16	16	15	15	15	14

Figure 3.4: Application of media filter on a window of 3x3



a) Original Image

b) Median filtered image

Figure 3.5: Median filtering of a 256x256 image of the ear

3.5.3 Canny edge detection

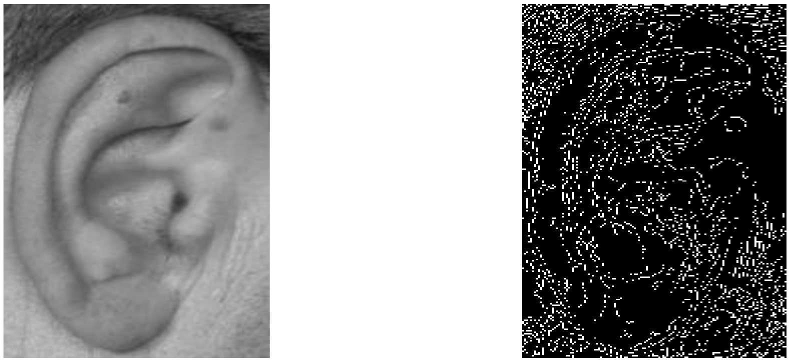
Canny edge detection or segmentation is the separation of image parts into groups. The meaning of this operation is to highlight some important areas or parts of an image. These areas are needed for processing tasks such as faces, tumors, and ear edges. In separate definition, segmentation is the assembly of important features of the considered image into important regions and less important regions. The unwanted zones can be easily then deleted by using some threshold techniques. Threshold technique uses the simple logic of eliminating the pixel value if it is less than a given value; And giving it a true value “1” if it is greater than that given value.

The edges in the images are generally defined in zones where sharp changes in pixels are existent. The process of identifying the hard contrast changes in the image is called the edge detection. There are different operators for the edge detection such as Sobel edge detection and Prewitt edge detection. In these two techniques a 3*3 kernel are used to be convolved with the image pixels to find the derivatives approximations of the lines and columns of the image. However, Canny detector is the most commonly used edge detection technique. This operator is considered to maximize the SNR (signal to noise ratio) and to accurately define the edges in the required image. Canny edge detection passes by different steps as follow:

Firstly, the image is been smoothened to de-blur it or remove the noise. This process is ensured using a Gaussian filter that is convolved with the image. Next step involves the derivation of the gradient of each line and column of the image. The gradient is higher whenever the changes in pixels horizontally and vertically are sharp. This gradient gets less if the changes are smaller. Thus, finding the edges is an approximation based on the gradient values. The smoothed image is convolved generally with the Gaussian filter derivative in two dimensions. The next formula can be used to find the gradient of a pixel in an image.

$$G_{xy} = \sqrt{G_x^2 + G_y^2} \tag{3.3}$$

Where the x and y indexes are pointing to the gradients found using Canny mask in the direction of x and y. the third step is to threshold the resulting pixels in order to find the potential edges and eliminate the weak ones. Threshold technique in Canny detection uses double threshold instead of single threshold. The two thresholds are seen as a hysteresis where values over the upper limits are considered directly as edges. Finally, the values that are not associated to very strong edge are eliminated to finish the edge detection process. The next Figure demonstrates the results of Canny edge detection of an ear image.



a) Gray scale image b) Segmented image

Figure 3.6: Canny edge detection of ear images

As it can be seen from the Figure above, it can be noticed how the treated image contains the edges and the other non important details are cleared or replaced by black zones or zeros. After the end of edge detection of the image, all unwanted components should be cleared and the image becomes clear (Shen & Tang, 2012). Canny edge detection is very useful and it has a very high performance in noise suppression, it is also very accurate in terms of edge detection, it is also more accurate and improved over the Sobel detection method (Helwan, 2014). In Figure 3.7, we can see the effect of using different threshold value on the accuracy and clearance of Canny edge detection or segmentation. Higher threshold like the one used in this Figure offers less features and reduces unwanted details.

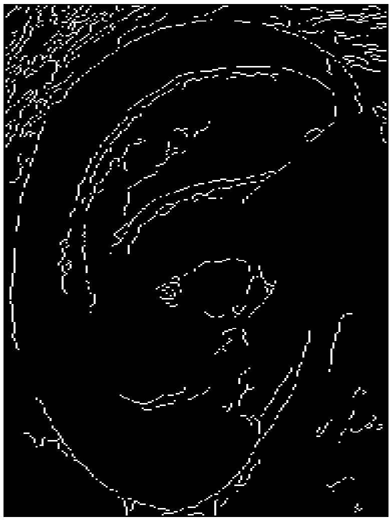


Figure 3.7: Canny edge detection with different threshold values

3.5.4 Segmentation by averaging of image features

The average is a well known term in mathematics and statistics. The average of a set of values is calculated by finding the mean value between all these values. Generally, the average of the values is calculated to create a description of the values with the minimum size possible. In image processing, averaging is used to reduce the image size with keeping all the main features without changes. A window of defined size is used to reduce the size of the processed image. The new image will contain less number of pixels where each pixel is the average of the content of the averaging window. The window moves horizontally and vertically over the image to find the new resized image. The scaled image is smaller in size but contains the same features that of the original image before scaling.

Averaging is very useful for the use with the artificial neural network systems due to the complexity and heavy processing load of ANN. A reduced size of the processed images will provide fast performance of the system. Thus, the use of average becomes indispensable with the ANN structures.

The idea of averaging process is explained by Figure 3.8, where the image matrix is divided into windows. Each one of these windows contains certain number of pixels. The average of all the pixels contained in the window is calculated and saved as one pixel in the processed image. This way each pixel of the non-processed image is considered and the information contained in that pixel is used in the construction of the processed image. At the same time the size of the processed image is reduced enough to be fed to the neural network. Figure 3.9 presents an original size image along with two averaged images with different size. It is clear from the comparison of three images in the Figure that the main features of the image remain constant while just the density of the image is reduced.

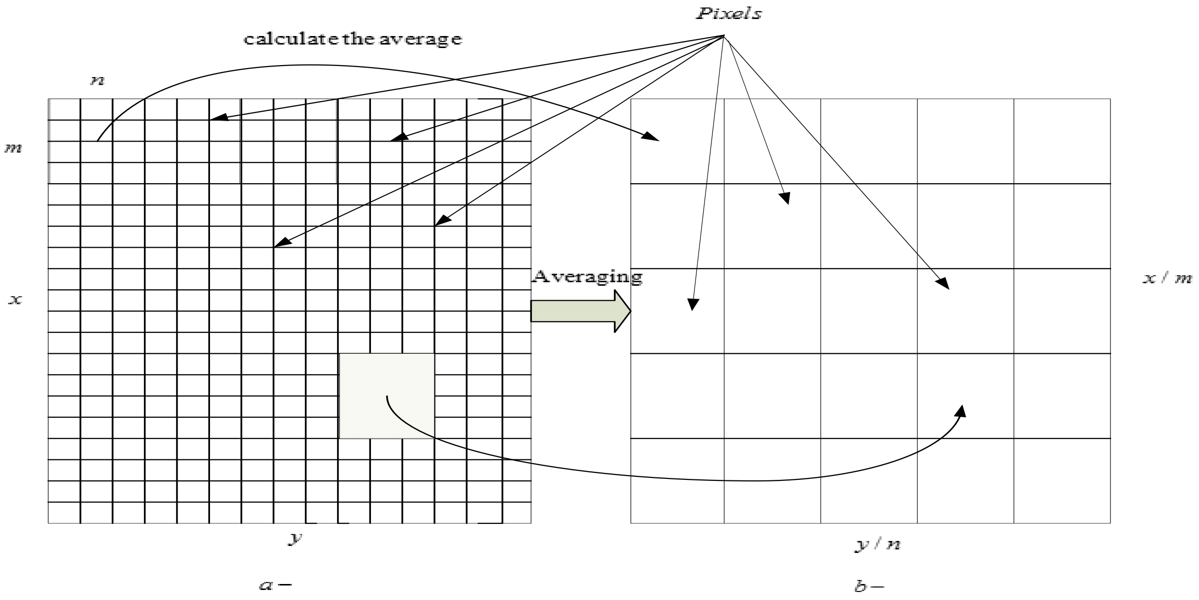
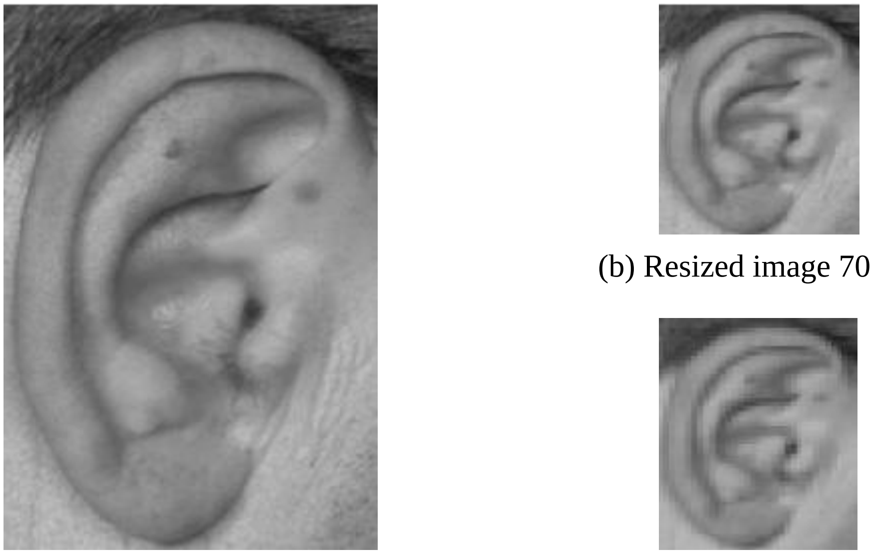


Figure 3.8: Averaging process structure



(b) Resized image 70*50

(c) Resized image 50*35

(a) Original Image

Figure 3.9: Original image versus averaged images of the ear

CHAPTER 4

ARTIFICIAL NEURAL NETWORKS

4.1 Overview of Artificial Neural Networks

The neural network was originated earlier in the last century. Scientists of that period were trying to find an artificial similitude of the biological brain that is able to perform complex tasks like recognition and learning. Actually, these ideas were not new ideas but they were mentioned in the literature of great thinkers like Aristole, Plato and others. The first publication claiming the idea of neural networks in the modern science was published in the 1940s and written by McCuloch and Pitts. It was a simple neuron application that can generate binary signal. Just in the next few years, the idea attracted many different researchers to start working on neural networks. One scientist called Hebb has discussed a revolution learning algorithm that created the basis of the neural networks algorithms. This algorithm was actually discussed in 1949 (Hebb, 1949).

