

**INTELLIGENT FACIAL EXPRESSIONS
RECOGNITION SYSTEM**

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

**By
SIHAM BESHA**

**In Partial Fulfillment of the Requirements for
the Degree of Master of Science
in
Electrical & Electronics Engineering**

NICOSIA, 2016

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:

Signature:

Date:

ACKNOWLEDGMENT

I would like to gratefully and sincerely thank Assist. Prof. Dr. Kamil Dimililer for his guidance, understanding, patience, and most importantly, his supervising during my graduate studies at Near East University. His supervision was paramount in providing a well-rounded experience consistent my long-term career goals. He encouraged me to not only grow as an experimentalist, but also as an instructor and an independent thinker. I am not sure many graduate students are given the opportunity to develop their own individuality and self-sufficiency by being allowed to work with such independence. for everything you've done for me Assist. Prof.Kamil Dimililer I thank you. I would also like to NEU Grand library administration members, since it provided me with the appropriate environment for conducting my research and writing my thesis. Additionally, I am very grateful for my family, in particular my father for his help throughout my life.

ABSTRACT

Recently, facial emotions recognition is becoming an imperative feature in the modern human computer interaction. Facial emotions recognition is to create methods and techniques that allow computers and other computerized systems to detect the facial expressions of humans by reading only their faces. In this work, we investigate the use of an intelligent system for the recognition of facial expressions. The proposed system aims to develop an intelligent system capable of recognizing the facial emotions of an image by extracting its useful features using pattern averaging and feed them into the network. Our method incorporates a series of image processing techniques prior to classification using a backpropagation neural classifier. Images are cleared of noises and their useful features are preserved using median filter that is applied on images during the preprocessing phase. The intensities of image's pixels are multiplied using the image adjustment technique, thus images get brighter. Moreover, the images undergo a technique called pattern averaging in which their size is reduced while their useful features are kept. The rescaled images are fed then into a backpropagation neural network that is capable of classifying the images into 7 facial expressions happy, angry, disgust, etc... This capability of classification is a result of a training phase according to some input parameters sets and a public database contains 210 images of 7 different facial expressions; 140 are used for training the network. The network was tested on different 70 images of same facial expressions. The experimental results show a great effectiveness, efficiency, and accuracy 93% of the developed network in recognizing the different facial expressions.

Keywords: Facial emotions recognition; intelligent system; pattern averaging; image processing techniques; median filter; backpropagation neural network; image adjustment

ÖZET

Son zamanlarda, yüz tanıma uygulamaları Modern insan bilgisayarla imbir zorunluluk özelliğine geliyor. Yüz tanıma uygulamaları bilgisayarlar ve diğer bilgisayarlı sistemler sadece yüzlerini okuyarak insanların yüz ifadelerini algılamasına izin vermektedir. Bu çalışmada, yüz ifadelerini tanıması için akıllı sistem kullanımı araştırmak. Önerilen sistem desen ortalaması kullanılarak ve içine onları beslemek için kullanılan özellikleri çıkartarak bir görüntü yüz uygulamaları yapabilen akıllı bir sistem geliştirmeyi amaçlamaktadır.

Bizim yöntem geriyayımsınır sınıflandırıcı kullanılarak önce sınıflandırmaya görüntüleme tekniği bir dizi içerir. Görüntüleri sesli olarak kullanılarak özellikleri önleme amaçlarındaki görüntüleri uygulama medyan filtresi kullanılarak korunur. Görüntünün piksel yoğunlukları görüntü ayarları kullanılarak artırılır, böylece görüntüleri daha parlak olsun. Ayrıca, görüntülerin kendi özelliklerini muhafaza ederken onların boyutu azalır hangi model ortalamaları olaraklandırılan bir tekniği sunar. Yeniden boyutlandırılmış 7 yüz ifadeleri mutlu, kızgın, tiksinti, vb. içine görüntüleri sınıflandırarak yeteneğine sahip bir geriyayımsınırın sonuçları ... Bu sınıflandırmaya yeteneği, bazı girişleri görebire itime amasının bir sonucudur parametreler setleri ve bir kamuya veritabanı 7 farklı yüz ifadeleri 210 görüntülerini içerir; 140 ayrı itime içine kullanılmaktadır. Ayrıca, aynı yüz ifadeleri farklı 70 görüntüleri üzerinde test edilmiştir. Deneysel sonuçları farklı yüz ifadelerini tanıma da büyük bir etkinlik, verimlilik ve geliştirmiş bir doğruluk % 93 göstermektedir.

Anahtar Kelimeler: Yüz tanıma uygulamaları; akıllı sistem; desen ortalaması; görüntüleme teknikleri; median filtresi; geriyayımsınır; görüntü ayarları

TABLE OF CONTENTS

ACKNOWLEDGMENT	iii
ABSTRACT.....	iv
ÖZET.....	v
TABLE OF CONTENTS	vi
LIST OF FIGURES	ix
LIST OF TABLES	xi

CHAPTER ONE: INTRODUCTION AND REVIEW OF THE THESIS

1.1 Introduction.....	1
1.2 Aim of Thesis.....	3
1.3 Overview.....	3

CHAPTER TWO:IMAGE PROCESSING

2.1 Image Analyses	5
2.2 History	5
2.3 Image Processing Application.....	6
2.3.1 Cinema.....	6
2.3.2 Medical industry	7
2.3.3 Machine vision	7
2.3.4 Digital camera images	8
2.4 Image Storage.....	8

2.4.1 How to Store Image in Computer.....	8
2.4.2 Color depth	9
2.4.3 Image file format and Size.....	9
2.4.3.1 Image file Format	9
2.4.3.2 Image size	9
2.5 Raster format	11
2.6 Different Image Processing Techniques	11
2.6.1 Image segmentation	11
2.6.2 Image compression	12
2.6.3 Edge detection.....	12
2.6.4 Canny operators	13
2.6.5 Image enhancement	14
2.6.6 Sobel operator	14
2.6.7 Top-hat transforms.....	15
2.7 Summary	16

CHAPTER THREE:ARTIFICIALNEURAL NETWORK

3.1 Overview.....	17
3.2 Introduction	17
3.3 History of Artificial neural networks	18
3.4 Analogy to the Human Brain	19
3.5 Artificial Neural Networks	21
3.5.1 Structure of ANN.....	22

3.5.2 Layers	22
3.5.3 Weights	23
3.5.4 Activation functions or transfer functions	24
3.5.4.1 Linear activation functions or ramp.....	24
3.5.4.2 Threshold function (Hard activation function)	25
3.5.4.3 Sigmoid function	25
3.5.5 Classification of ANNs.....	26
3.5.6 Training methods of ANNs	27
3.5.7 Back propagation learning algorithm	27
3.5.7.1 Modeling of back propagation algorithm	28
3.5.8 Applications of Artificial neural networks	30
3.5.8.3 Function approximation.....	31
3.5.8.4 Control	32
3.5.8.5 Filtering	33
3.6 Summary	33

CHAPTER FOUR: THE INTELLIGENT FACIAL EXPRESSIONS RECOGNITION

4.1 Images Dataset	35
4.2 The Proposed System Methodology	36
4.2.1 Flowchart of the developed system.....	37
4.3 The Image Processing Phase	38
4.3.1 Noise suppression using median filtering	39
4.3.2 Image adjustment	40
4.3.3 Size reduction.....	42
4.4 Classification Phase: Neural Network	43

4.4.1 The designed network architecture	44
4.4.2 The input setting of the developed network	45
4.4.3 The output classes coding	46
4.5 Summary	46

CHAPTER FIVE: RESULTS AND DISCUSSION

5.1 System Training	47
5.2 Experimental Results and System Performance	49
5.3 Results and discussion.....	51
5.4 Results Comparison.....	52

CHAPTER SIX: CONCLUSION

6.1 Conclusion	54
6.2 Future Recommendations	55

REFERENCES.....	56
------------------------	-----------

APPENDICES

Appendix 1: Source code.	60
Appendix 2: Source code 2	61

LIST OF FIGURES

Figure 2.1: Show image analysis	13
Figure 2.2: Show digital cinema image system	15
Figure 2.3: Shows a ct scan image showing a ruptured abdominal aortic aneurysm.....	16
Figure 2.4: Shows machine vision system	16
Figure 2.5: Shows digital camera	17
Figure 2.6: Shows digital photo workflow.....	18
Figure 2.7: Shows color images composed from 3 greyscale images.....	19
Figure 2.8: Shows images segmentation	19
Figure 2.9: Shows sample of edge detection	20
Figure 2.10: Shows sample of image enhancement	20
Figure 2.11: Shows the relationship between computer vision and other fields.....	21
Figure 2.12: Shows a color picture of a steam engine	21
Figure 2.13: Shows the sobel operator.....	23
Figure 3.1: Inter-connections between biological neurons.....	25
Figure 3.2: Structure of biological neuron.....	26
Figure 3.3: Analogy between human brain and neural networks.....	27
Figure 3.4: Basic structure of artificial neural network.....	29
Figure 3.5: Layers structure in anns.....	30
Figure 3.6: Ramp activation function	31
Figure 3.7: Hard activation function.	32
Figure 3.8: Logarithmic and hyper tangential sigmoid activation functions.....	33
Figure 3.9: Structure of ann and error back propagation.....	35

Figure 3.10: System identification using neural networks.....	38
Figure 3.11: Inverse system modeling using ann.....	39
Figure 3.12: The use of ann for control processes	40
Figure 4.1: Samples of Dataset Images.....	43
Figure 4.2: Flowchart of the designed facial emotions recognition system.....	44
Figure 4.3: Image enhancement using median filtering of angry expression image.....	46
Figure 4.4: Image enhancement using median filtering of angry expression image.....	46
Figure 4.5: Image adjustment of happy expression image.....	47
Figure 4.6: Image adjustment of surprise expression image.....	48
Figure 4.7: Size reduction of an angry expression image.....	49
Figure 4.8: Bpnn architecture.....	53
Figure 5.1: Learning curve.....	56
Figure 5.2: Regression plot.....	56
Figure 5.3: Matlab snapshot during the training phase.....	57
Figure 5.4: Correctly recognized images of the system	60

LIST OF TABLES

Table 4.1: The dataset images.....	49
Table 4.2: Input parameters.....	53
Table 4.3: Output classes.....	53
Table 5.1: Training and testing results.....	55
Table 5.2: Total recognition rate of the designed system.....	58
Table 5.3: Input parameters.....	59
Table 5.4: Results comparison.....	61

CHAPTER ONE

INTRODUCTION AND REVIEW OF THE THESIS

1.1 Introduction

Recently, growing interest in the use of all means of interaction between humans and machines has been faced. Researchers are pointed toward improving the interface and communication between the machine and the operator or user. The subject is gaining an increasing interest due to the importance of the correct decision and information transmission between a user and a machine (Fyfei, 2005). The same way humans are doing their conversations using language and body language to transmit their feelings and needs. The interest of transmitting these needs and feeling between the computer and the human is getting more and more interest. These needs and feelings are being translated by the machines in order to take the right decision and do the required tasks. The interface between humans and machines can be done using different ways such as typing, vocal interface, screens, and many other methods.

The need for more efficient and easy interfacing methods is considered very important task especially after the huge development in the artificial intelligence. That development introduces the use of emotional recognition and emotions detection to ensure the interface between the humans and the machine. Emotions are displayed by visual, vocal, and other physiological means (Fyfei, 2005). The new robots are now able to detect humans' emotions and behave according to these emotions. Researches on humans' emotions can be due to very long time ago. It has attracted a lot of researchers to study the emotions of humans and their meanings and significances. This work is sacrificed for the study of facial emotions and their detection using image processing techniques in addition to the Artificial Neural Networks. Seven different facial emotions can be basically defined, these emotions are: Angry, disgust, neutral, fear, surprise, happy, and sad. These different human emotions can be recognized using different methods from facial expressions.

Emotions detection and recognition methods are also very useful in many applications with psychological basis. As an example, they can be used to define user preferences from his own face expressions. Such methods are often employed in personality modeling and psychological

analysis. Also, e-learning can make use from these detection methods in order to improve the performance of learning based on the facial expression recognition feedback (Aman, 2007).

Since 1970s works on facial expressions and emotions recognition have appeared. Ekman and others proved with evidences that facial expressions are universal (Ekman, 2004). In his work he divided the universal face expressions to those expressions expressing happiness, sadness, fear, anger, disgust and surprise. Facial expressions from different cultures were studied and found that the face expressions and emotions are common between different races and cultures (Azkarate et al, 2005). Some differences in facial expressions were also found in some cultures (Izard, 1994). An increasing number of researches has appeared discussing the detection of human emotions based either on vocal or facial expressions. An improved performance has been proved by using systems including both facial and vocal detection (Mena, 2012).

Artificial Neural Networks are artificial, complex, self-learning, and nonlinear systems. They copy the function and structure of the human biological brain and nervous system. They can also carry out functions similar to those done by biological brain with acceptable efficiency and accuracy. Neural networks are used widely in the recent decades in classification, recognition, and security tasks. They proved their ability to perform very complex classification and recognition tasks in addition to many other functions.

The first seed of the science of artificial neural networks has appeared in the early 40s of the last century. It was a try to imitate the functions of human brain and create a machine that can learn and make decisions. The development of processing units and personal computers with very fast processing abilities opened the doors wide for very fast development of the ANNs. Artificial neural networks are employed nowadays in different aspects of science including image processing, recognition, control systems, forecasting and other fields.

In this work, the use of artificial neural networks and image processing techniques for facial emotions detection is proposed and discussed. Different image processing and filtering techniques will be applied on chosen facial expressions images are then used for the detection of different emotional expressions. The trained neural networks will be discussed and results will be tabulated.

1.2 Aim of the thesis

The facial emotions detection is a very important and promising application that is going to help for interfacing humans and machines. It is very important for new robots who can detect the emotions of human using different measures. One of these measures is the face expressions. Such robots can be used for the psychological support of humans. Smart robots and telephones can detect the psychological state of persons and suggest suitable music to listen, suitable sport to practice, or suitable actions to change or support person's mood. The goal of the thesis is to detect the expressions of different people using artificial neural networks and image processing. The system will decide whether the person is happy, sad, angry, fear, etc. The proposed work is composed of the dataset of 210 images of 10 persons. Each person has 21 images for seven different facial expressions 3 images each. These images will be processed and then used for the training and test of the ANN system. The neural network system will be used to classify them into the different expressions.

The image processing will be used in order to purify the used images by rejecting any type of noise and keeping the main features of the images. Also means of image size reduction will be used to simplify the process of training of networks. The proposed system will help in the development of smart systems that can detect the human's expressions and react based on it.

1.3 Overview of the thesis

In order to achieve the purposes meant by this work, the thesis will be divided into five main chapters as follow:

Chapter 1 is an introduction of the proposed work. This includes the importance of facial expressions recognition. The artificial neural networks and its uses and importance for face expressions recognition.

Chapter 2 is an explanation of the phase of image processing techniques and the effect of each one of these techniques. Details about processing methods and their theory are explained in this chapter.

Chapter 3 introduces the artificial neural networks. It presents the history of neural networks, the structure and functions of Artificial Neural Networks. The chapter discusses different neural network structures and functions.

Chapter 4 presents a discussion about the proposed recognition system. The discussion includes the preprocessing phase steps in addition to the neural network application. The used network is discussed and explained in details in this chapter.

Chapter 5 is the final step of this work discussing the practical results and explaining them. All the achieved results throughout this work are tabulated and discussed in this chapter of the work. Chapter 6 at the end of this chapter general conclusions and future works will be presented and suggested.

CHAPTER TWO

IMAGE PROCESSING

2.1 Image Analysis

Image analysis is the way of extract the meaningful information from image especially from digital image, this means of digital image processing techniques (Gonzales, 2008).