In 1954, Farley and Clark were the first who used computation machineries, to simulate a Hebbian network in the Massachusetts Institute of Technology (MIT), these machines were called calculators. Rochester, Holland, Habit, and Duda have created other neural networks calculation machines in 1956. By the 1962, Rosenblatt succeeded to establish a learning algorithm that converges always to the minimum error. This was an adjustment in the loop of the weights of the neural networks. The loop continues updating the weights until a good set of the required outputs is generated. However, the computers of that era were not suitable to perform the huge calculations required by such algorithms. This fact affected highly the development of the neural networks of that time.

The implementation of single layer networks was inefficient in creating solutions for different scientific problems. The multilayer networks were believed to be the solution for complex problems and able to offer high performance. Unfortunately, there were no effective learning algorithms that are able to provide convergence while training such multiple layered networks.

In 1975, the cognition system established by Fukushima was the first example of multilayer neural network that has a constructive efficient training algorithm. The structure of the system and the synaptic weights were changed between a structure of the neural and another. Each new structure had its advantages and disadvantages in addition to its strength points. Some neural networks are able to spread the information in one direction while other structures send data forward and backward firing different activation functions. Hopfield's networks invented in 1982 and had the ability to propagate information in bidirectional manner (Minsky & Papert, 1969). The implementation of the so named back propagation neural network was mostly the main motivation for the integration of the artificial neural networks in 1986. That algorithm has proposed the propagation of an error signal through the different layers of the network. The propagated error was then used to recalculate the new weight values in an ANN. A stochastic gradient descent algorithm was used to perform the training in that neural structure (Anderson & McNeill, 2010).

The back propagation algorithm attracted more interest as there was much debate on the possibility of implementing such algorithm in an artificial brain or not. This debate was mostly a result of the ambiguous idea about the training of the network at that time. The idea of using a target signal in a training process was a bit confusing and non clear. However, in the last decade different unsupervised learning algorithms were studied and proposed for single or multiple layer neural networks. Such procedures can be implemented to detect transitional versions even in the absence of desired signal (Rumelhart & McClelland, 1986).

4.2 Between human brain and human made brain

The human artificial system can be seen and explained as a structure composed of three phases. The first step or phase is the brain that is considered as the central human nervous system. Brain collects data constantly, process the collected information and responds with the suitable reactions. The second step in the human artificial system is a collection of lines and tracks that ensure the information transfer from biological receivers distributed all around the body in the form of electric signals. They also play a vital rule in transferring reaction instructions between the brain and the actuators of the body to do the reactions. The mentioned sensors are the third part or phase of this complex biological system. These sensors are situated in all the body receiving different types of senses and signals and transferring them in a closed process to the controller (brain).

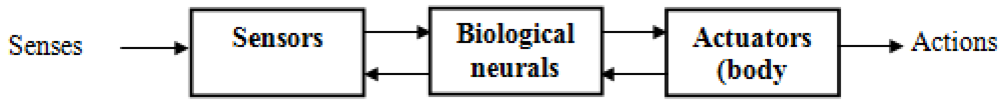


Figure 4.1: Human nervous system

The biological brain is created of an interconnected huge network composed of nervous elements known as biological neurons. These neurons are interfaced with sensors and actuators. The human brain is approximately built up of more than 100 billion cell of different shapes. The brain neural cells represent approximately 10% of that number of cells. The rest of cells are called glial or glue cells; they support cells for the neurons. Neurons interconnect through touching points known as synapses. On the average each neuron receives signals via thousands of synapses (Minsky & Papert, 1969).

The artificial neural network is a huge parallel processing unit constructed of simple units. These units have properties of storing the experience learned through examples and make it accessible when needed. The ANN is similar to human brain in many aspects:

1. Learning is gained by the network by examples processed during a process called training or learning, it is similar to the learning in human brain.
2. Synaptic weights are used to save the knowledge in the neuron connections.

4.3 Elements of Artificial Neural Network

A neuron is an information- processing unit that is fundamental to the operation of a neural network. The first attempts to simulate neural networks in the nervous system are to draw the main features of neurons and how interconnection and programmed it by a computer program to simulate these features. Our knowledge of nerves system is not great enough and our potential technological is limited. The basic artificial neural network structure is composed of three basic components; these are shown in Figure 4.2.

1. A set or number of sets of synapses links, each synapses link has a specific weight or own strength value, these weights are subject to continuous variation during the learning process according to specific criteria that will be discussed later in this work.

2. A summing function that is responsible to collect the weighted input signals after being weighted by the respective synapses of the neuron, the operations described here constitute a linear combiner.

4. Transfer functions for limiting the amplitude of the output of a neuron. They are also known by different terms like activation function as they decide whether to activate or deactivate the output from a set of neurons. They can be defined also as squashing functions as they have ability to squash or limit the output of the neural network. Typically the normalized amplitude rang of the output of a neuron is written as the closed unit interval [0 1] or alternatively [-1 1].

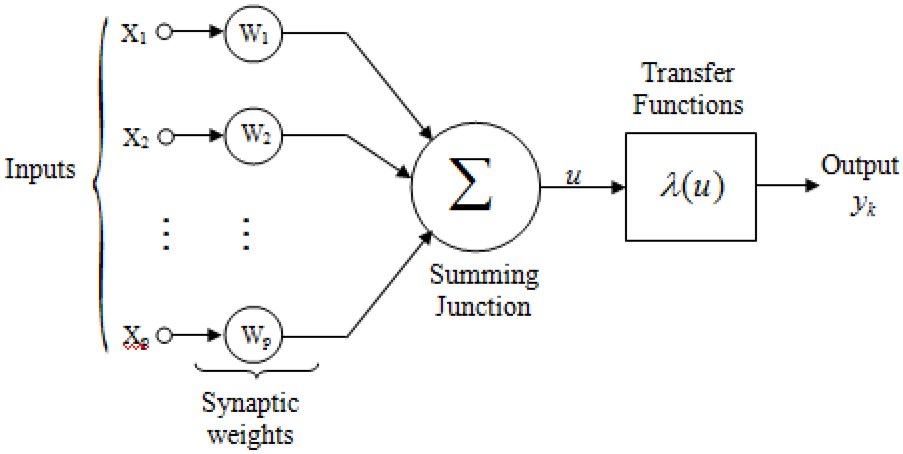


Figure 4.2: Basic components of ANN

4.3.1 Activation functions

The transfer function, known also as transfer function determines the result of the neuron in form of level of input efficiency. The input of an activation function is the sum of the weighted inputs of the neuron. Its output is found by applying the transfer function to the sum. In artificial neural networks, there are mainly few types of transfer functions like hard limit transfer function, ramp transfer function, logarithmic transfer function, and tangential transfer function (Rumelhart & McClelland, 1986):

4.3.1.1 Threshold function or hard limit: This form of function is usually known as a Heaviside transfer function. Its output can be one of two levels dependent on an input threshold. It can be easily expressed by:

$$y_k = \begin{cases} 1 & if \quad x_k \geq threshold \\ 0 & if \quad x_k < threshold \end{cases} \tag{4.1}$$

The value of the threshold in the previous equation is a user determined function. It can be set based on experience of the user or in some cases it can be set to zero for simplicity. In the model of McCulloch-Pitts presented in 1943, the output of a given neuron can take the value of one if its local field is positive and take null in the contrary case. Figure 4.3 presents the output of such a transfer function where it is clear that the output is sharp and changes in a sudden manner.

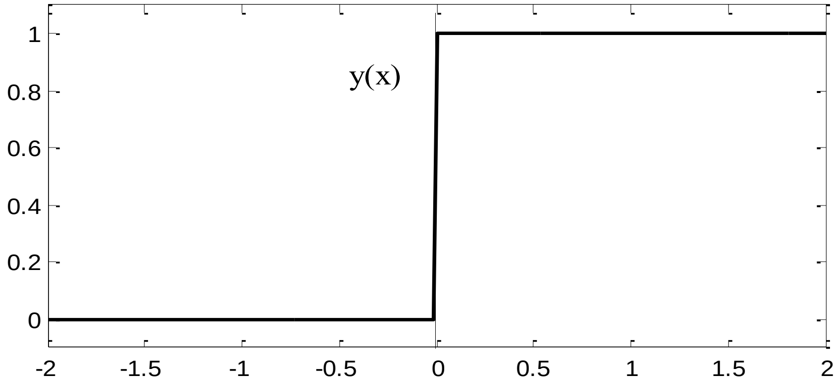


Figure 4.3: Threshold transfer function

4.3.1.2 Linear activation function: In this function, the output can also take one of two values like the threshold function. However, instead of changing its output sharply at a given threshold value; the value changes linearly within some limits to avoid sharp shape of the output. The magnification factor within the linear area is accepted to be 1. However, any other factor is acceptable if it can serve the goal of the network. This type of transfer functions can be seen as an estimate of an amplifier with saturation. The next cases can be considered as subcategories of the linear transfer function:

a. A linear summer appears if the linear area of function is kept under the saturation level.
b. The linear function changes to a threshold function if the slope of the linear area is infinitely big.

The output value of a neuron passed by a linear function can be easily defined by:

$$y(x) = \begin{cases} 1, & x \geq +\theta \\ x, & \theta > x > -\theta \\ 0, & x \leq -\theta \end{cases} \tag{4.2}$$

Figure 4.4 shows the Piecewise linear activation function

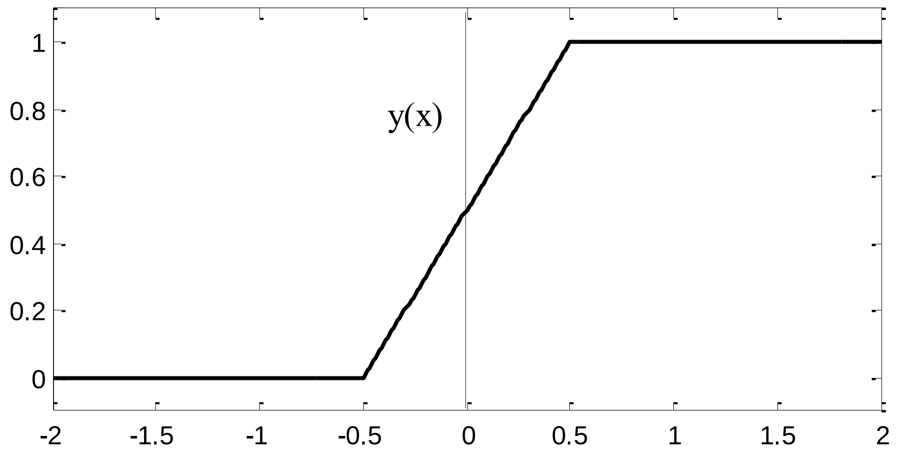


Figure 4.4: Linear Activation Function

4.3.1.3 Sigmoid functions: the sigmoid function is the mostly used activation function. It is implemented in most of the structures of artificial neural networks. It can be accepted as the most accurate increasing of the increasing input. In this type of functions, there is equilibrium between the linear and the nonlinear performance of the function (Seiffert, 2002). The sigmoid function has an important parameter that can vary the shape and behaviour of the function. This parameter is the slope of the function which is normally variable and can be adjusted based on the needs and application. By varying the slope, we obtain sigmoid function of different forms. When we make the slop parameter approaches infinity, the sigmoid function become a threshold function. Where the threshold function range value is 0 or 1, the sigmoid function has a continuous range of value from 0 to 1. The equation below shows the sigmoid activation function representation and how the output value calculated.

$$y(x) = \frac{1}{(1 + \exp^{-ax})} \tag{4.3}$$

The sigmoid activation function rang is between 0 and 1, but in some cases the range that used in this function is between $^{-}$1 to 1, where it is called tangent sigmoid function. The activation function is an odd mathematical function that creates neighbouring field. Figure 4.5 and 4.6 shows the two shapes of sigmoid activation functions.

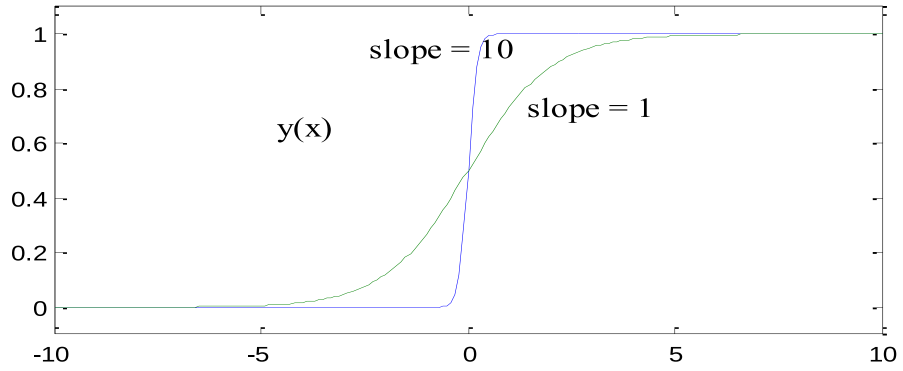


Figure 4.5: Sigmoid activation function, logarithmic

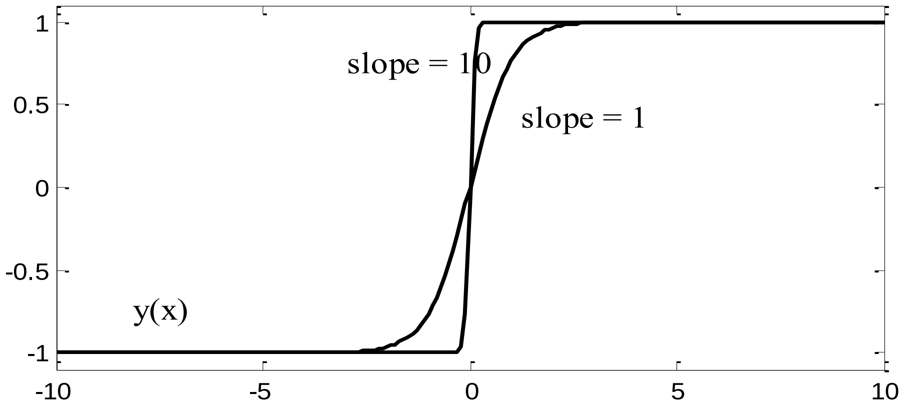


Figure 4.6: Sigmoid activation function, tangent

4.4 Neural Network Functional Structure

The perceptron is the simplest type of artificial neural network. It is mainly constructed from one single neuron that has one variable synaptic weight as demonstrated in Figure 4.7 (Minsky & Papert, 1969).

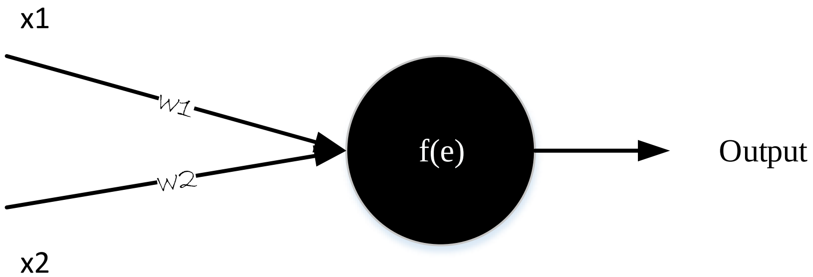


Figure 4.7: The Perceptron neural network (Khashman, Sekeroglu, & Dimililer, 2006)

The structures of ANN can be actually classified into three classes, these are:

1. Single layer perceptron feed forward networks (SLP).
2. Multi-layer perceptron feed forward networks (MLP).
3. Recurrent Networks.

4.4.1 Single layer perceptron (SLP)

Artificial Neural Networks are arranged in the neurons as a set of consecutive layers. Each layer is constructed from multiple artificial neurons. The simplest variety of these classes is the input layer. Input layer is the source layer where all inputs are fed to the network. This layer is followed by an output layer that is called computation node. The main feature of this type of ANN is that it has just a feed forward attack with no feedback process. The form of this network can be clarified in Figure 4.8 below.

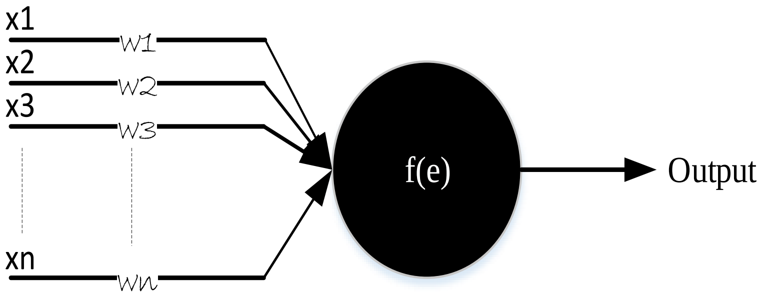


Figure 4.8: Single Layer Perceptron SLP

4.4.2 Multi-layer perceptron (MLP)

The second type of neural network discriminates itself by single or multiple hidden layers or nodes. Hidden layers are computational layers that are used in the output generation. The purpose and the task of hidden layers are to interface between the inputs and the outputs of the neural network in some useful processing approach. By inserting one or more hidden layers the neural networks can trigger the aptitude of enormous knowledge, pattern recognition, identification, and many other tasks. The process of neural networks is getting more complex by adding additional neural networks and get easier by decreasing the number of hidden layers. Weights of hidden layers are also variable and can be changed based on the requirement of a given application. Figure 4.9 presents the structure of multilayer neural network.

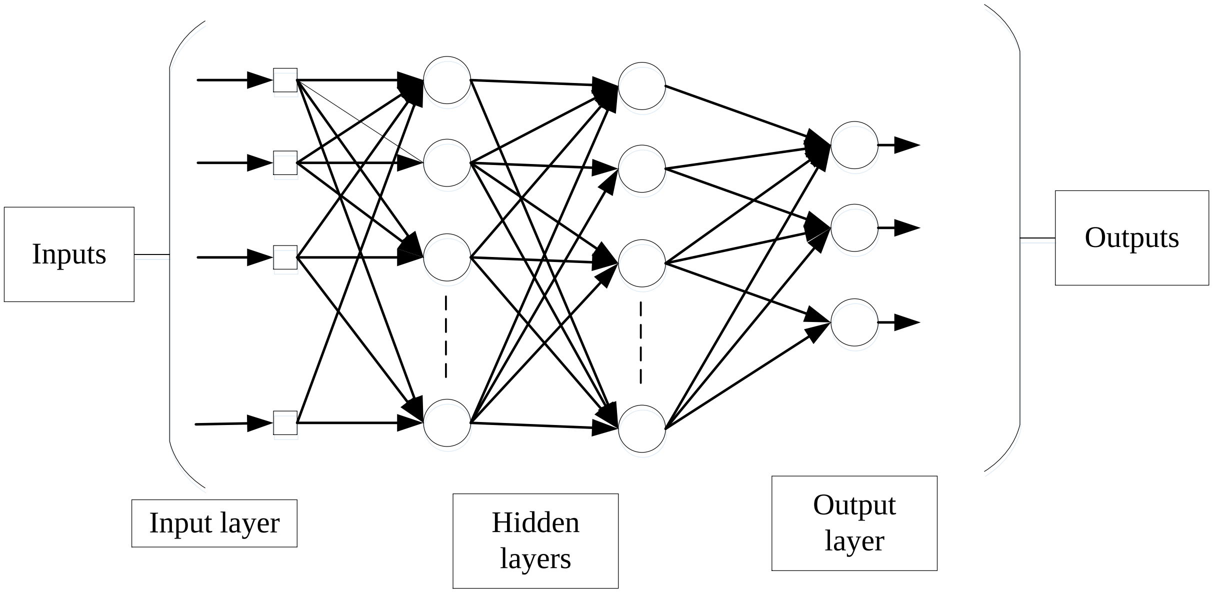


Figure 4.9: Multi-Layer Perceptron MLP with Two Hidden Layers

The source node in the input layer of the network supply elements to the computation neurons in the first hidden neurons in the next layer, which is her first hidden layer. Output of the second layer (first hidden layer) represents the input of the next layer, and so on for the rest of the neural network layers until the output layer which is the last computation layer in the neural network. All neurons in the neural network have inputs and also have outputs at the same time. Outputs of the last layer neurons (final layer) in artificial neural networks are the output of the network by processing the supplied input source node in the input (first) layer.

The neural network said fully connected in the sense that every node in each layer of the network is connected to every other node in the adjacent forward layer. If some of the communication links (synaptic connections) are missing from the network, the network is partially connected.

4.4.3 Recurrent networks

A recurrent neural network distinguishes itself from a feed forward neural network and it has at least one feedback loop. That means the output of the feed forward neural networks due to be input in the rest of the neurons and feedback in the same network but not to the same neurons itself.

The presence of feedback loops in artificial neural network gives it the ability to learn from the network and affect its performance, the Figure below shows the recurrent neural networks.