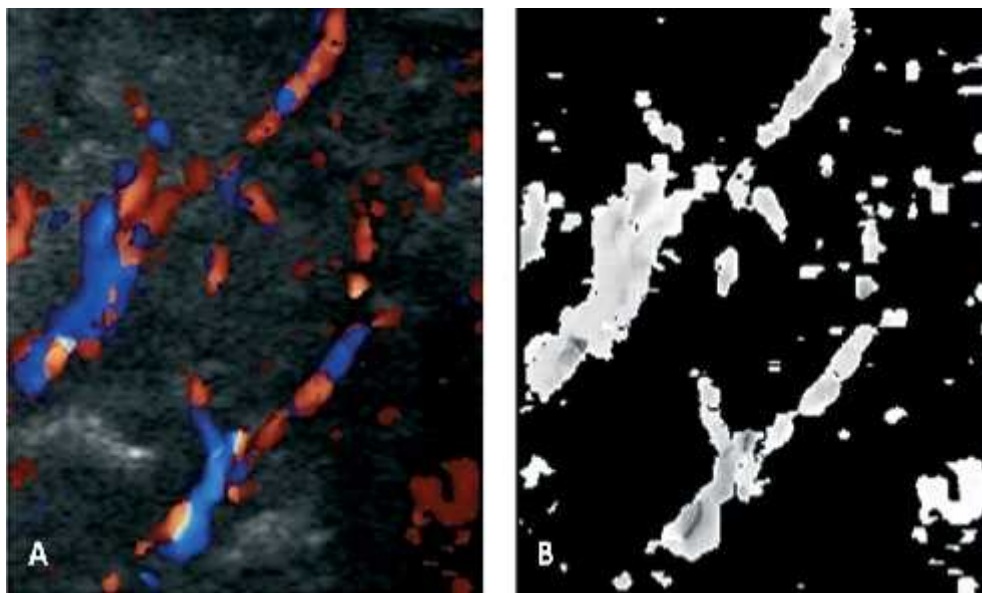


Figure 2.1:Shows image analysis

2.2 History

Early of 1920s cable picture transmission system was discovered by Bartlane, it was used to transmit newspaper images across the Atlantic. The images were coded and sent by telegraph then printed by a special telegraph printer. It took about three hours to send an image. The first systems supported 5 grey levels.

In 1964 – NASA’s Jet Propulsion Laboratory began working on computer algorithms to improve images of the moon. Image was transmitted by Ranger 7 probe.

In 1960s digital image processing was developed at the Jet Propulsion Laboratory, Massachusetts Institute of Technology, Bell Laboratories, University of Maryland (Michael, 2005).

2.3 Image Processing Application

The field of digital image has expansion in recent years. The usefulness of this technology is clear in many different disciplines (Kim et al., 1997).

The fields of image processing are:

- 1- Cinema
- 2- Medical industry
- 3- Machine vision
- 4- Digital camera images

2.3.1 Cinema

Digital Cinema is a system that used to delivercinema-quality programs to “theatres” throughout the world using digital technology.



Figure 2.2:Shows digital cinema image system

2.3.2 Medical Industry

Medical imaging is a process and art that used to create visual representations of the body for medical intervention and clinical analysis. Medical imaging seeks to the internal structures hidden by the skin and bones, so it can diagnose and treat disease (Michael, 2005).



Figure 2.3:Shows A CT scan image showing a ruptured abdominal aortic aneurysm

2.3.3 Machine vision

Machine vision (MV) is the technology or a method provides imaging-based automatic inspection and analysis for applications like control of process, robot guidance and automatic inspection in industry (Baxt, 1995).



Figure 2.4:Shows machine vision system

2.3.4 Digital camera images

A digital camera is a camera that encodes digitally digital images and videos and then stores them for later reproduction. Today most cameras are digital, and digital cameras are inserted into many devices ranging mobile phones (called camera phones) to vehicles (Baxt, 1995).



Figure 2.5:Shows digital camera

2.4 Image Storage

2.4.1 How to Store Image in Computer

We need specific care to make sure that the digital photos will not damage or lose. The environment of computer that digital photos are stored provides great opportunities and at the same time great dangers. If not properly backed up, computer failure can scratch out your digital photo collection. A small mistake in editing can overwrite the photo with a new file. To make sure that the digital photos are properly stored we required a workflow, a standard process of taking, storing, editing and archiving your digital photos (Gonzalez, 2008).

2.4.2 Color depth

Color depth, it is known as bit depth, is a number of bits used to indicate the color of a pixel, in a bitmapped image or video or the number of bits that used for a single pixel. For High Efficiency Video Coding (H.265), the bit depth specifies the number of bits used for each color (James, 2008).

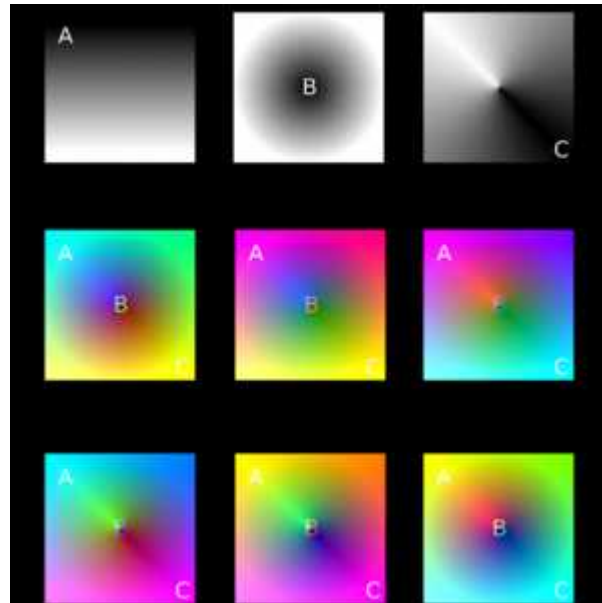


Figure 2.6:Shows color images composed from 3 grayscale images

2.4.3 Image File Format and Size

2.4.3.1 Image File Format

Image file formats are the means of organizing and storing digital images. Image files which composed of digital data can be rasterized for computer display.

2.4.3.2 Image Size

Image file size is correlated to the number of pixels in an image and the color depth, or bits per pixel, of the image. The image can be compressed in various ways(Gonzalez, 2008).

The image size depends on two things:-

a) Physical Size

The physical size of an image is based on two things: The size of the image on the screen and the file size. The file size is treated as a different issue (Gonzalez, 2008).

b) File Size

It is the size of the file on your hard drive.

Table 2.1:Shows the image formats and the size by pixel

	Color data mode -bits per pixel
JPG	RGB - 24-bits (8-bit color), Grayscale - 8-bits (only these)
TIF	Versatile, many formats supported. Mode: RGB or CMYK. 8 or 16-bits per color channel, called 8 or 16-bit "color" (24 or 48-bit RGB files). Grayscale - 8 or 16-bits, Indexed color - 1 to 8-bits, Line Art (bilevel)- 1-bit
PNG	RGB - 24 or 48-bits (called 8-bit or 16-bit "color"), Alpha channel for RGB transparency - 32 bits Grayscale - 8 or 16-bits, Indexed color - 1 to 8-bits, Line Art (bilevel) - 1-bit
GIF	Indexed color - 1 to 8-bits (8-bit indexes, limiting to only 256 colours)

2.5 Raster Format

The most common type of image files is raster graphics. Raster graphics are comprised of a grid of pixels where each one represents an individual color within the image. Web graphics and digital photos both are stored as raster graphics.

Common raster image include .BMP, .TIF, .JPG, .GIF, and .PNG (Gonzalez, 2008).

2.6 Different Image Processing Techniques

2.6.1 Image Segmentation

In computer image segmentation is the process of partitioning an image into multiple segments (sets of pixels). Segmentation is used to simplify and/or change the representation of an image into more easier to analyse. Image segmentation is used to locate (lines, curves, etc.) in images (Shapiro et al., 2000).

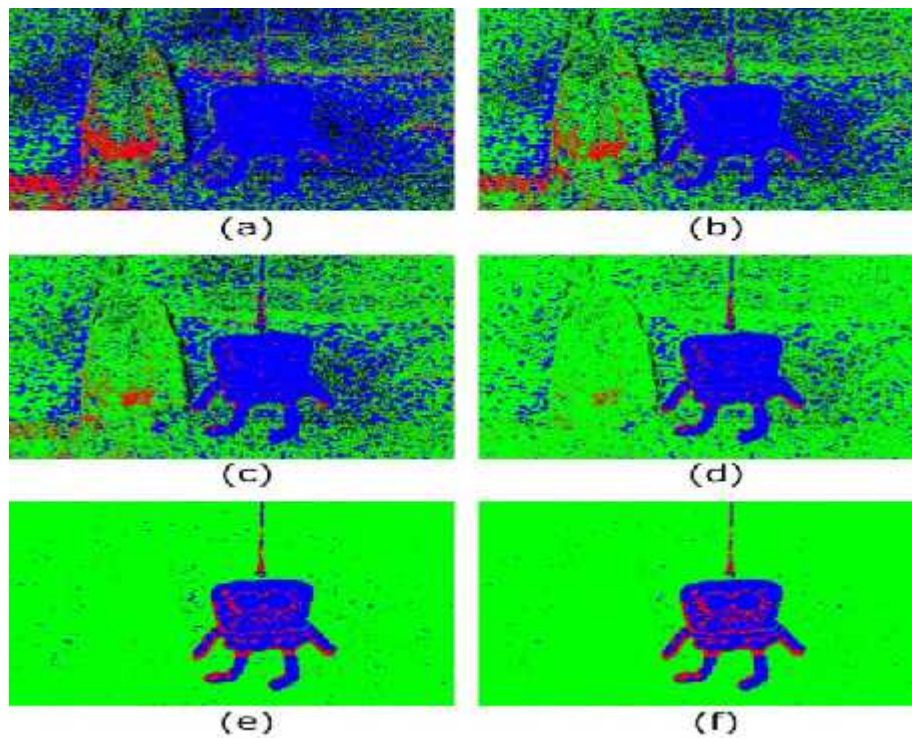


Figure 2.7:Shows images segmentation

2.6.2 Image Compression

Image compression is the way of minimizing the size of a graphics file without degrading the quality of the image (Dimililer, 2008). By reduce the file size more images can be stored in a given amount of disk or memory space. Also the time required for images to be sent will reduce over the Internet or downloaded from Web pages. There are several ways to compress the image file, the most common compressed image formats are JPEG and GIF (Dimililer,2013).

2.6.3 Edge Detection

Edge detection is a set of mathematical methods which used to identify points in a digital image at which the image brightness changes sharply. The points which changes sharply are typically organized into a curved line segments called edges (Jamil et al., 2012).

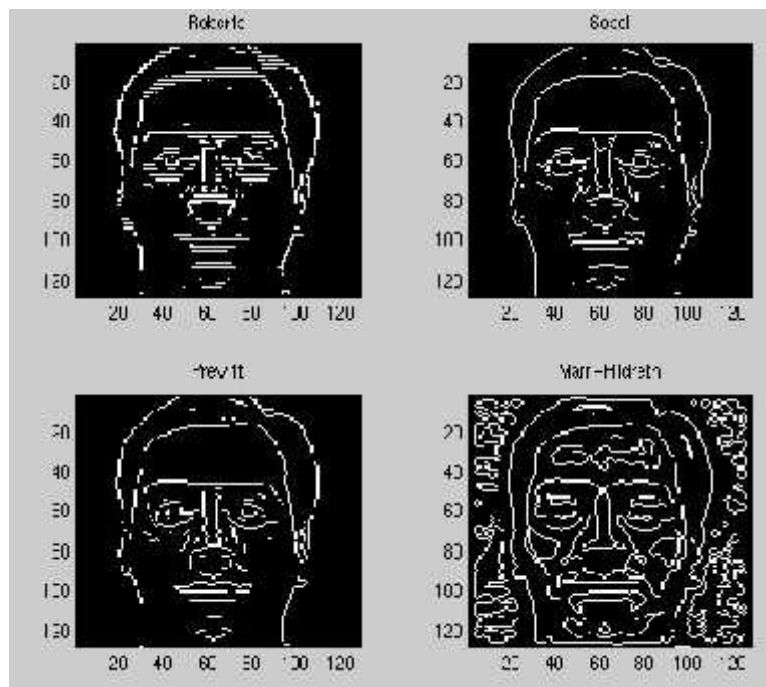


Figure 2.8:Shows sample of edge detection

3.6.4 Canny operators

The Canny edge detector is widely used in computer vision to locate sharp intensity changes and to find object boundaries in an image. Pixel edges are associated with some intensity changes or discontinuities; therefore, edge detection is the process of identifying such sharp intensity contrasts (i.e., discontinuities) in an image. Classical edge detection operators Sobel and Prewitt uses 3×3 kernels which are convolved with the original image to calculate approximations of the derivatives - one for horizontal changes, and one for vertical. In this proposed system, we detected edges using canny operators. This technique is the most common used method for detecting edges and segmenting the image. The Canny edge detector is considered as one of the best currently used edge detectors since it provides good noise immunity and detects the true edges or intensity discontinuities while preserving a minimum error (Helwan, 2004). Canny operator has been used for such algorithm with regard to the following criteria (Saif et al., 2012):

1. To maximize the signal-to-noise ratio of the gradient.
2. To ensure that the detected edge is localized as accurately as possible.
3. To minimize multiple responses to a single edge.

The steps of canny algorithm in order to segment an image into many regions are as follows:

1. Smoothing: it means blurring an image in order to remove noise and it is done by convolving the image with the Gaussian filter.
2. Finding gradients: Since the edges must be marked where the gradients of the image has large magnitudes, we have to find the gradient of the image by feeding the smoothed image through a convolution operation with the derivative of the Gaussian filtering both the vertical and horizontal directions.

Its magnitude value can be obtained using the following formula:

$$|G| = |G_x| + |G_y| \quad (2.1)$$

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.2)$$

2.6.5 Image Enhancement

In computer graphics, Image Enhancement is the process of improving the quality of a digitally stored image by treating the image with software., for example, to make an image lighter or darker. Image enhancement software also supports many filters for images (Wan et al., 1999).

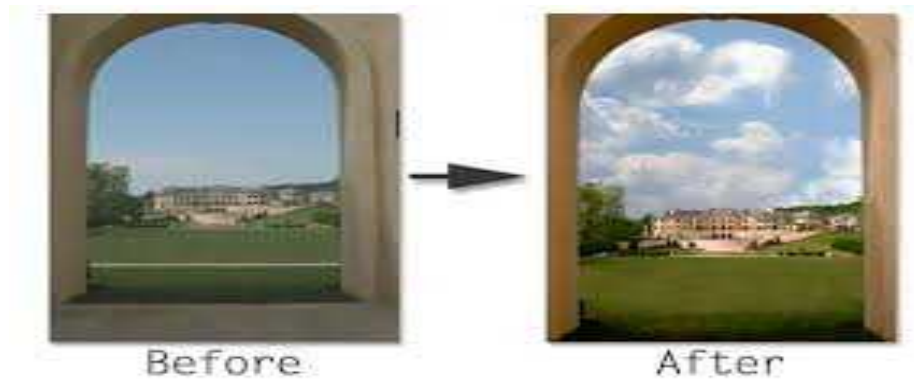


Figure 2.9:Shows sample of image enhancement

2.6.6 Sobel Operator

The Sobel operator used in computer, and image processing, especially within edge detection algorithms, which creates an image with emphasizes edges and transitions.

It is based on convolving the image within a small, separable, and integer valued filter in vertical and horizontal direction and is therefore relatively in terms of computations (Saif et al., 2012).



Figure 2.11:Shows a color picture of a steam engine



Figure 2.12:Shows the sobel operator applied to that image

The sobel operator performs a 2-D spatial gradient measurement on an image. Then, the approximate absolute gradient magnitude (edge strength) at each point can be found. The sobel operator uses a pair of 3x3 convolution masks, one estimating the gradient in the x-direction (columns) and the other estimating the gradient in the y-direction (rows)

$$|G| = |G_x| + |G_y| \quad (2.3)$$

Where G_x is the gradient in the x-direction, while G_y is the gradient in the y-direction.