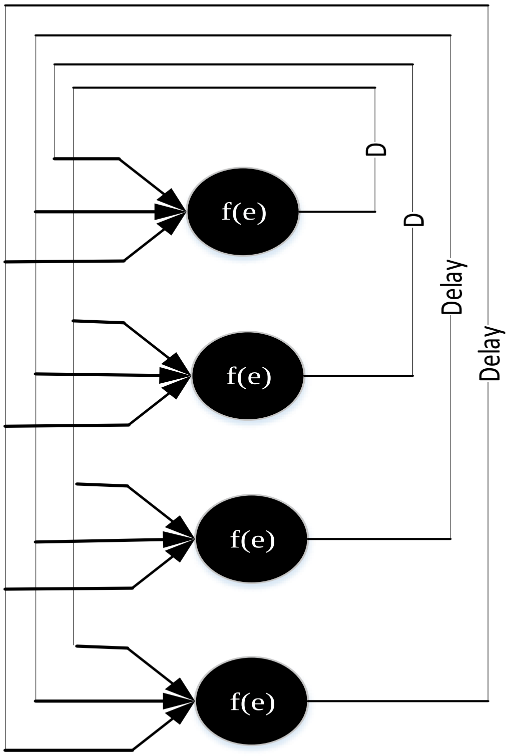


Figure 4.10: Recurrent Networks (Rumelhart & McClelland, 1986)

4.5 Artificial Neural Network Learning

The significance of artificial neural network is the ability of this network to learn from its environment. The process of learning in artificial neural networks and also referred to as training process depends mainly on the pattern recognition. Where the artificial neural networks like human it learns through examples. The neural networks learn about the environment through adjusting the synaptic weight value and bias levels. There are two types of learning artificial neural networks first one called the supervised learning. In this type of learning, the learning process is under the supervision of the teacher, in this case the programmer. Where the teacher is determined the desired output or as called target in terms of inputs the learning rate depends on the response of artificial neural networks to the training, and how closes the actual output resulting from the training process of the desired output (target). The second type of learning is unsupervised learning. In this type the neural network doesn't need to a teacher to determine the output of the network (Seiffert, 2002).

There are a large number of training algorithms that used to train artificial neural networks. The most important and best known and most widely used algorithm is Back-propagation training algorithm. This is the algorithm that was adopted to train the artificial neural network system in this thesis.

4.5.1 Learning paradigms

To make the learning process easier to be understood; Consider a neural network of single input neuron in input layer and one computation neurons (output layer) and one or more hidden layers. When the input node feed the network with the input signal it will be operations computation in the hidden layers to the arrival to the output layer. Then we have an actual output, represent the output of the neural network. We make a comparison between the actual output obtained from the neural network with the desired output or as called the target. This learning method is known as learning by error correction learning.

The objective of this method is to create a sequence of adjustment on the synaptic weight of the network to make the actual output signal of the neural network come closer to the desired output or target. There are two types of learning paradigms (Seiffert, 2002):

4.5.1.1 Supervised learning

Supervised leaning and also referred to learning with a teacher. In these paradigms the teacher must have knowledge of the environment surrounding the neural network. The teacher represented this knowledge as a sequence of input output examples. Suppose that the teacher wants to teach the neural network a particular process, the teacher must provide the neural network with the desired outputs according with the inputs in the training process. Certainly as we have already explain the training process done by modifying the values of the synaptic weights in the neural network then reduce the value of the actual output and compare it with desired outputs (target). Thus, step by step, the neural networks are simulating the teacher. In the training process knowledge is transferred from the teacher to the neural networks. When the neural network gets to this stage the neural network dispenses the teacher and start working and deal with the environment completely by itself (Seiffert, 2002) (Schmidt, 1996).

4.5.1.2 Unsupervised learning (self organization learning)

Unsupervised learning or as called learning without teacher or self organization learning is the simplest form of learning. Through the name of this learning paradigm there is no teacher to supervise the learning process. That mean it has no examples to illustrate the environment surrounding the neural networks. Neural networks without any outside help, it response to the input signal that feeds and automatically respond to the features of these inputs, and make an adjustment of the synaptic weight , because of that, these networks have a different response for each set of input (Caruana & Niculescu, 2006).

4.5.2 Learning rules

Neural networks are adjustable elements with statistical capabilities. The weights can be adjusted in an iterative manner to achieve a desired output. The adjustments are adopted based on the used learning method that can be either supervised (when a target output is identified and used to generate an error) or unsupervised learning where no target is required. There are many learning algorithms that are in use. Most of the existing methods are updates of the old learning algorithms like Hebb’s rule. Different learning algorithms are being introduced as new methods in different articles. Some scientists set the modelling of biological neural network as their main goal. Other scientists are in the race of finding models of their own perceptions about the learning nature. A few of the main learning rules are offered as exemplar:

4.5.2.1 Hebbs rule

The best known training algorithm was introduced by Hebb. The explanation of this algorithm as introduced in his book about Organization of Behaviour in the year 1949. His main rule is: If a neuron accepts an input from a neuron and if both are extremely active, the weight between two neurons increases if the two neurons activate simultaneously and reduce if they activate separately (Hebb, 1949).

4.5.2.2 Hopfield rule

Hopfield rule is the same as Hebb's rule except in that it identifies the amplitude of the reinforcement or deteriorating. It says, if the target and real output are active or inactive, increase the related weights with respect to the learning rate, otherwise decrease the weight according to the learning rate (Anderson & McNeill, 2010).

4.5.2.3 The Widrow-Hoff learning rule

This rule is an alternative of the rule of Hebb. This rule is among the best known and famous supervised learning rules. It is constructed on the straightforward scheme of constantly updating the strength of the neural connections to minimize the error (delta) between the actual and the desired output value in the last or output layer. This rule updates the synaptic variables in the direction that leads the mean squared error (MSE) of the network toward null value. This rule is also referred to as the Least Mean Square Error (LMS) Learning Rule based on the LMS algorithm. The error rule idea is based on the transformation of the error of the output layer through the transfer function’s derivative. The transferred value is then employed to adjust the weights in the previous layer of the network. In other terms, the error is propagated back through layers one by one until the first layer. This update process goes in loop until the actual error is minimized to an acceptable value. The process name Feed forward, Back-propagation is derived from this method of error values computation (Schmidt, 1996).

4.5.2.4 Gradient descent rule

The idea in gradient descent algorithm is to increasingly and constantly minimize the error by adjusting the weight values. The problem is in finding how to adjust the weight values. In reality, if we can build an idea about how the weight variation affects the error, then we seek to minimize that error in a mathematical manner (Schmidt, 1996).

4.5.2.5 Kohonens learning rule

This rule was developed by Teuvo Kohonen. It is stimulated by the learning in real biological neurons. The Kohonen rule permits the weights in a neuron to learn the pattern in an input set; thus, it is important for recognition purposes. In this rule, the neuron that has closest weight vector to the input vector is modified such that it becomes closer. This way, the winner neuron has more chances to be chosen in the next loop with similar inputs. As more inputs are presented to the network, each element in the layer becomes closer to a set of inputs and adjusts its weights according to it. As a result; if there is sufficient number of neurons each set of inputs will associate a neuron that generates 1 when a vector of that set is used. It will generate 0 with any other input vector. Thus, the competitive network is trained to classify the presented input vectors and categorize them.

4.6 Back Propagation Algorithm (BP)

Back propagation (BP) is the most famous method of teaching ANNs. It is used to minimize a desired function of choice. It is expressed as a multi-stage vibrant process optimization algorithm. In 1969 it was claimed that a two layer network is able to defeat many learning limits. However, until that time, there was no solution on how to update the weights in the input and hidden layers (Minsky & Papert, 1969) (Rumelhart & McClelland, 1986).

Back propagation is a supervised learning method. It requires a dataset of the wanted output for different inputs to construct the training dataset. It is most useful to train the neural network with the feed-forward networks and feed backward networks. The term back propagation is an abbreviation of error back propagation. Back-propagation process implies the use of differentiable activation function in order to update the weights correctly. The back-propagation process consist of two basic shoots through the network layers, forward shoot and backward shoot as presented in Figure 4.11.

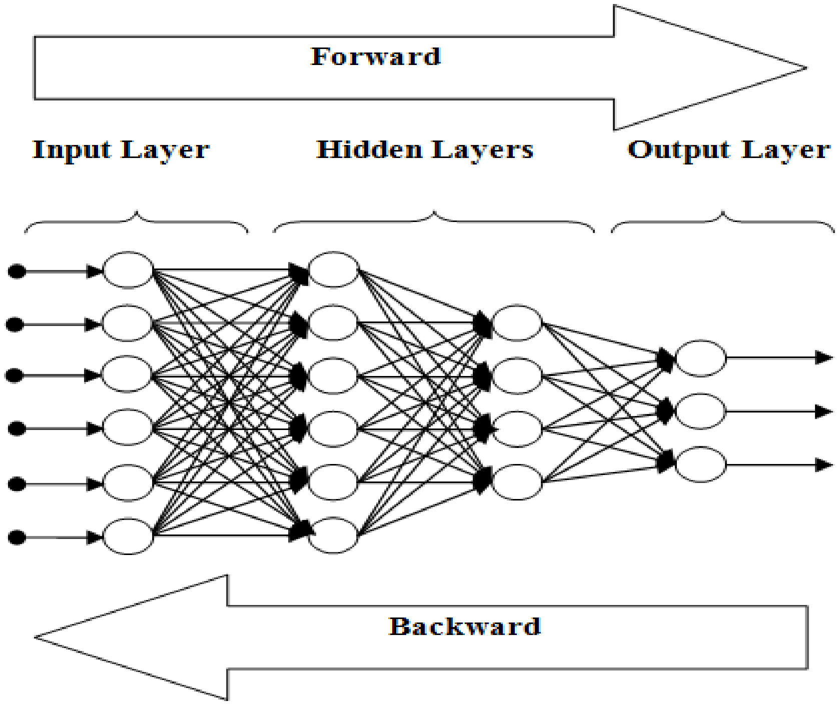


Figure 4.11: Back Propagation Network Architecture

BP algorithm, calculated the output signal from forward network depending on the error-correction rule, at this stage the synaptic weight does not change and remains constant, after comparing the output error signal or actual output with the desired output (target) and then propagate the error signal backward through the network and adjust the synaptic weight. Hence the name of the back-propagation algorithm comes from. This process will continues to make the actual output closer to the desired output. The steps of back propagation algorithm are:

Feed Forward: Each input pattern in a training set is applied to the input unites and then propagated forward. Initialize hidden and output weights to small random values. Calculate outputs of hidden neurons. Calculate outputs of output neurons. Make differences between the actual outputs of the output neurons and targets to get the error signal. If the actual output is equal or less than the target then the neural network was trained otherwise it must propagating the error signal backward.

Feed Backward: The error signal for each output pattern is back-propagated from the outputs to the input in order to adjust the synaptic weights in each hidden and output layers of the network.