2.6.7 Top-hat transforms

It is an operation which extracts small details and elements from images. There are two types of top-hat transform:

- 1- White top-hat transform is the difference between the input image and its opening .
- 2- The black top-hat transform is the difference between the input and the closing image. It is used for various image processing tasks, such as feature extraction, image enhancement, and others (Wang et al.,1999).

2.7 Summary

In this chapter the image processing and its basic techniques used in different fields such as medicine, industry, and computer vision were presented. A general review of image processing including its history and progress through time was discussed. Moreover, Different image processing techniques used for the image analysis and enhancement and segmentation were also explained such as canny and sobel operators for edge detection. Finally, the chapter presented the applications of image processing that are used in our life are very useful, especially for health, industry and security.

CHAPTER THREE

ARTIFICIAL NEURAL NETWORKS

3.1 Overview

This chapter discusses the theory of the artificial neural network. It introduces the history of the neural network and the different development stages of ANN structure. The similarities between ANN and biological brain are presented in this chapter. The structure and construction of the basic neural network is presented and its elements are discussed. The next part of this chapter discusses the back propagation learning method and presents it. Finally, different theoretical and practical functions of the artificial neural network are presented. The chapter is finished by a small summary.

3.2 Introduction

Artificial neural networks (ANNs) are the simple simulation of the structure and the function of the biological brain. The complex and accurate structure of the brain makes it able to do hard different simultaneous tasks using a very huge number of biological neurons connected together in grids. A first wave of interest in neural networks emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks (Rojas and Smagt, 1996). At that time, Von Neumann and Turing discussed interesting aspects of statistical and robust nature of brain-like information processing. But it was only in 1950s that actual hardware implementations of such networks began to be produced (Fyfe, 2005). ANNs are used widely nowadays in different branches of science. It is used for medical purposes like in (Khashman, 1999), (Dimililer, 2013), and (Dimililer, 2008). Used for image processing for different purposes like (Khashman and Demililer, 2007). It is also invested in power and power quality applications and active power filters (Valiviita, 1998) and (Sallam and Khafaga, 2002). In (Yuhong and Weihua, 2010) a survey on the application of the ANNs in forecasting financial market prices, financial crises, and stock prediction was presented.

The different mentioned applications of neural networks imply firstly the learning of the ANNs to do defined tasks. One of the most common methods of teaching ANNs to perform given tasks is the back propagation algorithm. It is based on a multi-stage dynamic system optimization method proposed by Arthur E. Bryson and Yu-Chi Ho in 1969. In 1974, it was applied in the context of ANNs through the works of Paul Werbos, David E. Rumelhart, Geoffrey E. Hinton and Ronald J. Williams, and it became famous and led to a renaissance in the field of artificial neural networks.

3.3 History of Artificial neural networks

The first real model of the theory of brain and artificial neural networks was suggested by the studies of McCulloch and Pitts (1943), Hebb (1949), and Rosenblatt (1958). McCulloch and Pitts wrote a research paper on how neurons might work. They tried to describe how human's brain is working and built an electrical circuit explaining their idea. Hebb presented the fact that neural connections are strengthened when they are used, that is the ways in which humans learn. If two nerves fire at the same time, the connection between them will be enhanced. The first model of neural networks was presented in these studies as a “computing machine”, “the basic model of network self-organization”, and “learning with a teacher” model respectively.

In 1959, Widrow and Hoff developed two models called adaptive linear elements abbreviated ADALINE and multiple ADALINES or MADALINES. ADALINES were used for binary patterns while MADALINES were the first real world application of neural networks. In 1962 they proposed a learning method where the values are being tested before the weights adjustment.

In 1972, Kohonen and Anderson proposed a similar system independently of one another. They used matrix mathematics to describe their ideas creating an array of analog ADALINE circuits. The neurons are supposed to activate a set of outputs instead of just one output.

After these works, it was until 1982 when the works on artificial neural networks were resumed again in the studies of Hopfield. He tried to create useful mechanisms by using bidirectional lines of connections (Robert, 2015). In 1986, three research groups proposed similar ideas to that of Hopfield. The so called back propagation algorithm using extended Widrow-Hoff rules with

multiple layers. The name back propagation was used for the fact that its propagating the error back through layers. The main drawback of the back propagation algorithm was its slow learning speed. It needs thousands of repeated iterations before learning (Robert, 2015).

In our days, neural networks are used in different application in many fields of science. The theory behind the development of the artificial neural network is that if it is able to work in real life, it can also work on digital computers. The future of neural networks is promising and depends on the development of hardware and processing units. The next few decades are expected to see a real expansion in the use of neural networks and extension of it in many real time applications.

Artificial neural networks are software or hardware models inspired by the structure and behavior of biological neurons and the nervous system, but after this point of inspiration all resemblance to biological systems stops. Artificial neural networks are composed of many computing units popularly called neurons. The strength of the connection or link between two neurons is called the weights. The values of the weights are true parameters and the subjects of the learning procedure. These values can be adjusted by training to perform determined tasks.

3.4 Analogy to the human brain

The artificial neural network is an imitation of the function of the human biological brain. It's using the structure and the function of brain. The human brain is composed of billions of interconnected neurons. Each one of these neurons is said to be connected to more than 10000 neighbor neurons. Figure (3.1) shows a small snip portion of the human brain where the yellow blotches are the body of the neural cells (soma). The connecting lines are the dendrites and axons that connect between the cells (Robert, 2015). The dendrites receive the electrochemical signals from the other cells and transmit it to the body of the cell. If the signals received are powerful enough to fire the neuron; the neuron will transmit another signal through the axon to the neighbor neurons in the same way. The signals are going also to be received by the connected dendrites and can fire next neurons.

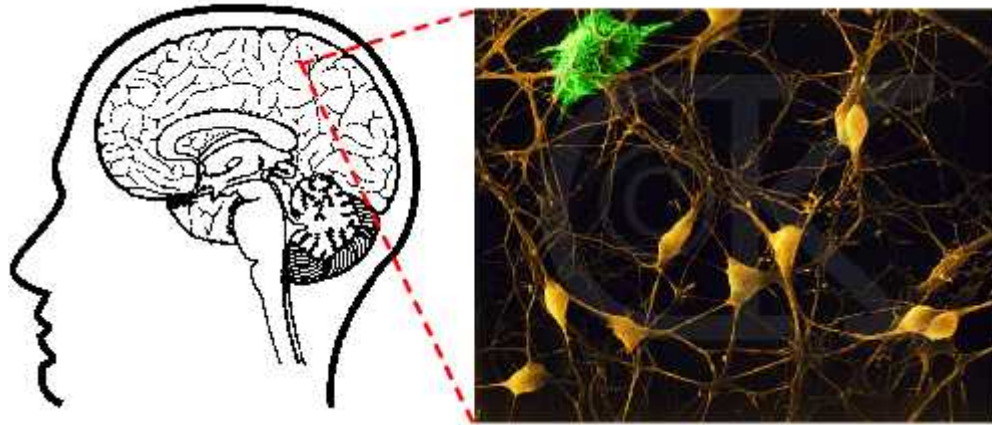


Figure 3.1: Inter-connections between biological neurons (Robert, 2015)

The whole brain is composed of these billions of connections that can be activated or deactivated based on the received electrochemical signals. It is important to notice that a neuron is activated if the sum of the signals at its inputs is more than certain level. If that sum is less than the activation level, the neuron will not be able to fire and stay deactivated. Figure 3.2 shows the different parts of the biological neuron. As seen from the figure, each neuron is composed of five parts. These are the cell body, nucleus, dendrites, synapse, and axons. Dendrites are the parts related directly to the cell body (Soma). They receive signals from the synaptic junctions. Synapses are the connection points between two different neurons. They are responsible for transition of signal between neurons in two directions. The signals are chemically transmitted in the junction points. The potential in the synapses changes based on the chemical materials being transmitted between the neurons. The potential affects the body of the cell causing its activation if it is enough powerful (Xiao, 1996).

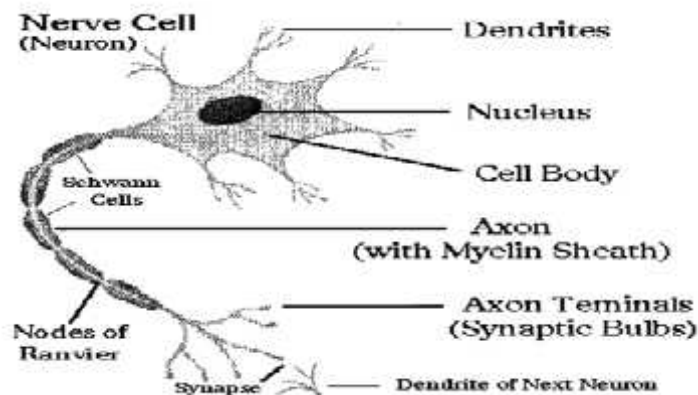


Figure 3.2: Structure of biological neuron (Kaki, 2009)

Artificial neural networks are based on the last described model of the biological neural networks. The artificial neural networks still not close to modeling the complex structure of the brain, but they have proved to be efficient in problems that are done easily by humans but difficult for classical computers. An example of these applications is image recognition and prediction based on existing data. Figure 3.3 present the relation and analogy between the human brain and the neural networks.

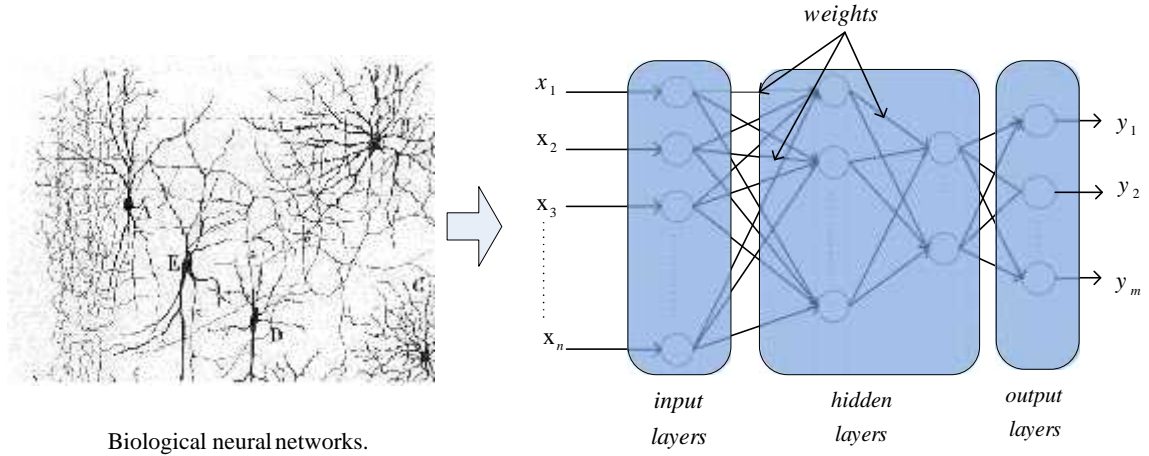


Figure 3.3: Analogy between human brain and neural networks

3.5 Artificial neural networks

Artificial neural networks are a structure that has inspired its origins from the human thinking centre or the brain. This structure has been inspired and developed to build a mechanism that can solve difficult problems in the science. Most of the structures of neural networks are similar to the biological brain in the need for training before being able to do a required task (Dimililer, 2008). Similar to the principle of the human neuron, neural network computes the sum of all its inputs. If that sum is more than a determined level, the correspondent output can then be activated. Otherwise, the output is not passed to the activation function. Figure 3.4 presents the main structure of the artificial neural network where we can see the inputs and weights in addition to the summation function and the activation function. The output function is the output of the neuron in this structure. The input of the activation function is given by:

$$TP = \sum x_n \check{S}_n \tag{3.1}$$

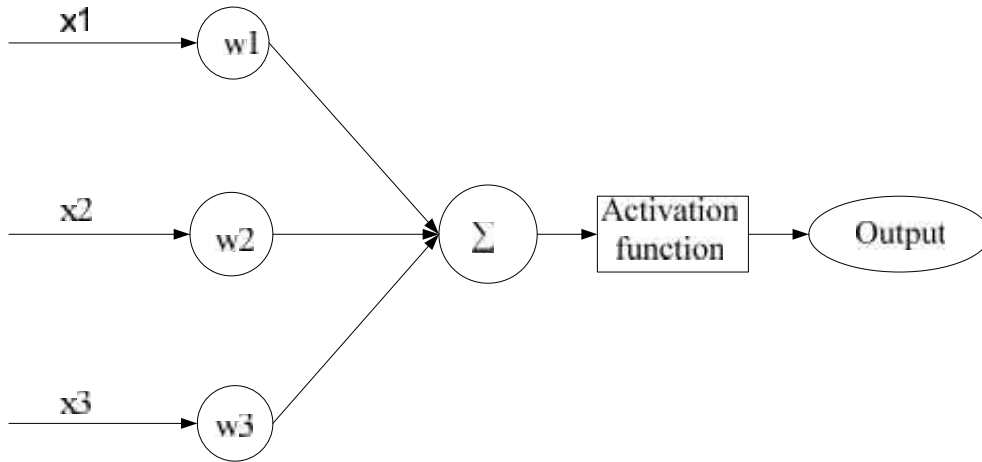


Figure 3.4:Basic structure of artificial neural network

3.5.1 Structure of ANN

The structure of ANNs consists mainly of three aspects in addition to the learning method. These aspects are the layers, weights, and activation functions. Each one of these three parts play a very important rule in the function of the ANN. The learning function is the algorithm that relates these three parts together and ensures the correct function of the network.

3.5.2 Layers

ANN is constructed by creating connections between different layers to each other. Information is being passed between the layers through the synaptic weights. In a standard structure of ANN there are three different types of layers:

- 1- Input layer: the input layer is the first one in a neural network. Its rule is the transmission of input information to the other layers. An input layer doesn't process the information; it can be considered as the sensors in biological system. It can also be called non processing layers.
- 2- Output layer: The last layer in the neural network whose output is the output of the whole network. In contrary to the input layer, the output layer is a processing layer.

- 3- Hidden layers: this is the main part of the network. It consists of one or more of processing layers. They are connecting the input layers to the output layers. Hidden layers are the main processing layers where the weights are being updated continuously. Each one of the hidden layers connects between two hidden layers or one hidden and input or output layer.

Figure 3.5 presents the layers of the neural network and the connections between the layers. As shown in the figure, the inputs are fed to the input layer. The output of the input layer is fed to the hidden layers. The output obtained from the hidden layers is fed to the output layer that generates the output of the network.

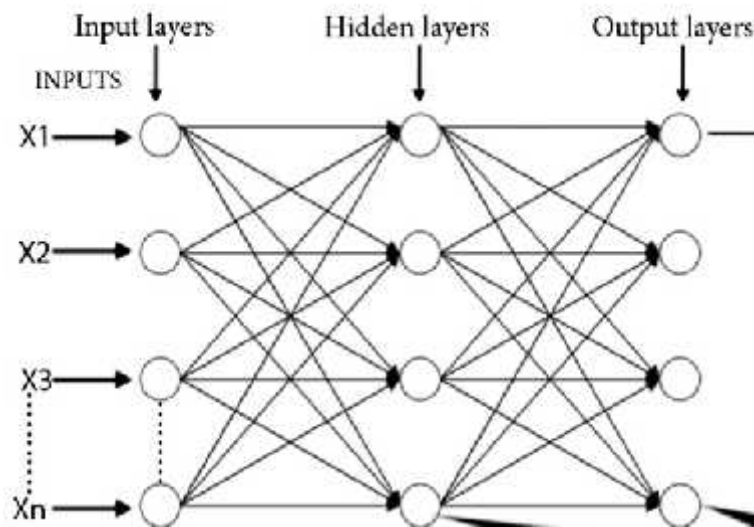


Figure 3.5: Layers structure in ANNs

3.5.3 Weights

The weights in an ANN represent the memory of that network in which all information is stocked. The values of the weights are updated continuously during the training of the network until the desired output is reached. The memory or weights are then stored to be used in future. After learning the values of these weights are used as the memory of network.

3.5.4 Activation functions or transfer functions

When the inputs are fed to the layers through the associated weights and finding the sum of them, an activation or transfer function is used to determine whether the output is to be activated or not. Or in some activation functions, the function is used to determine how much the processed input will share in constructing the total output of the network. Activation functions are very important in neural networks because they can decide whether the input to the neuron is enough to be passed to the next layer or not. There are many types of activation functions in artificial neural networks:

3.5.4.1 Linear activation functions or ramp

In this type of the activation function, the output is varies linearly when the input is small. If the input is large, the absolute output is limited by 1 as shown in figure 3.6. The function of this transfer function is defined by:

$$o(TP) = \begin{cases} -1 & TP \leq -1 \\ TP & -1 \leq TP \leq 1 \\ 1 & 1 \leq TP \end{cases} \quad (3.2)$$

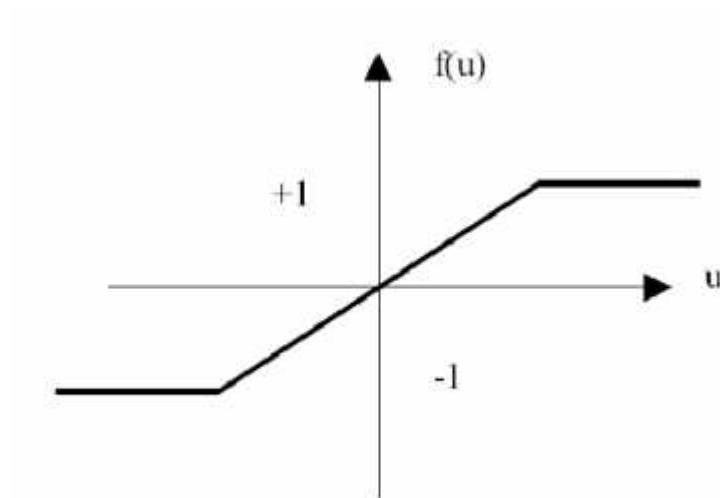


Figure 3.6: Ramp activation function

3.5.4.2 Threshold function (Hard activation function)

In the threshold function the output is zero if the summed input is less than certain value of threshold, and 1 if the summed input is greater than threshold. This way the output is oscillating between two values. It can be either activated or deactivated like in figure 3.7. The function of the hard function is defined by:

$$o(TP) = \begin{cases} 0, & TP < \theta \\ 1, & TP > \theta \end{cases} \quad (3.3)$$

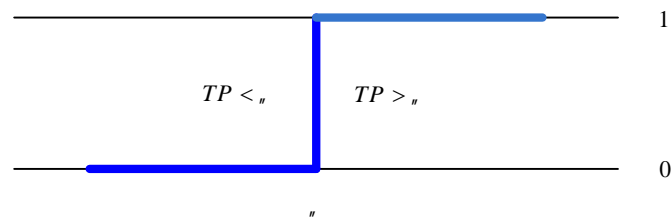


Figure 3.7: Hard activation function

3.5.4.3 Sigmoid function

This function can range between 0 and 1, but in some cases it can be useful to range it between -1 and 1. The logarithmic sigmoid and hyperbolic tangent is of the most common sigmoid functions. These two functions are the most used in the back propagation because they are differentiable. The formulas of these two functions in addition to the curves are presented in figure 3.8. The slope of the curves can be varied based on the application for which it is used (Kaki, 2009).

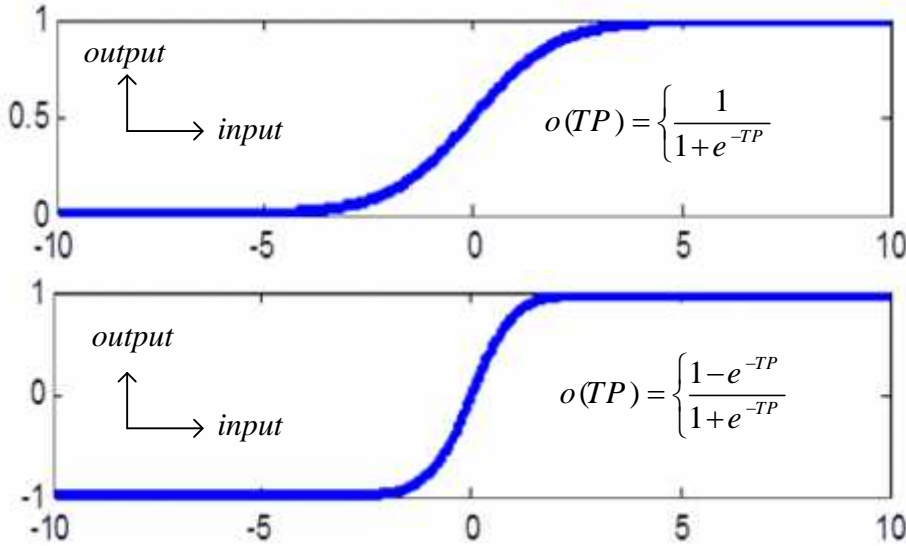


Figure 3.8: logarithmic and hyper tangential sigmoid activation functions

In the back propagation algorithms, the log-sig and tan-sig functions are the most used (Kaki, 2009). The main advantage of these two functions is the fact that they can be easily differentiated. The derivative of the logarithmic sigmoid is given by:

$$\frac{d}{dt} o(n) = o(n) * (1 - o(n)) \quad (3.4)$$

3.5.5 Classification of ANNs

ANNs can be classified based on different aspects; these are the flow of information, function or task, and the training method. The flow of information can be either from input layer toward hidden and output layers. It can also flow from next layer to the previous layer. According the function, neural networks are used to accomplish many different tasks. These tasks can be categorized into four main categories:

- Classification: where an object is assigned to a group of known categories.
- Association: linking objects to more précised categories.
- Optimization: where the task is to find the best solution for a case or problem.
- Organization.

The classification of ANNs can be done based also the training method.

3.5.6 Training methods of ANNs

Generally, the training of a network is an attempt to lead the network to converge toward desired output or outputs. Two main learning methods are used in teaching the networks. These are the supervised and the unsupervised learning method.

- **Supervised learning**

The ANN is provided by input data and desired target for this data. The network then updates its weights according to a defined algorithm rule until it converges to a minimum error or reaches a maximum number of iterations. A very important example of the supervised learning method is the error back propagation method.

- **Unsupervised learning**

In this method, the input data is provided to the network which in turn modifies its weights according to defined conditions.

3.5.7 Back propagation learning algorithm

The back propagation training algorithm uses a feed forward process, a back propagation updating method, and supervised learning topology. This algorithm was the reason of neural networks development in the 80s of the last century. Back propagation is a general purpose learning algorithm. Although it is very efficient, it is costly in terms of processing requirements for learning. A back propagation network with a given hidden layer of elements can simulate any function to any degree of accuracy (Gupta, 2006).

The back propagation algorithm is still as simple as it was in its first days. That is due to its simple principle and efficient algorithm. The input set of training data is presented at the first layer of the network, the input layer passes this data to the next layer where the processing of data happens. The results after being passed through the activation functions are then passed to the output layers. The result of the whole network is being then compared with a desired output. The error is used to make a one update of the weights preparing for a next iteration. After the adjustment of the weights, the inputs are passed again to the input, hidden, and output layers and a new error is calculated in a second iteration and vice versa.

The mentioned process continues until achieving an acceptable level of the error so that the network can be considered has learned. Figure 3.9 presents the structure of the network with layers and back propagation process.

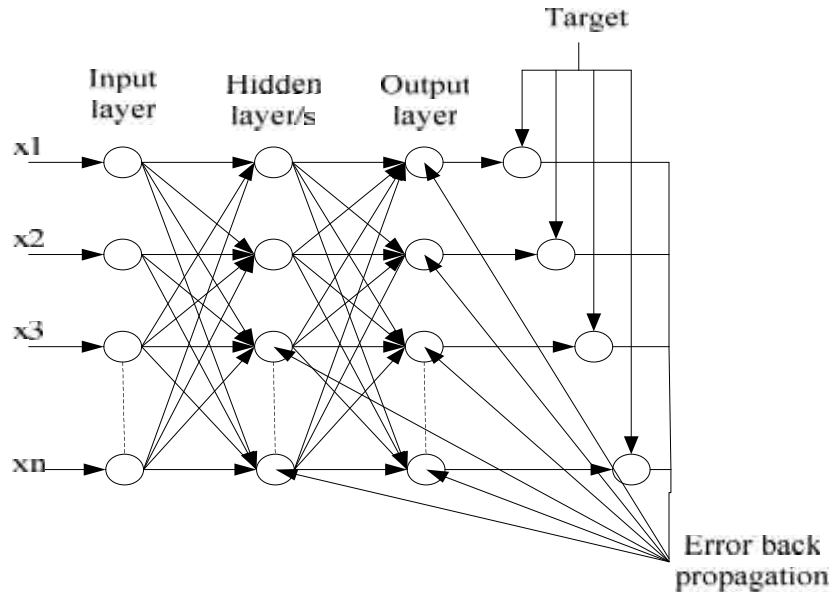


Figure 3.9: Structure of ANN and error back propagation

There are two essential parameters controlling the training of a back propagation network. The learning rate is used to control the speed of learning. It decides whether a great adjustment of weights will be done at each iteration or just small adjustments. It is important to mention here that a high learning rate is not advised because it can cause the network to memorize things instead of learning. A reasonable value of learning rate can do the job perfectly. Another parameter is the momentum factor which is used to control the oscillation of error in some local minimums. It is very important to avoid some kinds of falling into fake minimums and ensure the continuity of training (Gupta, 2006).

3.5.7.1 Modeling of back propagation algorithm

The back propagation is an algorithm that uses the theory of error minimization and gradient descent to find the least squared error. Finding the least squared error imposes the calculation of gradient of the error for each iteration. As a result, the error function must be continuous derivable function. These conditions lead to the use of continuous derivable activation functions

as they are the precedents of error calculation. In most of cases, the tangent or logarithmic sigmoid functions are used. The sigmoid function is defined by:

$$o(x) = \frac{1}{1 + e^{-ax}} \quad (3.5)$$

Where the variable “a” is a constant controlling the slope of function. Where the derivative of the sigmoid function is given by:

$$o'(x) = f(x)(1 - f(x)) \quad (3.6)$$

The equations describing the training of the network can be divided into two categories:

- 1- Feed forward calculations: used in both training and test of the network.
- 2- Error back propagation: used in training only.

In the feed forward process, the output or total potential can be given by:

$$TP = \sum x_n \xi_n + b_n \quad (3.7)$$

Where, x_n is the input vector, w_n is the weight matrix, and b_n is the bias values vector. The total potential obtained in each layer must be passed by an activation function. The activation function can be either linear or non-linear function. An example of a linear function that is mostly used in neural networks is the sigmoid function given in equation (2.5). Another example is the tangent sigmoid given by:

$$o(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.8)$$

It is important to notice that this function is also continuous and derivable. The derivative of this function is given by:

$$o'(x) = 1 - \frac{(e^x - e^{-x})^2}{(e^x + e^{-x})^2} \quad (3.9)$$

The output of the last activation function is the actual output of the neural network. This output is then compared with the goal of training to generate the error signal. The error signal is actually defined by equation (3.10). The goal of the training of neural network is always to minimize that error.

$$E = \sum (T - o)^2 \quad (3.10)$$

Where, T signifies the target output. An error function is then defined based on the value of E such that:

$$\Delta_j = (T_j - o_j)o_j(1 - o_j) \quad (3.11)$$

This value is propagated back to the network using the next equations to update the weights and biases of the different layers. The weights are then updated using the next equation:

$$\check{S}_{jhnew} = \check{S}_{jhold} + y\Delta_j o_h + r(u\check{S}_{jhold}) \quad (3.12)$$

Concerning the hidden layers, their weights are updated using the error update defined by:

$$\Delta_h = o_h(1 - o_h) \sum \check{S}_{jh} \Delta_j \quad (3.13)$$

The new weights values are then given by:

$$\check{S}_{hinew} = \check{S}_{hiold} + \sim \Delta_h o_i + r(u\check{S}_{hiold}) \quad (3.14)$$

The values of \sim and r are the well-known momentum factor and learning rate. At the end of weights update, a new feed forward iteration is done again. The error is being calculated at each iteration until it arrives an accepted error value.

3.5.8 Applications of Artificial neural networks

ANNs are used in different fields of science in many applications these days. In some applications they are still in the research mode. The neural network technology is a promising field for the near future. In this part of our work, different fields of application of ANN will be discussed. The neural networks are used mainly in pattern recognition, pattern association, function approximation, control systems, beam forming, and memory (Hykin, 1999).

3.5.8.1 Pattern association

It is a brain like distributed memory that learns by association. Auto association is a process where the neural network is supposed to store a set of vectors by presenting them to the network. In a hetero association structure, a set of inputs is being associated with an arbitrary set of outputs. The hetero association is supervised learning process.

3.5.8.2 Pattern recognition

Pattern recognition is a simple task done by humans in their everyday life with merely no effort. Simply, we can recognize the smell of some food that we have tasted before easily. Familiar persons can be recognized even if they are aged or their expressions have been changed since last time we saw. Pattern recognition is known as a process by which a received signal can be assigned to a prescribed number of categories (Hykin, 1999). Although pattern recognition tasks are very easy for humans, they are very difficult to be carried out using traditional computers. The neural networks have presented an excellent approach for carrying out pattern recognition tasks using computing machines.

A well trained network can easily recognize and classify a pattern or group of patterns to classes. Face recognition, fingerprint recognition, voice recognition, iris recognition and many other applications are examples of pattern recognition.

3.5.8.3 Function approximation

Interpolation and function approximation has been a very important field of numerical mathematics. It is very to determine the function describing the relation between discrete variables. Related set of input output numerical association can be modeled using linear or nonlinear functions. Neural networks can be used to describe the relation between input and output variables of the set. Neural networks can approximate function in two different ways:

1- System identification: Figure 3.10 shows the scheme of system identification task. If we have an unknown system that we need to model, a neural network can be associated with the system. The input output relationship of the system can then be modeled by the neural network during the training. The weights of the neural networks are updated until it will produce the same output of the system if subjected to the same input.

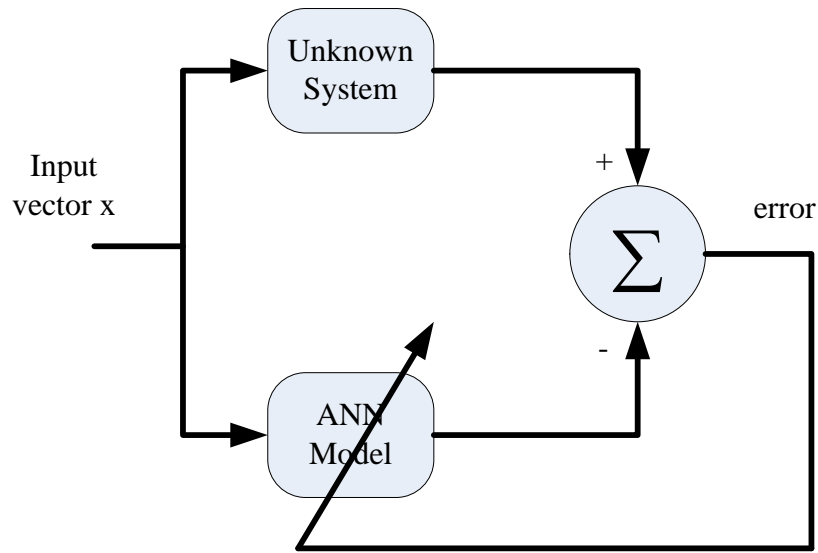


Figure 3.10: System identification using neural networks (Hykin, 1999)

2- Inverse system modeling: as shown in Figure 3.11 the inverse of the system to the left can be modeled using ANN. After training the input of the whole system should be equal to its output.