Repeat the above two passes to make the actual output closer to the desired output and repeated it many times until reaching the error goal. After a back propagation learning algorithm was learned the network. The network can be tested with a second set of input to see how well it trained and calculate the network accuracy.

4.6.1 Feed forward network stage

When the feedforward process started to training artificial neural network by using back-propagation learning algorithm, the input signal of the input vectors are passed forward through the network and then to the output layer through the weights. The initial point of any network is a neuron input as shown in Figure 4.12.

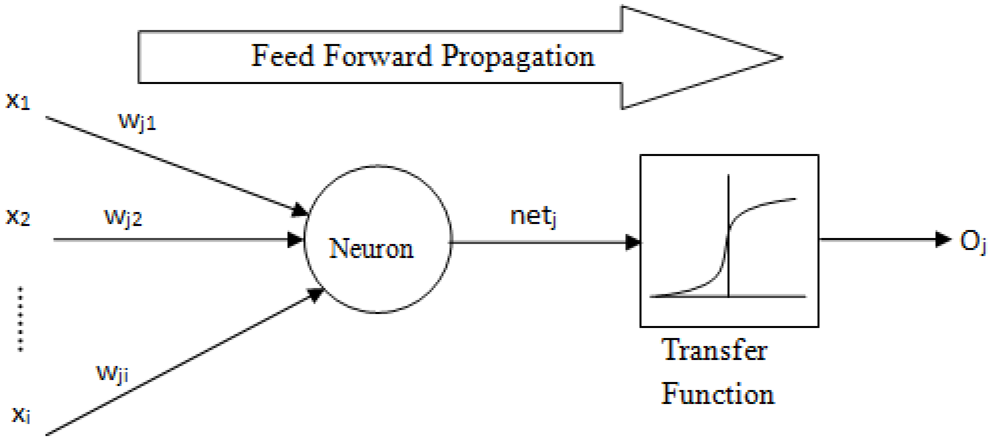


Figure 4.12: Feed Forward Neuron Structure

This input neuron is constructed from MISO system. Each one of the inputs is amplified through a weight that is multiplied with the input value. The neuron actually combines all these inputs together while the activation function will generate the output of the neuron. The idea of feed forward network is passing inputs forward, all the outputs are calculated through the transfer function, in most applications it was usually a logarithmic sigmoid function, as shown in Figure 2.6. The output of any neuron in any layer can be expressed by two equations:

$$net_j = \sum x_i w_{ji} \tag{4.4}$$

$$O_j = f_{th}(net_j) \tag{4.5}$$

For every neuron, *xi*, in a layer, each input neuron is cross multiplied by an initially generated weight w_{xi}. All the result of all the neurons of the same layer is summed together, generating the inner value of the summation. Then this value was feed through an activation function, f_{th}. The resulting output, O_j, is used as an input for the coming layer or it is an actual output of the ANN if speaking about the output layer.

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

$$f(x) = \frac{1}{(1+\exp^{-ax})} \tag{4.6}$$

4.6.2 Error signal

The calculation of the feed forward network during the training process is combined with backward propagation. The error signal is very important for calculation of new weight values that determines the ANN learning. The perception of learning a neural network is the real definition of error in the network.

Rumelhart and McClelland have defined the error based on the divergence between a desired output of the network and the actual real output. The actual output is the result of the feed forward process over the inputs. The error is an indication of the degree of success the training on a particular training set is (Rumelhart & McClelland, 1986).

In order to reduce the sum of the patterns error in the training set, the gradient descent error method is employed. In this method, weights are calculated iteratively with consideration of the error derivative's inverse as the next equation shows:

$$\Delta w_{ji} = -\eta \left[\frac{\partial E}{\partial w_{ji}} \right] \tag{4.7}$$

Where η the learning rate and E is the total error. As it can be seen from the equation above, the new value of the weight is the old value with a small delta term. This delta term is calculated from the derivative of the total error sum with respect to the weights at each one of the layers.

4.6.3 Mean squared error

Mean Squared Error (MSE) relation is the most famous error approach used in the process of Back Propagation learning. The error is calculated using the gradient descent method. This method is mainly based on the idea of decreasing the error between the desired output of the process and the actual output. The MSE is calculated based on the equation:

$$E = \frac{1}{2}\sum_{k}(t_{kj} - O_{kj})^2 \tag{4.8}$$

Where

t_{kj} is the desired output from node (k) to node (j)

O_{kj} is the network output value from node (*k*) to node *(j)*

4.6.4 Learning rate and momentum factor

These two parameters are effecting to the learning ability of the artificial neural network. The learning rate factor (η) that identifies the learning speed of the network. It represents the step size of weight adjustment at each iteration. Larger step size makes the convergence of the error toward a minimum faster. However, this can lead to a memorization phenomena rather than a learning process because the learning is done faster and the network can't detect the pattern in an input dataset. If the learning rate is large number, huge change in weight make a fluctuation in the search path and convergence more slowly than a direct descent. The momentum factor (α) is another important factor that affects the learning process of a neural network. It determines and attenuates the speed at which the neural network learning is done. It is very useful in making the neural network avoid getting caught in local energy minima. It could also smooth out the error path by stopping the hard variations in weight values. Both the learning rate and momentum factor can have a value between 0 and 1.

4.6.5 Weight adjustment

The weight has set to a random initially value as regularly executed. Weight updating is done in many stages. These stages start at the output layer after the end of the feed forward phase, and flow backward till reaching the input layer passing by the hidden layers. Suppose initial small random values of the weights that used in the neural network, where the values must be random and non-symmetrical that makes the neurons do deferent operation. If all neurons have the same weight values, they respond the same for any input pattern. The gradient, which updates the weights, would be the same for each neuron, that mean the weights would remain the same even after the update and this means there is no learning. The weights that feed the output layer and the hidden layers are updated using below equation:

$$w_{ji}(n+1) = w_{ji}(n) - \eta \Delta_{pj} O_{pi}^{T} \tag{4.9}$$

4.6.6 Back propagation training

The back-propagation learning algorithm is one of the supervised learning algorithms for artificial neural networks (ANNs). It extends the weight update rule for adjust the weight between the neurons in the network layers in multilayer feed forward artificial neural network. In Figure 4.13 the block diagram of back propagation learning algorithm is shown.

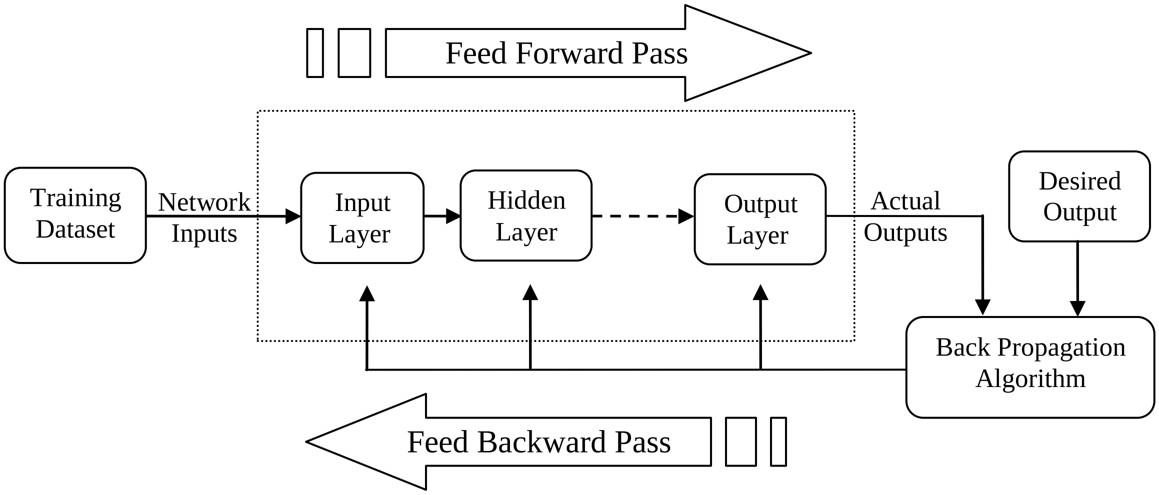


Figure 4.13: Block Diagram of Back Propagation Network

In back propagation artificial neural networks (BPNNs) the learning algorithm carried out the training process. Training input patterns are given to the input layer. The input prototype is spread forward via the network structure to the output layer. An actual result that is different from the target is found. An error is found and then spread backwards all over the network. The weights are updated based on the propagated error values. Figure 14 explains briefly all these training steps.

The advantages that make the back-propagation learning algorithm is the most common used to train the artificial neural network with respect to the simplicity of use and the fact that it is faster in training networks than other techniques. Back propagation has proven its ability to solve complex training problems at high speeds and low processing costs.

CHAPTER 5

ANN TRAINING AND TEST

This chapter will discuss the use of artificial neural networks and image processing techniques in the development of a computer based ear recognition. Ear images are going to be treated and processed using different techniques to improve the recognition performance. Different processing stages and neural network parameters were combined in multiple experiments to verify the power of ANN in recognition and identification of ear images. The ANN training was applied using Fujitsu Laptop computer with Core i3 processor that has a speed of 2.3 GHz and RAM of the computer is 4 GHz.

5.1 Weiner Filter with ANN

In the processing of the images, Weiner filter was applied to remove the noise and smoothen the processed images. The processing phase of this part was applied using the next processing methods:

1- Right and left ear images are read from the database directory.
2- The read images are converted into gray scale images.
3- Weiner filter with window of 3*3 is applied on the gray scale images to remove the noise from the images.
4- Filtered images are then averaged to achieve an image size of 50*30
5- The 50*30 images are converted to vectors of length of 1500 elements
6- The vector image is normalized to reduce the pixels range and simplify the recognition task.
7- The normalized vectors are supplied to the neural network for training

5.2 First Set of Subjects (4:1 right ear images)

The ear images were arranged in files in order to be identified by using the ANN. Images of 99 different subjects were used in the proposed system. Each one of these subject has 10 different images of right and left ears taken from similar distance. A sample of the processed data base ear images is presented in the next Figure. The training of artificial neural network is a long and time consuming process that can vary from few minutes to several hours or days. In order to optimize the recognition task of ANN, the training is applied times over times with different parameters until the best results are obtained. In the first step, images of the right ear were considered for the training and test of the neural network. 4 images out of 5 of each subject were used for the training while the fifth image was preserved for test purpose of the network. The MATLAB ANN toolset was used for the training as shown in Figure 5.2.