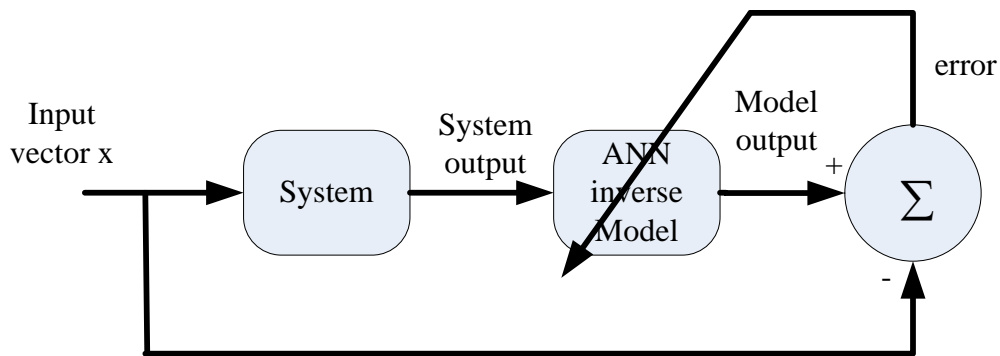


Figure 3.11: inverse system modeling using ANN

3.5.8.4 Control

The control of processes is another learning task neural networks can do. The brain is evidence that a distributed neural network can be used in the systems control. If we consider a feed-back process like the one shown in Figure 3.12, the system is using a unity feed-back to control the process. The plant output is fed back to the control that compares output with the desired output. A neural network controller can be used to generate the appropriate control of the plant.

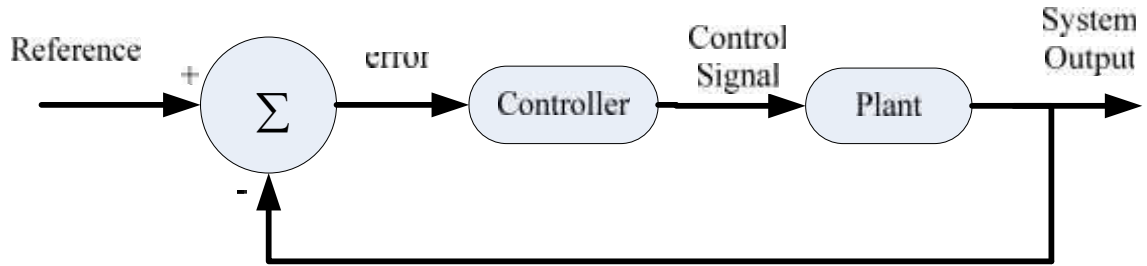


Figure 3.12: the use of ANN for control processes

3.5.8.5 Filtering

The term filtering is referred to the process or algorithm by which prescribed data is been extracted from noisy data. The noise is being then rejected. Filtering is a very important task. Noise rejection in microphones and speakers, in telephones, stereos, digital communication devices and many other communication means is done using filtering. A simple description of the filtering task using artificial neural networks is shown in the Figure 3.13.

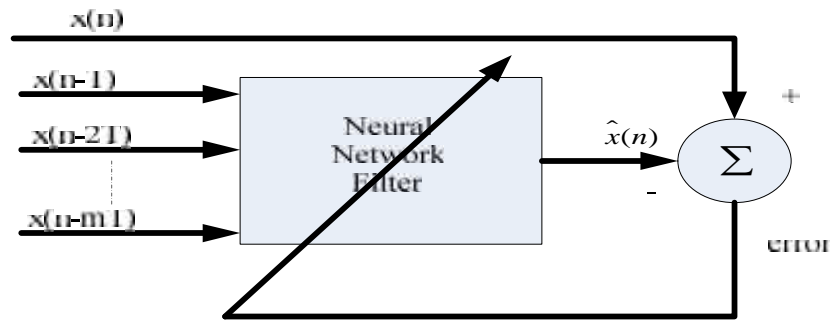


Figure 3.13: Filtering using artificial neural networks

3.10 Summary

This chapter discussed the theory of Artificial Neural Network. A brief historical review of ANNs and there development was presented at the beginning of the chapter. Different structures of artificial neural networks and their elements were presented. A detailed functional and structural comparison between the artificial neural networks and human neural network was discussed.

The supervised and non-supervised learning methods of ANNs were also presented. Due to its efficiency and ability to perform different tasks, the back propagation algorithm was also discussed and presented. At the end of the chapter, the main different applications of the neural networks were presented and discussed briefly.

CHAPTER FOUR

THE INTELLIGENT FACIAL EXPRESSIONS RECOGNITION SYSTEM

4.1 Images dataset

The images were obtained from the online public resource available on the internet: JAFFE Japanese Female Facial Expression (Yorozu, 1997). The database contains 10 unique females. Each one has 7 poses for 7 different expressions. Therefore, a total of 210 images are used for this developed system. Sample images are shown in Figure 4.1.

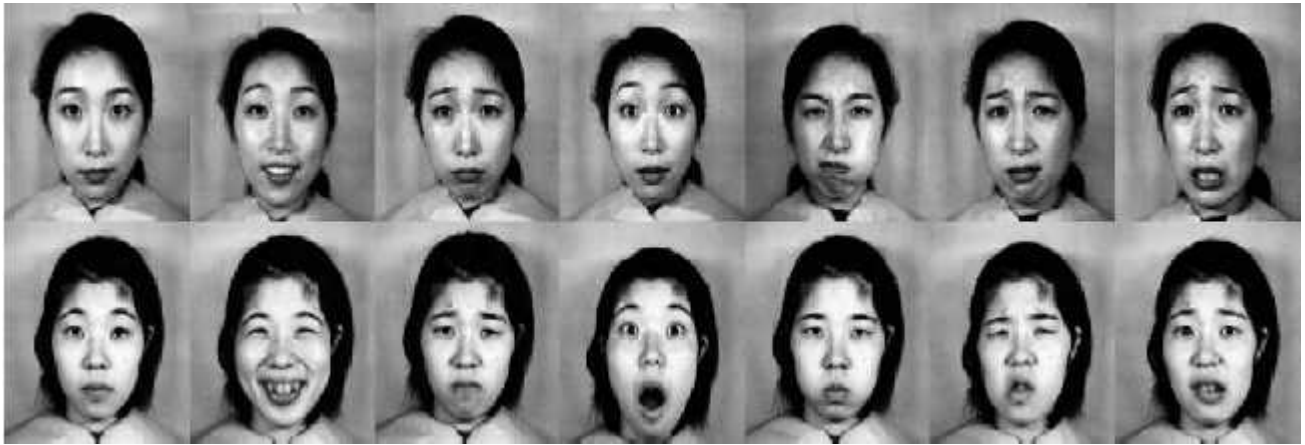


Figure 4.1: Samples of dataset images

Table 4.1: The dataset images

Facial expression	Nb.of expression per females	Nb. Of poses per expression	Number of females
angry	7	21	10
sad	7	21	10
happy	7	21	10
neutral	7	21	10
surprise	7	21	10
disgust	7	21	10
fear	7	21	10
Total	7	210	10

As Table 4.1 shows, a total of 210 images are used for the developed system including 21 different expressions posed by 10 different Japanese females.

4.2 The Proposed System Methodology

This thesis presents the use of backpropagation neural network in order to recognize different facial expressions such as happy, sad, disgust etc.

The images are not directly fed into the neural network, however, an image processing phase takes place first in order for the used images to be filtered and remove some useless components.

In this image processing phase the images are filtered for the purpose of removing noise and keep the main features of images. Moreover, Pattern averaging technique is applied to images in this phase for the purpose of size reduction that would simplify the process of training of networks. Once the images are processed, they are fed into a neural network and then the training phase starts using the adaptive learning algorithm: backpropagation algorithm. Once the mean square error is minimized and the network converges, the training phase stops. After training, the network is tested using different facial images that represent the same facial expressions such as happy, sad, disgust etc.

The images used for training and testing purposes of the network are obtained from the adapted version of JAFFE database (the Japanese Female Facial Expression Database).

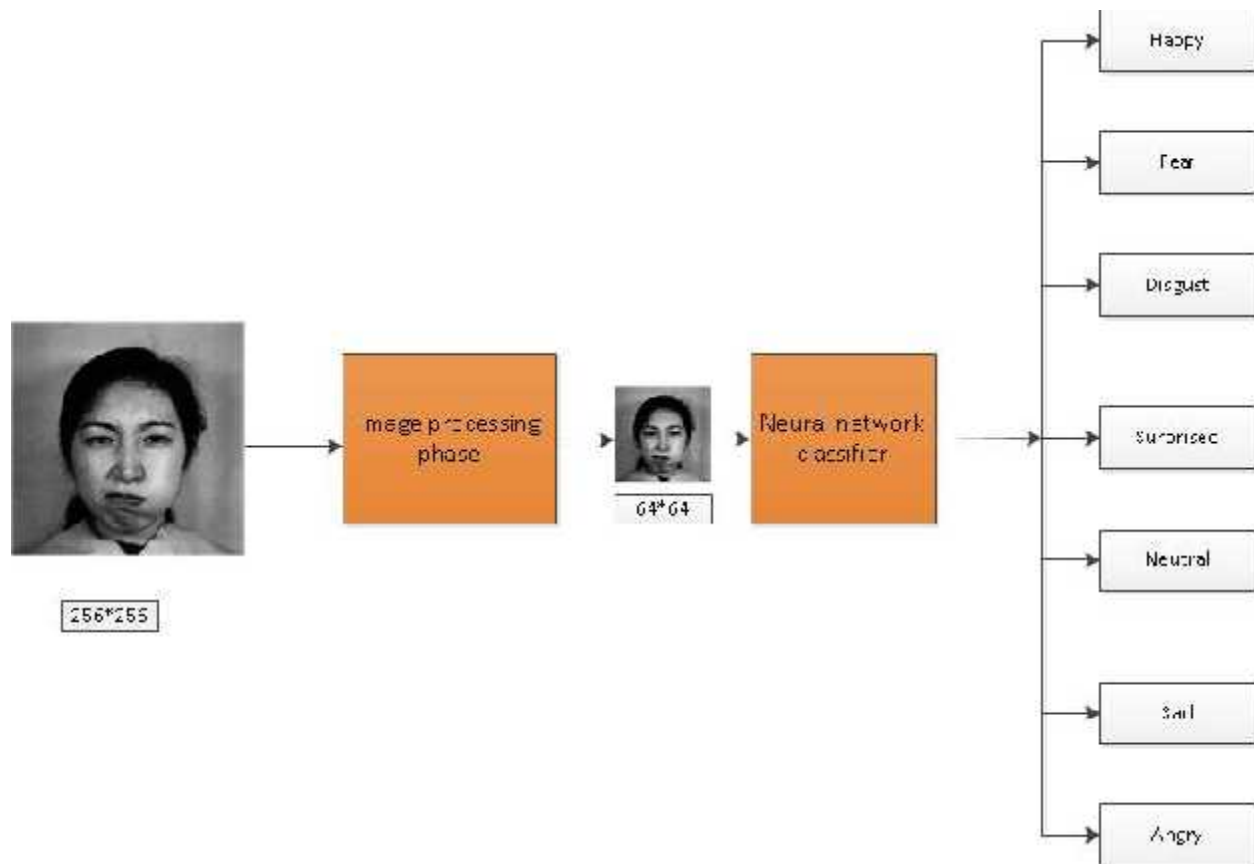


Figure 4.2: The proposed system design and phases

The figure above illustrates the system designed including the two phases: image processing and classification phase. It can be seen that the images are first filtered and adjusted using different filters and adjustments techniques. Then, the size of images is reduced to 64*64 using a method called patten averaging which is discussed in details in section 4.4.3.

Once the size of images is adjusted, the images are then fed into network and the training phase starts.

4.2.1 Flowchart of the developed system

The flowchart below shows the developed system methodology, in which images are processed using some image processing techniques in order to be ready to be fed into the neural network

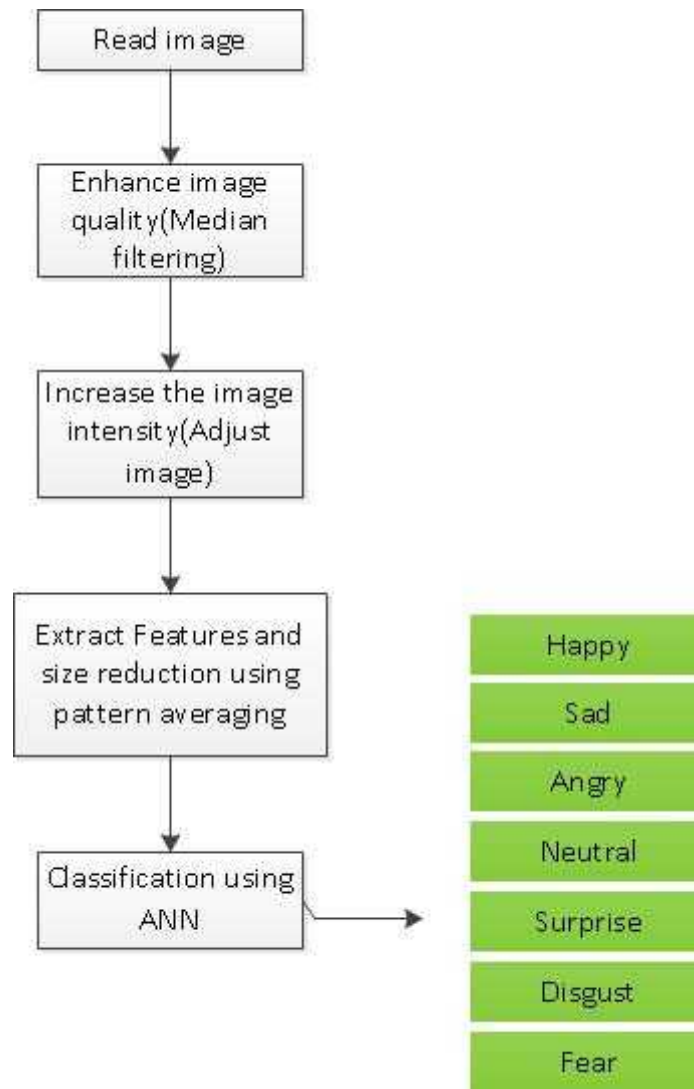


Figure 4.3: Flowchart of the designed facial emotions recognition system

4.3 The Image Processing Phase

The images were enhanced for adequate classification through a series of image processing methods and techniques. These methods constitute the adequacy of the system in the processing phase.

Different image processing techniques are used to obtain high quality and less dimensionality images.

4.3.1 Noise suppression using median filtering

Image enhancement is an image processing technique used in order to suppress the noise in an image to produce less pixelated and clearer image. Most smoothing techniques are based on low pass linear filters. It is mostly based on the averaging technique of the input image or the middle (median) value technique (Abiyev, 2012).

In order to enhance the quality of images while keeping the main features, we applied a linear filter to the images. The most common type of smoothing filters is the linear filters such as median filter which is used in our system. This filter is used to reduce impulsive noise or the salt-and pepper in an image with preserving the useful features and image edges. Median filtering is a linear process in which the output of the being processed pixel is found by calculating the median of a window of pixels that surrounds that studied pixel (Helwan, 2014). In other words, the median filter goes through each element of the image and replaces each pixel with the median of its neighboring pixels which are located in a square neighborhood (kernel) around the evaluated pixel. Median filtering smooths the data while keeping the small and sharp details.

James Church et al. (James et al., 2008) have investigated the use of median filter for the noise removal in digital images. In their research, a comparison of different smoothing filter was presented, and the medial filter was the best to use since it has many good impacts on the images such as:

Avoid handling the limits, with or without editing the sign or image limit a while later.

- Fetching sections from different places in the sign. With images for instance, sections from the far even or vertical limit may be chosen.
- Shrinking the window close to the limits, so that each window is full.

Figure below illustrates the application of a median filter and its mechanism to reduce the noise on one of the database images by setting a kernel or window of 4*4.

It can be noticed that the noise was removed and the image is clearer and smoothed after applying the median filter.

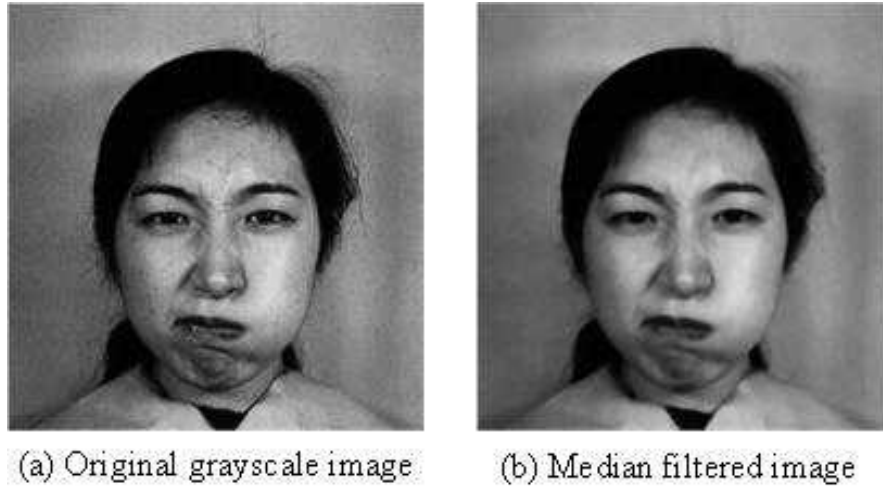


Figure 4.4: Image enhancement using median filtering of angry expression image

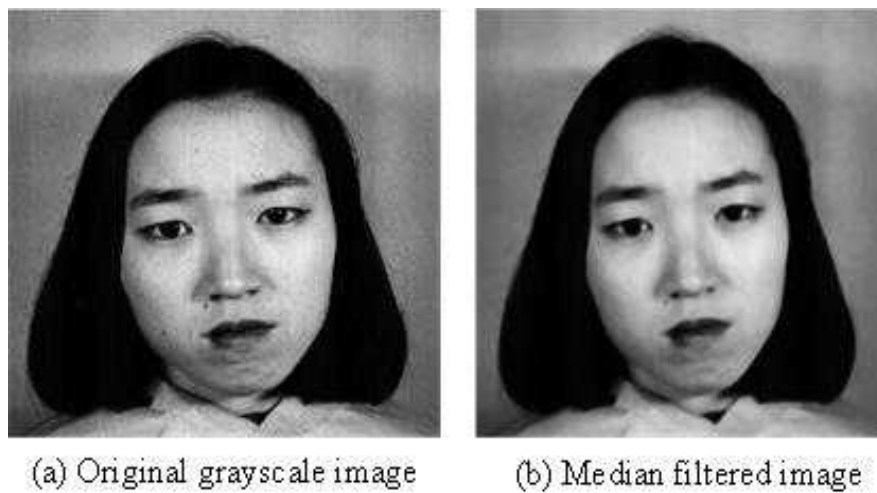


Figure 4.5: Image enhancement using median filtering of angry expression image

4.3.2 Image Adjustment

Image adjustment is to increase the intensity of images pixels by mapping the input image's intensities to a new range of in the output image. This can be done by setting the low and high input intensity values that should be mapped and the scale over which they should be mapped (Gonzalez and Woods, 2002). This image processing technique that aims to enhance the contrast of the image by increasing the intensity of its pixels.

During this operation, the intensity value of each pixel in the input image is transformed using a transfer function to form a contrast adjusted image. Gamma contrast adjustment is the most common used transfer function.

Ankit Aggarwal et al. developed a system for the adaptive image enhancement technique preserving brightness level using gamma correction. Their proposed technique is that the weighted average of the histogram leveled, gamma corrected and the first picture are consolidated to acquire the upgraded processed image. The proposed calculation accomplish contrast enhancement as well as preserve the brightness level of images (Aggarwal et al., 2013).

Figures 4.5 and 4.6 show the use of this image processing techniques used in the developed system running on happy and surprise images from the used database.

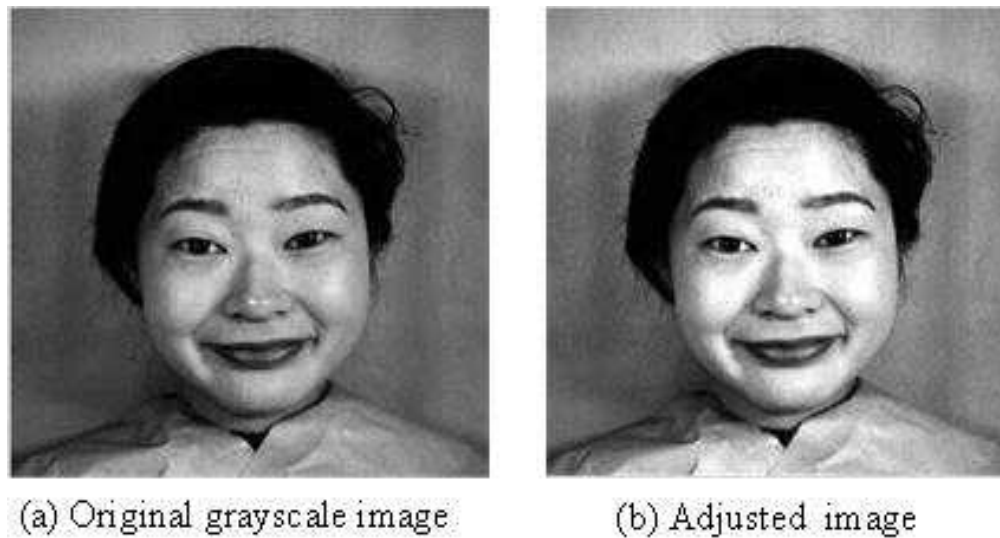


Figure 4.6: Image adjustment of happy expression image

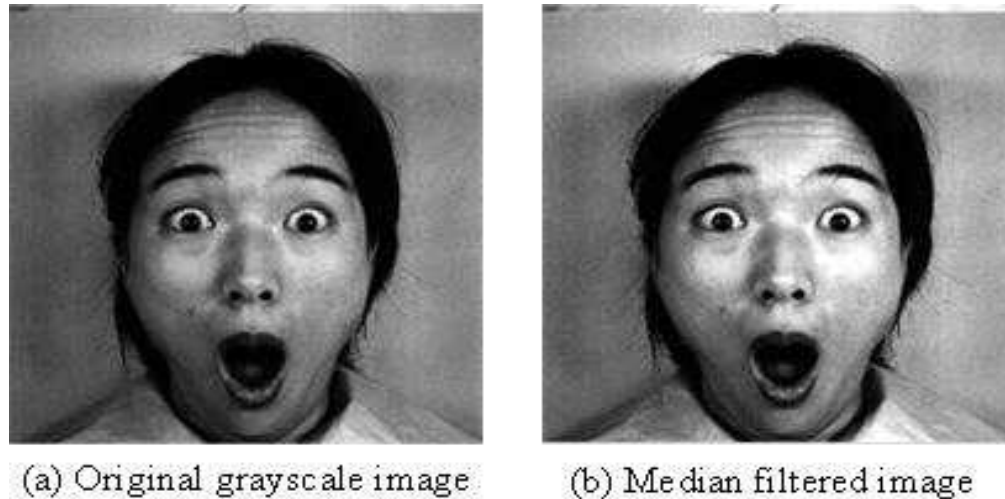


Figure 4.7: Image adjustment of surprise expression image

4.3.3 Size Reduction

Once the images are filtered and adjusted, the rescaling of size of the images take place. The purpose of size reduction is to remove the unneeded features of image while preserving the main and key-features. Moreover, the size reduction of images reduces the number of input neurons of the network so that the generalization capability of network improves. To reduce the size of images while keeping the useful and needed features extracted by the previously used methods, we used pattern averaging technique. This technique is to select a window of 4*4 segments. Then, each studied pixel is then the average of the 16 neighbor's pixels in the selected window. Thus, we come up with a rescaled image with the same features and properties of the original one; however, with a smaller size, for the purpose of fast processing and easy computing.

A system for the Intelligent Classification of Sonar Images was developed by Boran Sekeroglu et al. (2006) for the identification of the three blood cells. The authors used pattern averaging to reduce the size of the coins while preserving the needed features. An intelligent blood cell identification system was developed by Adnan Khashman (Khashman, 2008) for the identification of the three blood cells. The authors used pattern averaging to reduce the size of the blood cell images while preserving the needed features. The technique of pattern averaging was used to achieve the rescaling of images, and the formula used to achieve this is given in Equation 4.1.

$$Seg_i = (\sum P[x, y]) / D \quad (4.1)$$

$$D = (TP_x \cdot TP_y) / S \quad (4.2)$$

Where Seg denotes the segment number, P denotes the pixel value and D is the total number of pixels in each segment; TP denotes the x and y pixel size of image and S is the total segment number (Kashman, 2013).

In Figure 3.7 an example of a processed rescaled angry expression image using pattern averaging is shown.



Figure 4.8: Size reduction of an angry expression image

4.4 Classification Phase: Neural Network

The developed system aims to intelligently recognize or detect the facial emotions of females, posing different expressions. A neural network is used as a classifier in this system since its effectiveness in these tasks. A neural network is a modelling of the human brain information processing system that is capable of performing different tasks if it learns well. It is an adaptive learning technique in which it has a different and specific learning methodology; the learning by examples. Therefore; some complex tasks can be handled using neural networks such as prediction, recognition, and classification (Rojas, 1996). Thus, there should be a learning algorithm that trains the used network to be then able to recognize the different facial emotions. A backpropagation learning algorithm is used in our system due to its simplicity and the

sufficient number of images. Backpropagation is an adaptive learning technique for the feed forward multilayer neural networks. It has two passes through the different layers; the forward pass and the backward pass. In the forward pass the weights are summed and then combined in the output layer. In the backward pass the weights are corrected. The actual output is subtracted from the desired one in order to produce the error. The error is then propagated back to all previous layers in order to update the weights and get the desired output (Al-Milli, 2013)

4.4.1 The designed network architecture

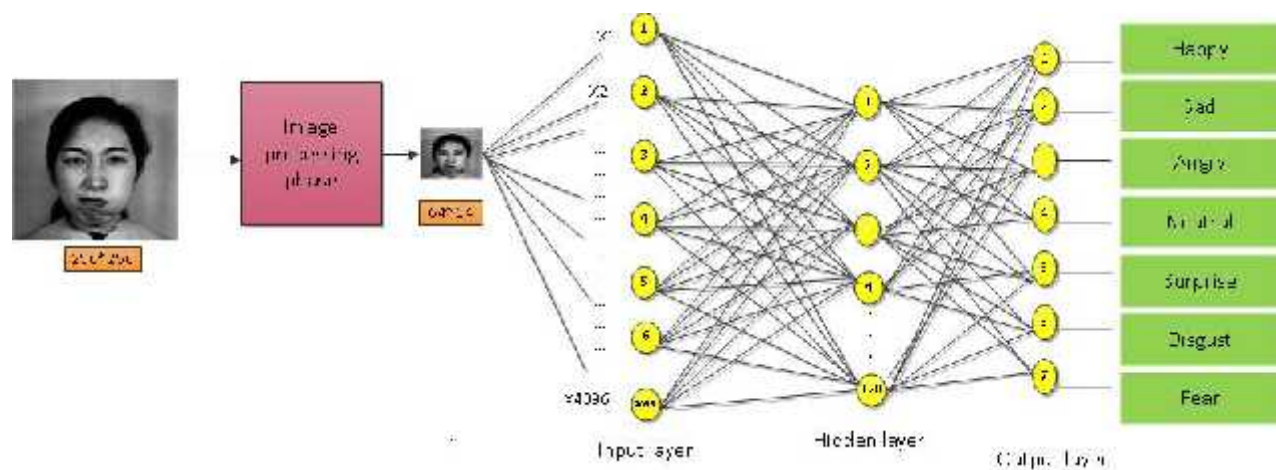


Figure 4.9: BPNN architecture

$X_1 \dots X_n$ represent inputs of the network. As the figure above represents, the developed system neural network comprises of 4096 neurons in the input layer since the input images are of size 64*64. The hidden layer is composed of 120 neurons which are enough for getting least mean square error and least processing time. The output layer consists of 7 neurons since the networks is meant to classify 7 different classes: happy, angry etc..

4.4.2 The input setting of the developed network

The network was created on the Matlab software using the back propagation algorithm. The first step was to create a basic network and train it for simple operation such as “AND” or “OR” in order to reduce the mean sum error value to 0.001. The backpropagation learning algorithm was used with the adaptive learning rate and momentum rate for training network; with the function ‘traingdx’ and with the transfer function “logsig”. The network was fed with the normalized datasets for the three sets and their output targets respectively. The network ran for 50000 iterations in the training phase. The table below shows the input parameters and their corresponding values

Table 4.2: Input parameters

Input parameters	Corresponding values
Number of input neurons	4096
Number of hidden neurons	120
Number of output neurons	7
Iteration number	50000
Momentum rate	0.06
Learning rate	0.04
Error	0.001
Training time (sec)	250
Activation Function	sigmoid

4.4.3 The output classes coding

The table 4.3 shows the output classes of the designed facial emotions recognition system. As noticed they are 7 classes since the system is meant to recognize 7 different facial expressions.

Table 4.3: The Output Classes

Output classes	Coding representations
Happy	1 0 0 0 0 0 0
Sad	0 1 0 0 0 0 0
Angry	0 0 1 0 0 0 0
Neutral	0 0 0 1 0 0 0
Surprise	0 0 0 0 1 0 0
Disgust	0 0 0 0 0 1 0
Fear	0 0 0 0 0 0 1

4.5 Summary

In this chapter, both the first and second phases (image processing and neural network) of the developed facial emotions recognition system were presented. We presented the developed algorithm of the system through a flowchart that lists the image processing techniques used in order to extract the patterns of interests and make images ready to be fed into the network. Moreover, the neural network input parameters and also the output classes were revealed through tables and figures.

CHAPTER FIVE

RESULTS AND DISCUSSION

5.1 System Training

To create an efficient network that can be capable of such classification task, it is best to train the network on simple tasks first. To do this, the network is first trained on ideal vectors until it has a low sum squared error. In order to decrease the error value to 0.001, we have to start training the neural network on simple and mathematical operations such as AND or OR operations.

The training is done using backpropagation learning algorithm with both adaptive learning rate and momentums, with the function '*traingd*'. After making sure that the error is minimized; we started feeding the neural network with the input images and their targets respectively.

The network was trained on 140 images of 10 persons. Each person has 21 images for seven different facial expressions 3 images each. The table 5.1 represents the total number of images used in both training and testing phase. It also shows the total number of facial expressions posed per female.

Table 5.1: Training and testing sets

Females number	Total nb. Of Facial expressions per female	Total number of images	Training sets	Training time	Testing sets
10	21	210	140	811 sec	70

Figure 5.1 illustrates the learning curve of the developed system that is based on a backpropagation neural network. The network was trained on a large number of images obtained from the JAFFE database for Japanese facial expression images. Thus, the result is perfect and the network was well trained since the mean square error was diminishing as much as the number of iterations was rising.