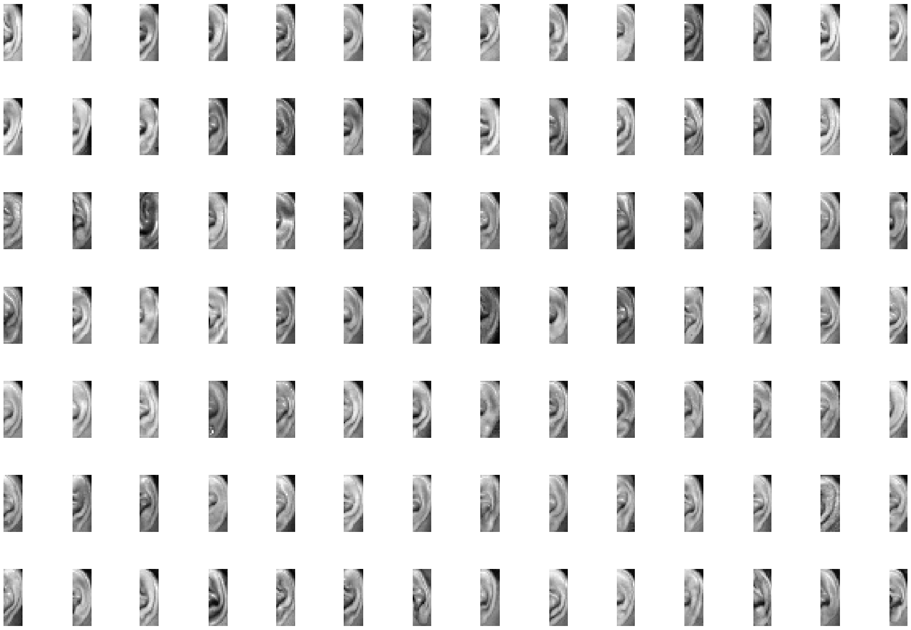
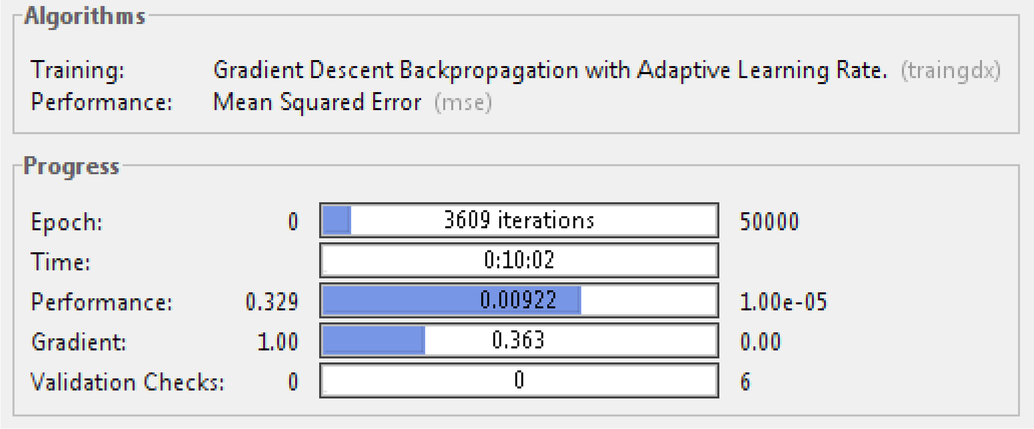


Figure 5.1: Sample of the processed data base images



Algorithms

Training: Gradient Descent Backpropagation with Adaptive Learning Rate. (traingdx)
Performance: Mean Squared Error (mse)

Progress

Epoch:	0	3609 iterations	50000
Time:		0:10:02	
Performance:	0.329	0.00922	1.00e-05
Gradient:	1.00	0.363	0.00
Validation Checks:	0	0	6

Figure 5.2: Training of ANN with 4 right ear images

The parameters of the used network in this case are presented in table 5.1 below. Figure 5.3 illustrates the curve obtained from the MSE of the training. The training of this set took approximately 26 minutes and 16 seconds. After choosing a suitable threshold value that is based on the training and test results, the efficiency of training and test was obtained. A sample of the obtained results is presented in table 5.2 for illustration.

Table 5.1: Parameters of the Training network.

Learning rate	**0.01**	**Maximum epochs**	**50000**
Momentum factor	0.1	Maximum accepted MSE	1e-2
Hidden layers	350, 400	Output layers	99
Training time	26.26 min	Activation	Tansig
Input layers	1500		

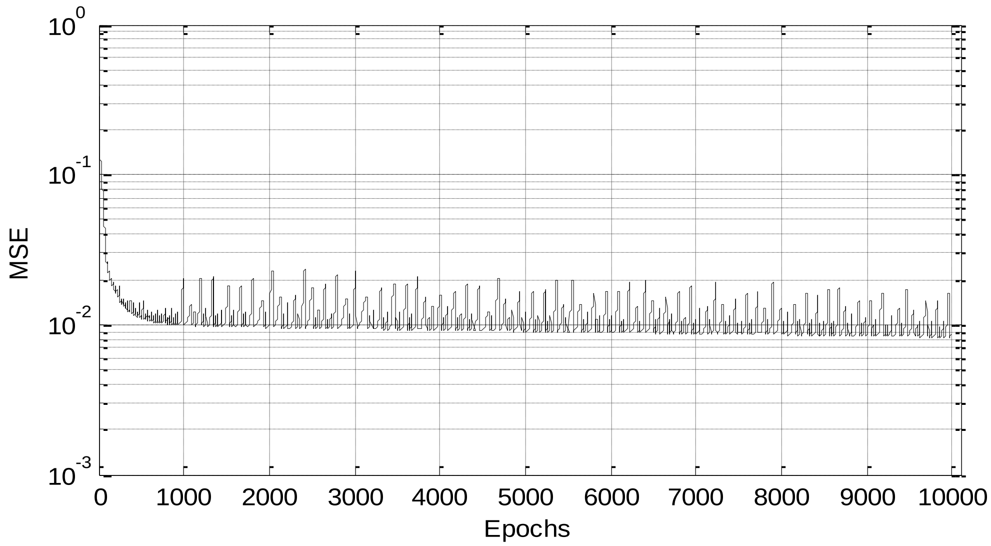


Figure 5.3: MSE curve of the first training set (4 R)

Table 5.2: Sample of the training results of the first set

Sub. 1	**Sub. 2**	**Sub. 3**	**Sub. 4**	**Sub. 5**	**Sub. 6**	**Sub. 7**
0,21	0,05	-0,01	0,09	0,03	0,10	-0,01
0,01	0,41	0,00	0,15	-0,02	0,05	-0,03
-0,04	-0,05	0,24	0,00	0,02	-0,07	0,00
0,01	0,01	0,01	0,52	-0,03	0,02	-0,01
0,01	-0,02	0,05	-0,04	0,24	0,04	-0,04
0,16	0,05	-0,08	0,02	0,02	0,45	0,00
-0,01	-0,04	0,02	0,04	-0,02	-0,03	0,46

After analysing all the training results, it was found that 364 images out of 396 training images were recognized correctly. The efficiency of the training was 91.9%. The test of the 5th image was applied on the trained network. Table 5.3 presents a sample of the obtained results. 69 subjects out of the 99 were correctly recognized in this set with an efficiency of approximately 69.7%. This result reflects a low efficiency compared with the training results. However, the result will be accepted as a first attempt of ANN recognition system.

Table 5.3: Sample of the test results of the first set

Sub. 1	**Sub. 2**	**Sub. 3**	**Sub. 4**	**Sub. 5**	**Sub. 6**	**Sub. 7**	**Sub. 8**
0,57	0,09	-0,01	0,05	0,01	0,15	0,01	0,04
0,12	0,23	0,01	0,04	0,01	0,08	0,02	0,00
0,04	0,01	0,48	0,02	0,01	0,02	0,00	0,01
0,10	0,02	0,00	0,50	0,00	0,21	0,03	0,04
0,01	0,03	0,05	0,02	0,35	0,02	0,03	0,02
0,13	0,01	0,04	0,08	0,01	0,37	0,05	0,02
-0,03	0,00	0,02	0,04	0,02	0,05	0,21	0,07
0,03	0,00	0,05	0,01	0,03	0,06	0,04	0,69

5.3 Second Set of Subjects (4:1 left ear images)

In this part, the five left images were used in the system. 4 images were implemented for the training of the system and 1 image was preserved for testing the quality of the learning. As in the previous set, the training time was 63 minutes and 42 seconds. The minimum obtained MSE was 0.006 after a total of 25800 iterations. The parameters of the system during the training are given in table 5.4. Table 5.5 presents a sample of the training results of this system.

Table 5.4: Parameters of the training of the second set of subjects

Learning rate	**0.1**	**Maximum epochs**	**25800**
Momentum factor	0.02	MSE	6e-3
Hidden layers	450, 140	Output layers	99
Training time	63.68 min	Activation	Tansig
Input layers	1500		

Table 5.5: Sample of the training results of the second set

Sub. 1	**Sub. 2**	**Sub. 3**	**Sub. 4**	**Sub. 5**	**Sub. 6**	**Sub. 7**	**Sub. 8**
0,42	0,14	0,01	0,13	0,05	-0,02	-0,04	-0,02
0,05	0,39	0,04	0,12	0,00	0,08	-0,07	-0,04
0,01	0,04	0,36	-0,01	0,12	0,01	-0,13	0,03
-0,01	0,13	-0,04	0,52	0,01	0,04	-0,08	0,01
0,01	-0,02	0,10	0,01	0,42	-0,01	-0,02	0,04
0,17	0,21	-0,02	0,10	0,01	0,63	-0,03	0,01
0,05	0,05	-0,05	-0,05	-0,05	0,05	0,46	-0,03
-0,05	-0,08	-0,05	0,08	0,07	0,14	-0,01	0,42

Figure 5.4 illustrates the curve of the MSE evolution during the training of the neural network. It is clear that the MSE was decreasing very slowly during the training. 390 images out of 396 training images were recognized correctly with a performance of 98.5%. Table 5.6 presents the results of the test of the second set ANN. It was found that 15 subjects out of the 99 test subjects were not surely recognized. However, the rest of subjects were recognized correctly with a performance of 84.8%.