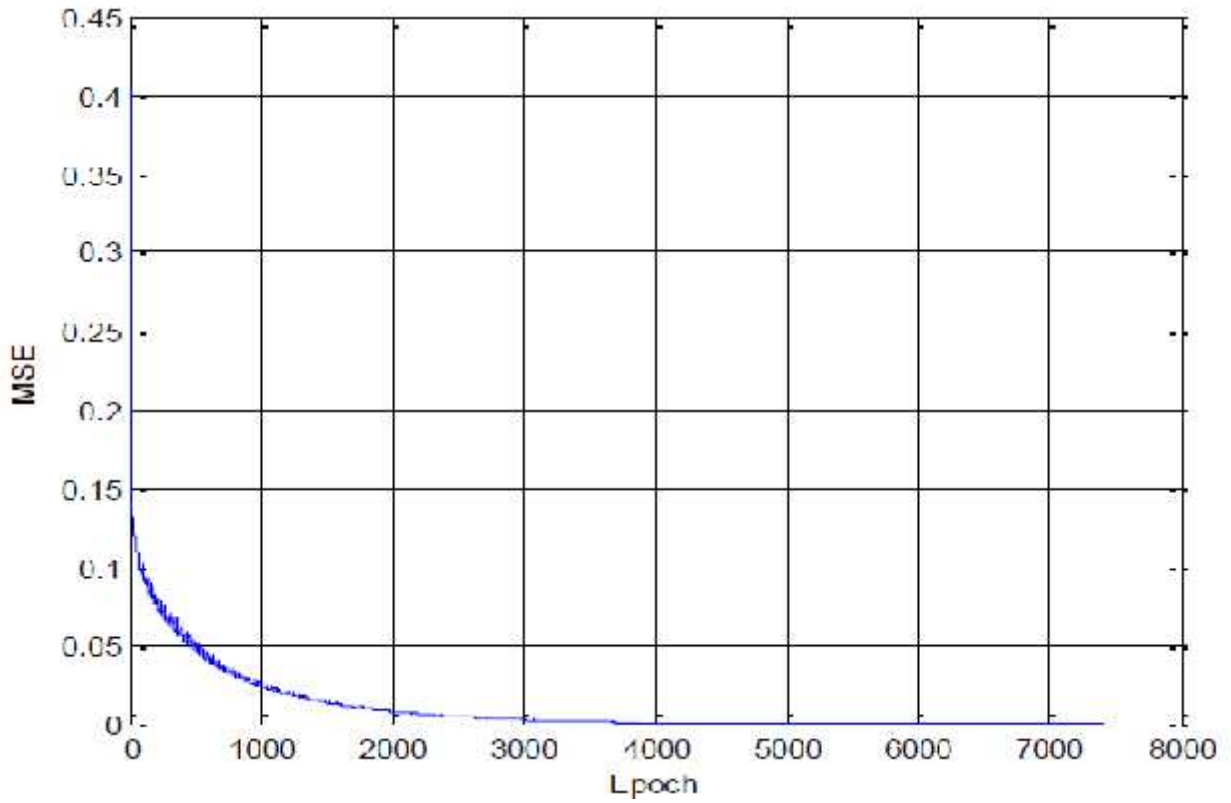


Figure 5.1: Learning curve of the developed system: MSE versus iteration number

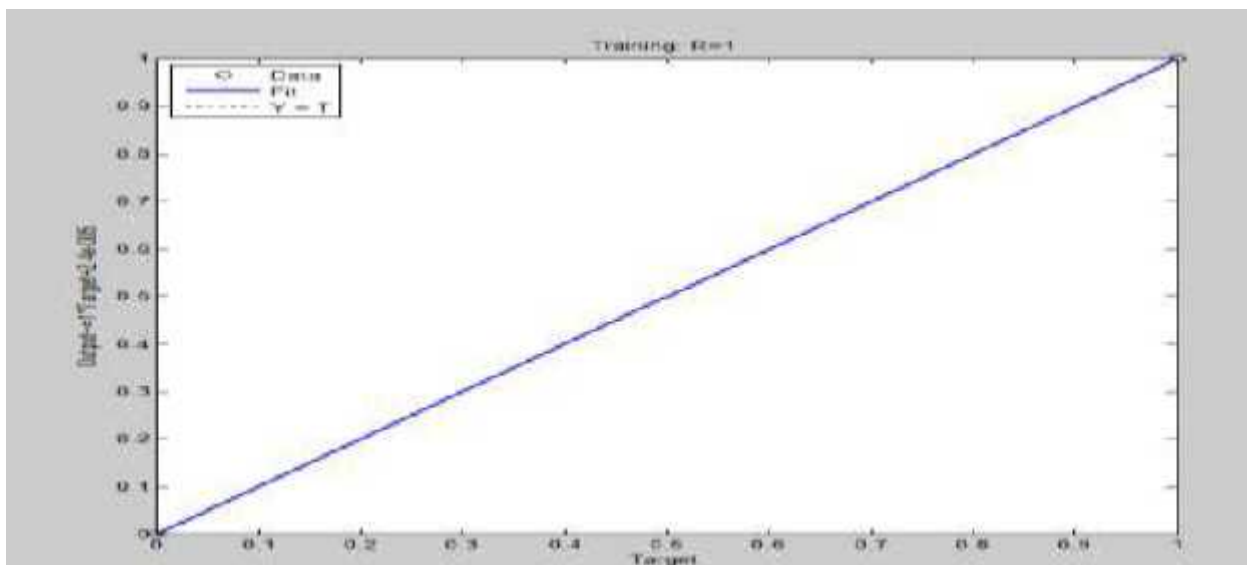


Figure 5.2: Regression plot of the system: Actual versus desired output

This figure above represents the regression plot of the desired output (dotted line) and the actual output. As the actual output is far from the target as the error is increased. In this figure, it is

remarked that the target and the actual output are overlapped which means that the error is minimized and the network well trained (training ratio = 100 %).

No. Training Images	Training set	No. of females	Facial expressions per female
Training Image	24	10	FE
Training Image	25	10	HE
Training Image	26	10	HE
Training Image	27	10	SA
Training Image	28	10	SU
Training Image	29	10	AN
Training Image	30	10	DI
Training Image	31	10	FE
Training Image	32	10	HE
Training Image	33	10	NE
Training Image	34	10	SA
Training Image	35	10	SU
Training Image	36	10	AN
Training Image	37	10	DI
Training Image	38	10	FE
Training Image	39	10	HE
Training Image	40	10	NE
Training Image	41	10	SA

Figure 5.3: Matlab snapshot during the training phase

5.2 Experimental Results and System Performance

The network was simulated and trained on Matlab software and tools. It was implemented using 2.7 GHz PC with 4 GB of RAM, Windows 7 OS and Matlab 2013a software tools. Matlab is an

established technical computing language for scientists and engineers and also an educative platform for developing algorithms and highly used in programming and data functions.

The network was tested on a dataset of 70 images; 21 images for each facial expression: angry, happy, sad, neutral, surprise, disgust, and fear. Table 5.2 represents the total recognition rate of the designed system. The table shows the training and testing rate of each facial expressions used in the developed system. This table below represents the number of images that were accurately recognized by the network in the training and the testing phase. It also shows the percentage of images that were not recognized during the testing phase. The images were obtained from JAFFE public database of size 256*256 pixels. They were processed and then rescaled to 64*64 pixels using pattern averaging for the purpose of easy computing and faster processing during the learning phase of network.

The number of recognized images was divided by the total number of images with respect to each case set (happy, angry, etc...). The result of this fraction is called the recognition rate, which is the efficiency of the neural network in recognizing the facial expression of an image. The experimental results of the developed facial expression recognition system were as follows: 100% using the training image set (210 images, 21 for each facial expression). The overall recognition rate was eventually calculated and the result is approximately 93% correct recognition rate.

Table 5.2: Total recognition rate of the designed system

	Facial expressions	Recognition rate	
		Training set	Testing set
	Happy	20/20 100%	90%
	Sad	20/20 100%	94%
	Angry	20/20 100%	95%
	Neutral	20/20 100%	91%
	Surprise	20/20 100%	92%
	Disgust	20/20 100%	91%
	Fear	20/20 100%	93%
Total recognition rate	All expressions	140/140 100%	93%

5.3 Results and Discussion

In this thesis, an intelligent facial expression recognition system based image processing and neural network has been developed. The system aims is to classify the facial emotions images into 7 expressions: angry, happy, sad, neutral, surprise, disgust, and fear. The developed intelligent system consists of two main phases: image processing and neural networks. In the first phase, the images are preprocessed in order to be ready for the next phase which is to be fed into a neural network for classification. The preprocessing includes the enhancement of image quality and noise removal using media filters and image adjusting. Moreover, this image processing phase includes also the size reduction using pattern averaging in order to reduce size of images and extract the good features. The extracted features using pattern averaging were used as inputs for a neural classifier based on a backpropagation learning algorithm. After learning and getting good training recognition rate, the network was tested on different images and the recognition rate was promising (93%).

As discussed in table 5.2 the system was able to recognize most of the facial emotions of images different than those used in the training phase.

Figure5.3 show some correctly recognized testing images that undergo the different phases of the developed system till they get classified into one of the 7 facial expressions.

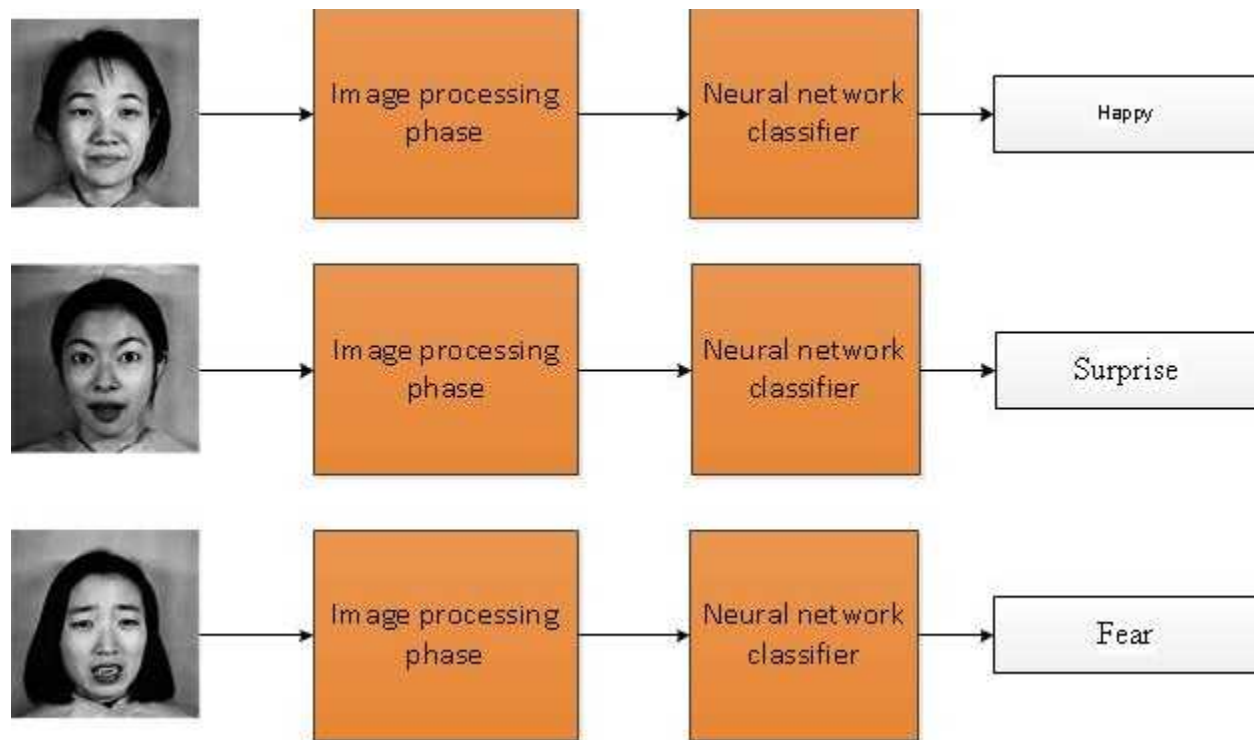


Figure 5.4: Correctly recognized images of the system

5.4 Results Comparison

Many researchers have been conducted in order to develop effective systems for the human facial expressions recognition based on image processing and intelligent classifier such as neural networks. For each research, the authors created image processing algorithms for the facial shapes detection and then they started the classification process that is based on intelligent systems like neural networks or other classifiers. In some papers, the authors used the whole processed facial expression image to be fed into the neural network, while others extracted the important features that may indicate the facial emotion of a person and they used as input vector for a neural network. In this thesis, the images were rescaled to 64×64 using pattern averaging to preserve the needed features and then they were fed into the network to be classified.

Niese et al.(2010)proposed a new method for the automatic recognition of facial expressions related to different basic emotions such as happy, sad etc...Their system aimed to recognize emotion based on 2d 3d facial feature extraction from color image sequences. It involved a series of image processing techniques, low level 3D computer vision and pattern recognition

techniques. Different features are then extracted such as eyebrow gradient, Mouth region of interest, mouth blob, and fiducial image points. The features were then fed into a neural network for classification purposes. The experimental result shows that their framework achieves robust and superior classification results, also across a variety of head poses with resulting perspective foreshortening and changing face size.

On the other hand, deep networks were also used by researchers in order to recognize human emotions from face images. Popovi et al. (2013) proposed an automatic emotion recognition system based on deep architectures neural network. Deep belief networks were used in this system to classify the facial emotions based on lip shapes of images obtained from JAFFE database. Deep belief network was trained first on the in a layer-wise greedy manner during the unsupervised training. However, during the second training phase, the network was trained supervisory using the images obtained from the dataset.

Table 5.4: Results comparison

Paper title	Authors	Methods and classifier used	Recognition rate
Deep Architectures for Automatic Emotion Recognition Based on Lip Shape	Branislav Popovi , Stevan Ostrogonac, Vlado Deli	Image processing and deep belief network (Same database)	92%
Emotion Recognition based on 2D-3D Facial Feature Extraction from Color Image Sequences	Robert Niese, Ayoub Al-Hamadi, Axel Panning and Bernd Michaelis	Features extraction and backpropagation neural network (Same database)	90%
A MATLAB based Face Recognition System using Image Processing and Neural Networks	Jawad Nagi, Syed Khaleel Ahmed	Image processing and SOM (self-Organizing Map) (Different database)	79.43%
Proposed system	Siham besha, Kamil Dimililer.	Pattern averaging and backpropagation neural network	93%

CHAPTER SIX

CONCLUSION

6.1 Conclusion

In this thesis, an intelligent facial expression recognition system was developed. The system is based on image processing and backpropagation neural network. The images are first processed in which they are filtered using median filter which is an adaptive filter used to remove noise and preserve the useful facial expression features. Another image processing technique is then used in order to increase the intensity of pixels of image and contrast using image adjustment. Moreover, the images undergo a process called pattern averaging in which size of image is reduced to 64*64 while preserving the good and useful facial features. Pattern averaging technique is used for the purpose of decreasing the processing time and makes the neural computations easier since the network learns only the extracted features rather the whole images.

A dataset of 210 images representing 7 different facial expressions was used for training and testing the network. The 210 images are of 10 Japanese females obtained from the public database available on the internet JAFFE database. Each female has 21 images for seven different facial expressions, 3 images each. For both phases of network; training and testing, the images were divided as follows: 140 for training and 70 for testing. Both sets contain images of the 7 different facial expressions: happy, angry, sad, surprise, disgust, neutral, and fear.

Once the images were processed during the image processing phase, the extracted features were fed into the network to start the learning. The network was trained using the backpropagation learning algorithm due to its implementation simplicity and effectiveness for such applications.

Once the network converges and the mean square gets minimized, the learning phase stops and the results were revealed. The training recognition rate was 100% which means that the network was capable of learning all different facial expressions used in that phase.

The network was then tested using 70 images of 7 different facial expressions. The experimental results of the network during that phase shows that the developed facial expression back

propagation neural network was capable of recognizing most of images except some. The testing result was 93% which is a promising result; however it can be enhanced.

6.2 Future Recommendations

Furthermore future work to improve the relevance of this new system can be to increase the database; increase in database size can yield better results. Also, the limitation in the image resolution the database used in the thesis can be solved when using modern technologies in medical science that can produce optimum image resolution. Also, a test of this system on other neural network examples and algorithms such as Support vector machine, Competitive learning, and Radial basis function in order to compare the best results the produced optimum results is suggested. Lastly, using other different methods for extracting medical features such as K-nearest neighbor can prove as advancement to this model research area.

References

- Sallam, A. A., & Khafaga, A. M. (2002). Artificial neural network application to alleviate voltage instability problem. In *Power Engineering 2002 Large Engineering Systems Conference on, LESCOPE 02* (pp. 133-141). IEEE.
- Khashman, A., & Dimililer, K. (2007, September). Neural networks arbitration for optimum DCT image compression. In *EUROCON, 2007. The International Conference on &# 34; Computer as a Tool&# 34;* (pp. 151-156). IEEE.
- Khashman, A. (2000). Neural networks arbitration for automatic edge detection of DNA bands in low-contrast images. In *Electrotechnical Conference, 2000.MELECON 2000.10th Mediterranean* (Vol. 2, pp. 469-472). IEEE.
- Khashman, A. (1999). Automatic edge detection of DNA bands in autoradiograph images. In *Industrial Electronics, 1999.ISIE'99.Proceedings of the IEEE International Symposium on* (Vol. 3, pp. 1184-1188). IEEE.
- Abiyev, R. H. (2014). Facial Feature Extraction Techniques for Face Recognition. *Journal of Computer Science*, 10(12), 2360.
- Baxt, W. G. (1995). Application of artificial neural networks to clinical medicine. *The lancet*, 346(8983), 1135-1138.
- Kröse, B., & Smagt, P. (1993). An introduction to neural networks, University of Ámsterdam. *Faculty of Mathematics & Computer Science*.
- Izard, C. E. (1994). Innate and universal facial expressions: evidence from developmental and cross-cultural research.
- Fyfe, C. (2005). Artificial neural networks. In *Do Smart Adaptive Systems Exist?* (pp. 57-79). Springer Berlin Heidelberg.

- Gupta, C. (2007). *Implementation of Back Propagation Algorithm (of neural networks) in VHDL* (Doctoral dissertation).
- Dimililer, K. (2013). Backpropagation neural network implementation for medical image compression. *Journal of Applied Mathematics*, vol(pp. 57-79).
- Gonzalez, R. C., & Woods, R. E. (2008). *Digital image processing. Nueva Jersey*.
- Church, J. C., Chen, Y., & Rice, S. V. (2008, April). A spatial median filter for noise removal in digital images. In *IEEE SoutheastCon 2008* (pp. 618-623).IEEE.
- Khashman, A., & Dimililer, K. (2008).Image compression using neural networks and haar wavelet. *WSEAS Transactions on Signal Processing*,4(5), 330-339.
- Khashman, A. (2008). IBCIS: Intelligent blood cell identification system.*Progress in Natural Science*, 18(10), 1309-1314.
- Kiran Kumar, K., & Singh, V. P. (2009). *Parallelized Backpropagation Neural Network Algorithm using Distributed System* (Doctoral dissertation).
- Kim, Y. T. (1997). Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE transactions on Consumer Electronics*, 43(1), 1-8.
- Lee, H. C., Wu, C. Y., & Lin, T. M. (2013).Facial Expression Recognition Using Image Processing Techniques and Neural Networks. In *Advances in Intelligent Systems and Applications-Volume 2* (pp. 259-267). Springer Berlin Heidelberg.
- .Michael, N. (2005). Artificial intelligence: A guide to intelligent systems. 2nd editon. *Harlow, England: Addison Wesley*.
- Al-Milli, N. (2013). Backpropagation neural network for prediction of heart disease. *Journal of theoretical and applied information Technology*, 56(1), 131-135.
- Popovi , B., Ostrogonac, S., Deli , V., Janev, M., &Stankovi , I. (2013). Deep architectures for automatic emotion recognition based on lip shape. In *12th International Scientific*

- Professional Symposium INFOTEH-JAHORINA, Jahorina, Bosnia and Herzegovina* (pp. 939-943).
- Rojas, R. (1996). *Neural Networks-A Systematic Introduction* Springer-Verlag. *New York*.
- Valiviita, S. (1998, May). Neural network for zero-crossing detection of distorted line voltages in weak AC-systems. In *Instrumentation and Measurement Technology Conference, 1998. IMTC/98. Conference Proceedings. IEEE* (Vol. 1, pp. 280-285). IEEE.
- Wang, Y., Chen, Q., & Zhang, B. (1999). Image enhancement based on equal area dualistic sub-image histogram equalization method. *IEEE Transactions on Consumer Electronics*, 45(1), 68-75.
- Li, Y., & Ma, W. (2010, October). Applications of artificial neural networks in financial economics: a survey. In *Computational Intelligence and Design (ISCID), 2010 International Symposium on* (Vol. 1, pp. 211-214). IEEE.
- Popovi , B., Ostrogonac, S., Deli , V., Janev, M., & Stankovi , I. (2013). Deep architectures for automatic emotion recognition based on lip shape. In *12th International Scientific Professional Symposium INFOTEH-JAHORINA, Jahorina, Bosnia and Herzegovina* (pp. 939-943).

APPENDICES

APPENDIX 1

SOURCE CODE

```
##### Normal images processing
clearall
closeall
clc
N=93;
AN=[];
pattern=[];
test=[];
target=[];
%saveData=zeros(4096,93);
##### Load the image,
cdsiham
%angry
for k = 1:10
I =imread(strcat(['AN',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
I=medfilt2(I,[4 4]);
g2=imadjust(I);
S5=g2;
%S5 = edge(g2, 'canny', 0.15);
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
AN=[AN vec];
end
for k = 1:10
I =imread(strcat(['AN',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
AN=[AN vec];
end
for k = 1:10
I =imread(strcat(['AN',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
AN=[AN vec];
end

%DI
DI=[];
for k = 1:10
```



```

I =imread(strcat(['DI',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
DI=[DI vec];
end
for k = 1:10
I =imread(strcat(['DI',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
DI=[DI vec];
end
for k = 1:10
I =imread(strcat(['DI',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
DI=[DI vec];
end

%FE
FE=[];
for k = 1:10
I =imread(strcat(['FE',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
FE=[FE vec];
end
for k = 1:10
I =imread(strcat(['FE',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
FE=[FE vec];
end

```

```

for k = 1:10
I =imread(strcat(['FE',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
FE=[FE vec];
end

%HA
HA=[];
for k = 1:10
I =imread(strcat(['HA',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
HA=[HA vec];
end
for k = 1:10
I =imread(strcat(['HA',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
HA=[HA vec];
end
for k = 1:10
I =imread(strcat(['HA',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
HA=[HA vec];
end

%NE
NE=[];
for k = 1:10
I =imread(strcat(['NE',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

```

```

I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
NE=[NE vec];
end
for k = 1:10
I =imread(strcat(['NE',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
NE=[NE vec];
end
for k = 1:10
I =imread(strcat(['NE',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
NE=[NE vec];
end

```

```

%SA
SA=[];
for k = 1:10
I =imread(strcat(['SA',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SA=[SA vec];
end
for k = 1:10
I =imread(strcat(['SA',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SA=[SA vec];
end

```

```

for k = 1:10
I =imread(strcat(['SA',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SA=[SA vec];
end

```

```

%SU
SU=[];
for k = 1:10
I =imread(strcat(['SU',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SU=[SU vec];
end

```

```

for k = 1:10
I =imread(strcat(['SU',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SU=[SU vec];
end

```

```

for k = 1:10
I =imread(strcat(['SU',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
I=medfilt2(I,[4 4]);
%S5 = edge(g2, 'canny', 0.15);
S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SU=[SU vec];
end

```

```

cd ..

```

```

for i=1:20

```

```

    pattern=[pattern AN(:,i) DI(:,i) FE(:,i) HA(:,i) NE(:,i) SA(:,i)
SU(:,i)];
target=[target eye(7)];
end

for i=21:30
    test=[test AN(:,i) DI(:,i) FE(:,i) HA(:,i) NE(:,i) SA(:,i) SU(:,i)];
end

pattern=double(pattern)./255;
test=double(test)./255;

net =
newff(minmax(pattern),[100,120,7],{'logsig','logsig','logsig'},'traingdx');

net = init(net);
net.LW{2,1} = net.LW{2,1};
net.b{2} = net.b{2};

% TRAINING THE NETWORK

net.trainParam.goal = 0.00001;
net.trainParam.show = 50;
net.trainParam.epochs = 50000;
net.trainParam.mc = 0.06;
net.trainParam.lr= 0.04;
net.trainParam.min_grad=1e-25;
[net,tr] = train(net,pattern,target);

train = sim(net,pattern);
test = sim(net,test);
disp('Training results');
train;
disp('Test results');
test;

for i=1:20
for k=1:7
    x=k+(i-1)*7;
if(train(k,x)>0.5)
switch (k)
case 1, msg=strcat(['training image ',num2str(x),' is AN']);
case 2, msg=strcat(['training image ',num2str(x),' is DI']);
case 3, msg=strcat(['training image ',num2str(x),' is FE']);
case 4, msg=strcat(['training image ',num2str(x),' is HE']);
case 5, msg=strcat(['training image ',num2str(x),' is NE']);
case 6, msg=strcat(['training image ',num2str(x),' is SA']);
case 7, msg=strcat(['training image ',num2str(x),' is SU']);
end

else
msg=strcat(['training image no: ',num2str(x),' is not recognized']);
end

```

```

disp(msg);
tic;
while(toc<0.05)
end
end
end

for i=1:10
for k=1:7
    x=k+(i-1)*7;
if(train(k,x)>0.5)
switch (k)
case 1, msg=strcat(['testing image ',num2str(x),' is AN']);
case 2, msg=strcat(['testing image ',num2str(x),' is DI']);
case 3, msg=strcat(['testing image ',num2str(x),' is FE']);
case 4, msg=strcat(['testing image ',num2str(x),' is HE']);
case 5, msg=strcat(['testing image ',num2str(x),' is NE']);
case 6, msg=strcat(['testing image ',num2str(x),' is SA']);
case 7, msg=strcat(['testing image ',num2str(x),' is SU']);
end
else
msg=strcat(['test image no: ',num2str(k),' is not recognized']);
end
disp(msg);
tic;
while(toc<0.1)
end
end
end

plot(tr.perf);
grid;
xlabel('Iterations');
ylabel('MSE');

disp(strcat('learning rate = ', num2str(tr.trainParam.lr)));
disp(strcat('momentum factor = ', num2str(tr.trainParam.mc)));
disp(strcat('training time = ', num2str(max(tr.time)), ' Seconds'));

```

APPENDIX 2

SOURCE CODE

```
##### Normal images processing
clearall
closeall
clc

AN =[];

##### Load the image,
cdsiham
%angry
for k = 1:10
I =imread(strcat(['AN',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);
S5=g2;

T = blkproc(S5, [4 4], @mean2);
vec=T(:);
AN=[AN vec];
end
for k = 1:10
I =imread(strcat(['AN',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
AN=[AN vec];
end
for k = 1:10
I =imread(strcat(['AN',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
AN=[AN vec];
end

%DI
DI=[];
for k = 1:10
I =imread(strcat(['DI',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
DI=[DI vec];
```

```

end
for k = 1:10
I =imread(strcat(['DI',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
DI=[DI vec];
end
for k = 1:10
I =imread(strcat(['DI',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
DI=[DI vec];
end

%FE
FE=[];
for k = 1:10
I =imread(strcat(['FE',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
FE=[FE vec];
end
for k = 1:10
I =imread(strcat(['FE',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
FE=[FE vec];
end
for k = 1:10
I =imread(strcat(['FE',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
FE=[FE vec];
end

```



```

%HA
HA=[];
for k = 1:10
I =imread(strcat(['HA',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
HA=[HA vec];
end
for k = 1:10
I =imread(strcat(['HA',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
HA=[HA vec];
end
for k = 1:10
I =imread(strcat(['HA',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
HA=[HA vec];
end

%NE
NE=[];
for k = 1:10
I =imread(strcat(['NE',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
NE=[NE vec];
end
for k = 1:10
I =imread(strcat(['NE',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);

```

```

NE=[NE vec];
end
for k = 1:10
I =imread(strcat(['NE',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
NE=[NE vec];
end

%SA
SA=[];
for k = 1:10
I =imread(strcat(['SA',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SA=[SA vec];
end
for k = 1:10
I =imread(strcat(['SA',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SA=[SA vec];
end
for k = 1:10
I =imread(strcat(['SA',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SA=[SA vec];
end

%SU
SU=[];
for k = 1:10
I =imread(strcat(['SU',num2str(k),' (1).tiff']));
I=imresize(I, [256 256], 'nearest');

```

```

g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SU=[SU vec];
end
for k = 1:10
I =imread(strcat(['SU',num2str(k),' (2).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SU=[SU vec];
end
for k = 1:10
I =imread(strcat(['SU',num2str(k),' (3).tiff']));
I=imresize(I, [256 256], 'nearest');
g2=imadjust(I);

S5=g2;
T = blkproc(S5, [4 4], @mean2);
vec=T(:);
SU=[SU vec];
end

cd ..

target=[eye(70) eye(70)];
pattern=[AN(:,1:10) DI(:,1:10) FE(:,1:10) HA(:,1:10) NE(:,1:10) SA(:,1:10)
SU(:,1:10)];
pattern=[pattern AN(:,11:20) DI(:,11:20) FE(:,11:20) HA(:,11:20) NE(:,11:20)
SA(:,11:20) SU(:,11:20)];
test=[AN(:,21:30) DI(:,21:30) FE(:,21:30) HA(:,21:30) NE(:,21:30)
SA(:,21:30) SU(:,21:30)];

pattern=double(pattern)./255;
test=double(test)./255;

net =
newff(minmax(pattern),[100,120,70],{'logsig','logsig','logsig'},'traingdx');

net = init(net);
net.LW{2,1} = net.LW{2,1};
net.b{2} = net.b{2};

```

```

% TRAINING THE NETWORK

net.trainParam.goal = 0.000001;
net.trainParam.show = 50;
net.trainParam.epochs = 50000;
net.trainParam.mc = 0.06;
net.trainParam.lr= 0.04;
net.trainParam.min_grad=1e-25;
[net,tr] = train(net,pattern,target);

train = sim(net,pattern);
test = sim(net,test);
disp('Training results');
train;
disp('Test results');
test;

for k=1:70
if(train(k,k)>0.3)
msg=strcat(['training image no: ',num2str(k),' is recognized']);
else
msg=strcat(['training image no: ',num2str(k),' is not recognized']);
end
disp(msg);
tic;
while(toc<0.2)
end
end

for k=71:140
if(train(k-70,k)>0.3)
msg=strcat(['training image no: ',num2str(k),' is recognized']);
else
msg=strcat(['training image no: ',num2str(k),' is not recognized']);
end
disp(msg);
tic;
while(toc<0.05)
end
end

for k=1:70
if(test(k,k)>0.3)
msg=strcat(['test image no: ',num2str(k),' is recognized']);
else
msg=strcat(['test image no: ',num2str(k),' is not recognized']);
end
disp(msg);
tic;
while(toc<0.1)
end
end

```

```
plot(tr.perf);  
grid;  
xlabel('Iterations');  
ylabel('MSE');  
  
disp(strcat('learning rate = ', num2str(tr.trainParam.lr)));  
disp(strcat('momentum factor = ', num2str(tr.trainParam.mc)));  
disp(strcat('training time = ', num2str(max(tr.time)), ' Seconds'));
```