Table 5.6: Sample of results of test of the last 8 subjects (second set)

Sub 92	**Sub 93**	**Sub 94**	**Sub 95**	**Sub 96**	**Sub 97**	**Sub 98**	**Sub 99**
0,20	-0,06	-0,02	0,05	-0,01	0,09	-0,06	-0,05
-0,07	0,79	0,02	0,00	-0,03	0,00	0,00	-0,02
0,02	0,01	0,42	0,02	0,15	0,10	0,11	0,21
0,05	-0,01	0,00	0,69	-0,03	0,09	0,03	0,04
0,00	0,02	0,09	0,06	0,50	0,04	0,03	-0,02
0,07	0,02	0,11	0,02	-0,03	0,46	-0,01	-0,01
-0,03	-0,03	0,04	0,06	-0,05	-0,01	0,70	0,04
0,02	-0,01	0,22	0,02	0,01	0,09	-0,08	0,48

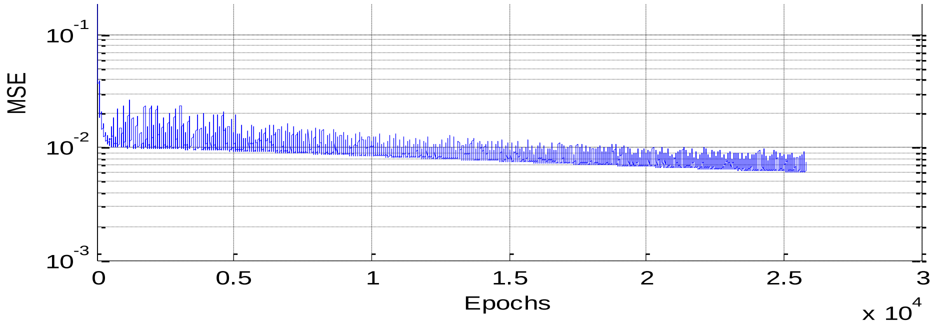


Figure 5.4: MSE evolution during the training of the second set

5.4 Third Set of Subjects (2:3 right ear images)

In this part, the number of training images was reduced to two while the rest of images were used for test of network efficiency. The parameters of the trained network are presented in table 5.7. The training of the network was very slow; the time of training was about 100 minutes. Two hidden layers were used with "tansig" activation functions. Sample of the training results is presented in table 5.8. 198 images out of 198 training images were recognized correctly with an efficiency of 100%. In the test, 257 images out of 297 test images were recognized correctly in this structure with efficiency of 86.5%.

Table 5.7: Parameters of the training of the third set of subjects

Learning rate	**0.1**	**Maximum epochs**	**41579**
Momentum factor	0.8	MSE	1.86e-3
Hidden layers	550, 310	Output layers	99
Training time	100.08 min	Activation	tansig
Input layers	1500		

Table 5.8: Training results of the third set of images

Sub. 1	Sub. 2	Sub. 3	Sub. 4	Sub. 5	Sub. 6	Sub. 7	Sub. 8	Sub. 9	Sub 10
0,85	0,07	0,00	0,02	0,00	0,04	0,00	-0,01	-0,02	0,00
0,03	0,73	-0,01	0,06	0,03	-0,03	0,05	0,00	-0,02	0,01
-0,03	0,02	0,76	0,01	0,04	0,00	-0,02	0,05	0,01	-0,02
0,05	0,06	0,01	0,81	0,04	0,02	-0,02	0,01	0,00	0,05
0,00	-0,04	-0,02	0,00	0,74	0,00	0,00	0,03	-0,01	-0,01
0,03	-0,05	0,03	0,04	-0,03	0,81	0,00	0,01	0,03	-0,01
-0,02	0,01	-0,02	-0,01	-0,01	-0,02	0,64	-0,05	-0,04	0,06
0,03	-0,04	0,00	0,02	0,03	0,00	-0,01	0,70	-0,02	-0,02
0,00	0,01	0,01	-0,01	0,01	0,04	-0,02	0,00	0,66	-0,04
0,00	-0,02	0,02	0,04	-0,06	-0,01	0,00	0,01	-0,02	0,62

Table 5.9: Test results of the third set of images

Sub 1	Sub 2	Sub 3	Sub 4	Sub 5	Sub 6	Sub 7	Sub. 8	Sub 9	Sub 10
0,46	-0,10	-0,08	-0,09	0,02	0,19	-0,15	-0,16	-0,08	-0,05
0,37	0,79	0,09	0,00	0,12	-0,07	0,05	-0,04	-0,05	0,08
-0,20	-0,14	0,58	-0,07	0,04	-0,04	0,09	0,18	0,03	0,07
0,18	-0,02	0,07	0,78	0,05	-0,20	-0,01	-0,20	0,15	-0,08
-0,06	0,11	0,33	-0,01	0,44	0,04	-0,08	0,13	-0,13	0,02
0,34	-0,14	0,00	0,12	0,06	0,83	0,01	0,36	-0,01	-0,01
-0,10	-0,01	-0,06	0,04	0,05	-0,06	0,22	-0,07	0,05	-0,06
0,12	-0,10	0,04	-0,06	0,09	0,12	0,13	0,05	0,06	0,08
-0,04	0,02	-0,01	-0,08	-0,11	-0,01	-0,19	-0,04	0,08	-0,11
-0,13	-0,21	-0,09	-0,02	-0,06	-0,07	0,08	-0,17	0,10	0,02

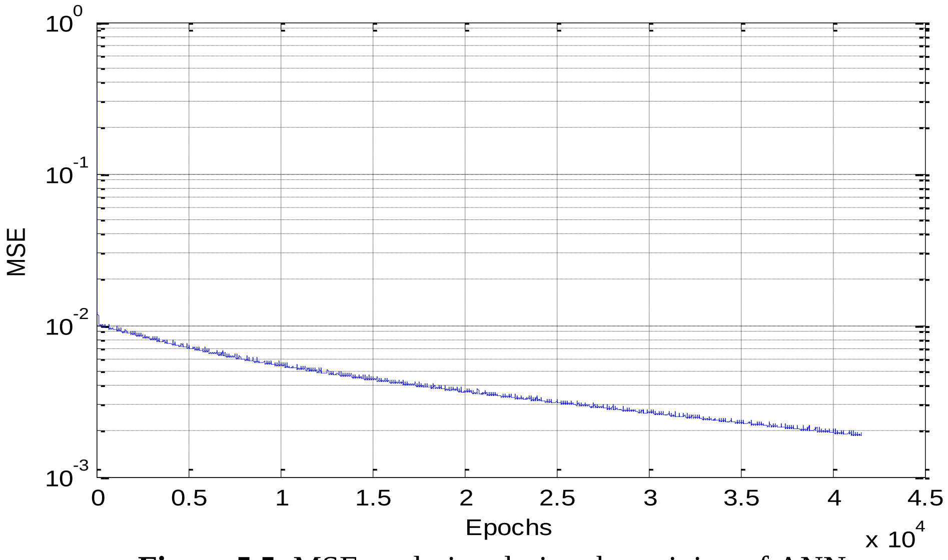


Figure 5.5: MSE evolution during the training of ANN

5.5 Fourth Set of Subjects (2:3 left ear images)

As in the last section, two images were used for training of the network in this section. The rest of images were conserved for test of the network. Table 5.10 presents the parameters used in the training of the ANN. Figure 5.6 illustrates the evolution of the training error throughout the training period. It shows that the error was decreasing continuously until it reached an acceptable error value. All the training images (198 images) were correctly recognized with 100% performance. During the test of the network, 270 images out of the 297 test images were correctly recognized. The efficiency of the test was about 90.9%.

Table 5.10: Parameters of the training of the fourth set of subjects

Learning rate	**0.1**	**Maximum epochs**	**25000**
Momentum factor	0.8	MSE	0.0048
Hidden layers	450, 280	Output layers	99
Training time	43.2 min	Activation	tansig
Input layers	1500		

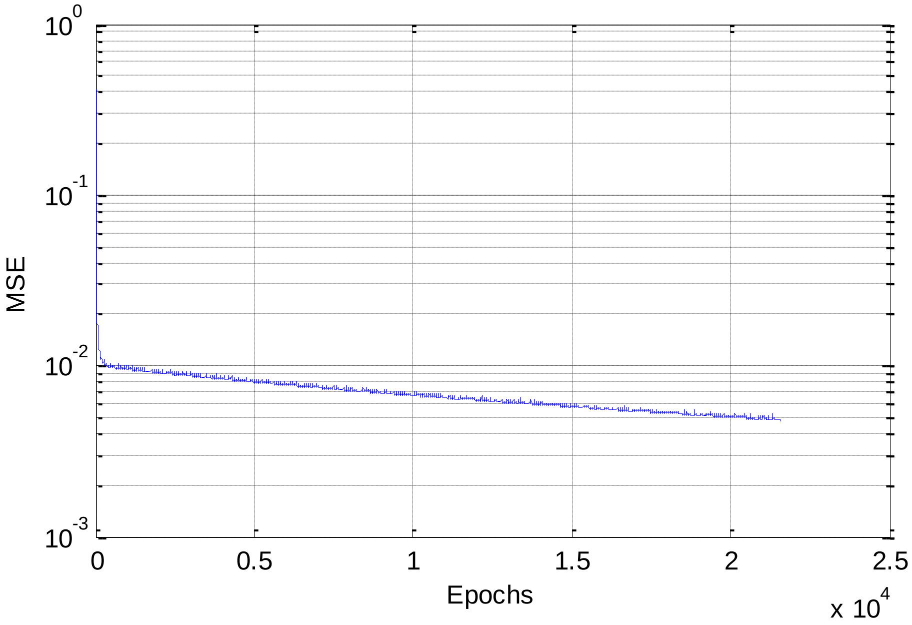


Figure 5.6: MSE curve of the trained network

5.6 Training of Combined Left and Right Image

In this part, instead of separating the left images and right images of each subject, they are used in the same system and the network is been trained to recognize both images at once. The training of this structure was the most difficult and took 5 hours and 50 minutes of training. Parameters of the system are presented in table 5.11. The Training mean squared error curve is presented in Figure 5.7.

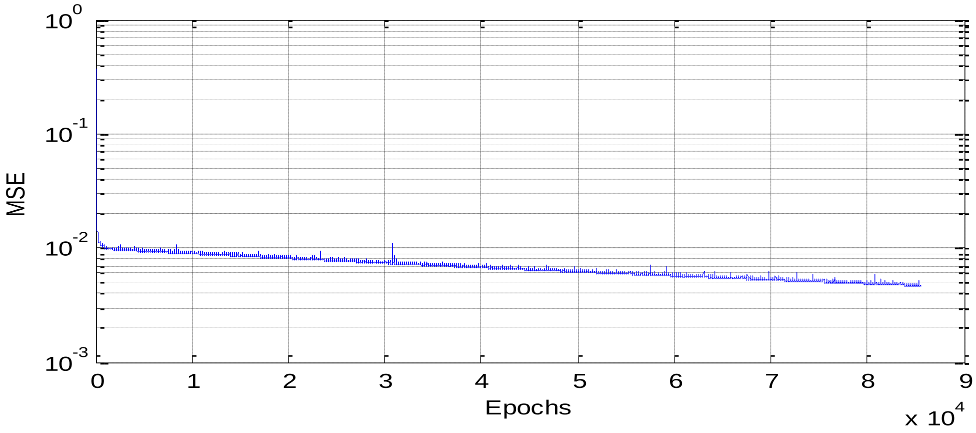


Figure 5.7: Training MSE curve of ANN

Table 5.11: Parameters of the training of the combined set of subjects

Learning rate	**0.02**	**Maximum epochs**	**85490**
Momentum factor	0.4	MSE	0.0046
Hidden layers	350, 450	Output layers	99
Training time	350 min	Activation	tansig
Input layers	1500		

In this section, 3 images of each ear were used for the training of the network. Six images were fed to the network during the training with total of 99*6=594 training images. 2 images of this dataset were not recognized during the training. This means a performance of 99.6% in the training phase of the network. For purpose of accuracy verification, the rest of images (396 images) were used for the test of the network performance. These images were not presented to the network during the training. 335 images of the test images (160 right and 175 left) were correctly recognized with good accuracy. The performance of test was then 84.6%.

Table 5.12: Sample of the training results of the combined right and left images

Sub. 1		**Sub. 2**		**Sub. 3**		**Sub. 4**		**Sub. 5**		**Sub. 6**		**Sub. 7**		**Sub. 8**	
R	**L**	**R**	**L**	**R**	**L**	**R**	**L**	**R**	**L**	**R**	**L**	**R**	**L**	**R**	**L**
0,5	0,6	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,0	0,0	0,0	0,0
0,0	0,0	0,5	0,3	0,0	0,1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
0,0	0,0	0,0	0,0	0,6	0,5	0,0	0,0	0,0	0,1	0,0	0,0	0,0	0,1	0,0	0,0
0,1	0,1	0,0	0,0	0,0	0,0	0,4	0,3	0,0	0,1	0,0	0,0	0,1	0,1	0,0	0,0
0,1	0,1	0,1	0,1	0,1	0,1	0,0	0,0	0,4	0,5	0,0	0,0	0,0	0,0	0,0	0,1
0,1	0,1	0,1	0,1	0,0	0,0	0,0	0,0	0,1	0,0	0,5	0,5	0,0	0,0	0,1	0,0
0,1	0,0	0,1	0,0	0,0	0,0	0,1	0,0	0,0	0,0	0,0	0,0	0,4	0,5	0,0	0,1
0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,1	0,1	0,1	0,0	0,5	0,4

5.7 Simulation of The System with Different Hidden Neurons

In this part, the simulation of the system will be applied with different hidden layers neurons to check the ability of the neural network. Because a lot of data is going to be obtained in this part, the results will be presented in table showing the efficiency under different parameters of hidden neurons. The results obtained shows that the number of hidden neurons is very important in the results. Different number of neurons give different efficiency and need different training time.

	Hidden Layers		**Efficiency**		
Set of data	first	Second	Training %	Test %	Training time
6 training and 4 test images	600	600	95	87	50 min
	800	600	92	83	72.5 min
	200	99	75	70	10.7 min
	600	198	86	75	23.3 min

CHAPTER 6
CONCLUSIONS AND FUTURE WORKS

In this work, an approach for ear recognition for security and identification purpose was proposed and discussed. The proposed system implements the back-propagation learning algorithm in the training of ANN for ear biometrics recognition. A database composed of 990 images of 99 different subjects was used in the training and validation process of the proposed recognition system. Different training to validation database was applied to verify the efficiency of the structure.

An averaging algorithm in accordance with image processing was used to reduce the complexity, size, and processing cost of learning process without losing any of the main features of the images. The implemented neural network algorithm was carried out through two phases; training phase and validation phase. The images were grouped into four different sets. Each one of the sets contains either the left or the right ear images of each subject. This way, the recognition system can double check the identity of person by using the right and left ear's image.

The results obtained from the system were satisfactory, different values of the accuracy of the system were obtained by varying the ANN parameters. The accuracy ranged between 89% and 100% in the training image sets while an accuracy ranging between 67% and 91% was obtained in the verification phase of the system. The obtained results illustrate the effect of the hidden layers neurons and other parameters on the system performance. An optimum number of hidden layer's neurons must be found for each individual set of subjects. It is important to notice that the processing or learning time in the ANN increases as the number of subjects increase.

The obtained results in this work prove the ability and performance of the artificial neural networks in biometrics and ear recognition.

This work opens new horizons and shows new perspectives for the investigation of an automated artificial neural network ear recognition system. Observations collected throughout this work clarified the need for the conduction of researches in multi-position ear image acquisition. Multi-modal biometric systems based on the artificial neural networks are also advised.

REFERENCES

Anderson, D., & McNeill, G. (2010). *Artificial Neural Networks Technology*. Rome: Rome Laboratory.

Caruana, R., & Niculescu. (2006). An Empirical Comparison of Supervised Learning Algorithms. *International Conference on Machine Learning*, (pp. 161-168). Pittsburgh. doi:10.1145/1143844.1143865

Chora, M. (2007). Image Feature Extraction Methods For Ear Biometrics - A Survey. *6th International Conference On Computer Information Systems And Industrial Management Applications* (pp. 261-265). IEEE.

El-Bakry, H. (2007). *Ear Recognition by using Neural Networks, MSc. thesis.* Mansoura: Mansoura University.

Ibrahim, M., & Ibrahim, S. (2012). *Wavelet Based Approaches For Detection and Recognition in Ear Biometrics, PhD Thesis.* University Of Southampton.

Kasprowski, P. (2004). *Human Identification Using Eye Movements,Ph.D Thesis.* Silesian University of Technology. DOI: 10.13140/RG.2.1.3466.3924.

Khashman, A., Sekeroglu, B., and Dimililer, K. (2006). *Intelligent coin identification system*. *Conference on Computer Aided Control System Design* (pp. 1226-1230). IEEE.

Minsky, M. L., & Papert, S. A. (1969). *Perceptrons.* Cambridge: MIT Press.

Prakash, S. (2013). *Human Recognition using 2D and 3D Ear Images, PhD thesis.* India: Indian Institute Of Technology Kanpur.

Purkait, R. (2007). Ear Biometric: An Aid To Personal Identification. *Anthropologist Special*, 215-218.

Rahman, M. M., Islam, M. R., Bhuiyan, N. I., Ahmed, B., & Islam, M. A. (2007). Person Identification Using Ear Biometrics. *International Journal of The Computer, the Internet and Management*, Vol. 15, 1-8.

Rumelhart, D. E., & and McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition.* New York: MIT Press.

Saleh, M. I. (2007). *Using Ears For Human Identification, MSc. thesis.* Blacksburg: Virginia Tech. University.

Schmidt, A. (1996). *A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors*. Manchester: Manchester Metropolitan University.

Seiffert, U. (2002). Artificial Neural Networks on Massively Parallel Computer Hardware. *European Symposium on Artificial Neural Networks Bruges*, (pp. 319-330). Belgium.

Shailaja, D. N. (2006). *A Simple Geometric Approach For Ear Recognition, MSc. Thesis.* India: Indian Institute Of Technology.

Singh, S., & Singla, S. K. (2013). Review on Biometrics and Ear Recognition Techniques. *International Journal of Advanced Research in Computer Science and Software Engineering*, 1624-1630.

Victor, B., Bowyer, K., & Sarkar, S. (2000). An Evaluation Of Face And Ear Biometrics. *IEEE*, 429 - 432.

Watabe, D., Sai, H., Sakai, K., & Nakamura, O. (2008). Ear Biometrics Using Jet Space Similarity. *IEEE*, 1259 - 1264. doi:10.1109/CCECE.2008.4564741

Yang, P. (2006). *Ear Biometrics in Human Identification, MSc thesis.* Indiana: University of Notre Dame.

APPENDIX

PROGRAM LISTING

```
clc
Rimg = input('Would you like to read images again, y/n  ','s');
clc
path = cd ;
if(Rimg == 'y')
clear
close all
r = 20; % size of the new image
cho = input('Would you like to show images during the work, y/n  ','s');
NP = 99; % persons
path = cd ;
folder = strcat(path,'\data base\Data base ANN Prof Kamil\');
trainM=[];
cd(folder);
trainMR=[];
testMR =[];
targetMR =[];
trainML=[];
testML =[];
targetML =[];
clc
for k=1:5
   k
   NR = strcat(['R (',num2str(k),').jpg']);
   NL = strcat(['L (',num2str(k),').jpg']);
for i=1:NP
    ii = sprintf('%03s',num2str(i-1)) ;
   cd(ii)
```

```
a = imread(NR);
b = rgb2gray(a);
b = imresize(b,[r r]);
%weiner filtering
wF = medfilt2(b);
b=wF;
d=b;
e = reshape(d,r^2,1);
% Normalize images
f = double(e)/255 ; % normalization
if(k==3 ||  k== 4)
   trainMR = [trainMR f];
   tg = zeros(99,1); tg(i)=1;
   targetMR = [targetMR tg];
else
   testMR = [testMR f];
end
    a = imread(NL);
b = rgb2gray(a);
b = imresize(b,[r r]);
%weiner filtering
wF = medfilt2(b);
b = edge(wF,'canny',0.01);
    d=wF;
e = reshape(d,r^2,1);
% Normalize images
f = double(e)/255 ; % normalization
    if(k==3 || k==5)
   trainML = [trainML f];
   tg = zeros(99,1); tg(i)=1;
   targetML = [targetML tg];
   else
     testML = [testML f];
   end
       if(cho == 'y' || cho == 'Y')
        subplot(221);
        imshow(Kmedian);
        subplot(222);
        imshow(BW)
        subplot(223);
        imshow(b);
        subplot(224);
        imshow(d)
        pause(0.2);
        end
      cd ..
end
clc
end
end
cd(path);
%minmax(trainMR)
net = newff(trainMR,targetMR,[600 198],{'tansig','tansig','tansig'},'traingdx'); %( for change *************
nets = init(net);
% TRAINING THE NETWORK
nets.trainParam.goal = 0.00001; % Sum-squared error goal.
nets.trainParam.lr = 0.2;% Learning Rate ( for change *************
nets.trainParam.show = 100; % Frequency of progress displays (in epochs).
nets.trainParam.epochs = 50000; % Maximum number of epochs to train.
nets.trainParam.mc = 0.3; % Momentum Factor.( for change *************
nets.trainParam.min_grad = 0;
```

```
 nets.trainParam.max_fail=6000;
 % nets.b{2} = nets.b{2}*0.01;
% nets.b{1} = nets.b{1}* 0.01;
% nets.b{3} = nets.b{3}* 0.01;
[netR,TR] = train(nets,trainMR,targetMR);
AOut = sim(netR, trainMR);
BOut = sim(netR, testMR);
count=0;
for i=1:99
            if(AOut(i,i)>=0.3)
             count=count+1;
        end
end
count/99
```