FACE IMAGE RECOGNITION USING PCA, 2DCT AND NEURAL NETWORK

by

NGUELA MIRABEAU

Submitted in accordance with the requirements for
the degree of

MASTER OF TECHNOLOGY

in the subject

Electrical Engineering

at the

UNIVERSITY OF SOUTH AFRICA

SUPERVISOR: Professor Zenghui Wang

JULY 2018

THIS PAGE IS LEFT INTENTIONNALY BLANK

DECLARATION

Name NGUELA MIRABEAU

Student number: 53571274

Degree: MASTER OF TECHNOLOGY

TITLE OF THE DISSERTATION

FACE IMAGE RECOGNITION USING PCA, 2DCT AND NEURAL NETWORK

I declare that the above dissertation is my own work and that all the sources that I have used or quoted have been indicated and acknowledged by means of complete references.

SIGNATURE

01/08/2018
DATE

Research Summary

Automatic face recognition is a very important research area in computer science since it has been widely used in security systems. During the last two decade, the research area of face recognition has focused much attention from the scientific communities with the aim to provide highly intelligent human-machine interaction with high performance. This study proposes a system that encompasses a reduction of significant variable features using the principal components analysis on one hand, feature extraction using 2DCT on the other hand and then a combined method using both. Afterward each sample image is classified according to their pattern class using a Neural Network. The experimental results obtained shows an improvement in term of recognition rate when we combined the two methods.

KEY TERMS:

Principal component analysis, neural network, two-dimensional cosine transforms, eigenvalues and eigenvectors, covariance matrix, feed forward network, neural network.

ACKNOWLEDGMENT

I take this solemn opportunity to express my forever gratitude to my parents who have not spare their time and effort to make my life even better day by day. I want to express my appreciation to Unisa staff particularly my supervisor Professor Zenghui Wang who have guide me through and help me achieve in writing this dissertation Professor Yanxia Sun from the University of Johannesburg was a good mentor to me. Without forgetting my friends and family for their unconditional support, I give them a big thank you.

TABLE OF CONTENTS:

LIST OF ACRONYMS AND ABBREVIATIONS

FERET	The face Recognition Technology Evaluation
DARPA	(Defense advanced research product agency
A/D	(Analogue to digital converter)
AI	Artificial intelligence
DCT2 or 2DCT	Two-dimensional cosine transforms
DFT	Discrete Fourier transform
Fig.	Figure
PCA	Principal component analysis
ASMs.	Active shape model
JPEG	Meaning Joint photographic expert group
KLT	Meaning Karhunen-Loève transform
BMP	Windows bitmap
FRVT	The face recognition vendor tests
BioID	Meaning Biometric technology; it uses voice, face and lip movement to recognize a person
MAT	File) MATLAB file
ANN	Artificial neuron network
JPG, JPEG	Joint photographic experts group
FRGC	Face recognition grand challenge
Face database	A collection of face pictures.

CHAPTER 1- INTRODUCTION

Faces recognition has become one of the main focus research areas in computer sciences due to its application in security industries [23-33], [35-40]. It has drawn a lot of attention from the scientific communities with the aim to provide highly intelligent human-machine interaction with high performance. Our approach in this work is to propose methods that encompasses a reduction of significant variable features using the principal components analysis and classification method through neural network. The detected features can be subsequently compared with specific features or characteristics of the face of a person with those already known within a set of data base, thus for the actual face recognition or verification purpose.

1.1 HISTORY OF FACE RECOGNITION AND EVOLUTION

The history behind the development of facial recognition software started in the middle of the year 1960s, when the scientific community shows interest and start working on using computers to identify and recognizing human faces. The prototype sequence falls into three blocks as indicated in Fig. 1. The pioneers such as Woody Bledsoe, Helen Chan Wolf, and Charles Bisson were to

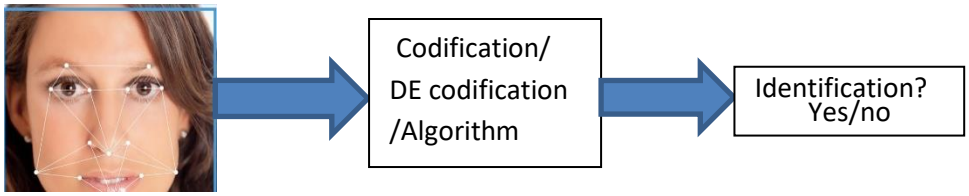


Fig. 1 location of essential feature using by Blesoe

develop a semi-automated facial recognition program. [1]

Woodrow Bledsoe uses a novel technique at that time called “man-machine facial recognition”, with other researches Helen Chan and Charles Bisson they started in California Panoramic Research, Inc. The algorithms use simple geometric models.

Their programs required the computer user to locate essential features such as the head, ears, nose, eyes, and mouth on the picture photograph (Fig.1). From those locations and a common reference point, inter-feature distances and ratios are calculated which was then compare to a set of known data. From it, Bledsoe designed and experiment a semi-automatic system [2].

During the year 1970´s Goldstein, Harmon, and Lesk represent a vector, containing 21 impressionistic features like hair color, ear form, lip size, eyebrow weight or nose thickness, as the basis elements to identified faces using pattern classification techniques to process automatically; however, the measurements and locations were to be manually computed, causing the program a great deal of computational time [3] .

However, in the year 1973 Kenade was the first researcher that developed a complete automatize system of face recognition. From there, Kenade designed and implemented a face recognition program. The software algorithm could extract sixteen facial parameters automatically. By comparing this automated extraction of parameters to a human or manual extraction, Kenade has showed that there is only a small difference between them. He achieved a result of 45 to 74% correct identification rate. Furthermore, Kenade has proven that even better results could be obtained when irrelevant features were not used. [4]

In the year 1988 to 1991, Kirby and Sirovich used a principal component analysis which is a standard linear algebra on face recognition on one hand and Turk and Pentland realize that in using the eigen faces technique, the residual error contained in the difference could be used to detect faces in images; a milestone in the real-time automated face recognition system. [5] The development didn´t stop there; the Defense Advanced Research Product Agency (DARPA) from the year 1993 to 1997 sponsored the face Recognition Technology Evaluation (FERET) and encourage the study and development of face recognition and technology by assessing the prototypes of faces recognition systems. From 2002 till 2006, the Face Recognition Vendor Tests(FRVT) evaluate the potential of commercially available system and measure technical progress. The conclusion is as follows:

- For a proper lighting condition, the face recognition technique is almost 91% verification at a small amount of 2% false accept rate
- By introducing the use of pliable models, in the aim to overcome lighting and pose variations we can use a 2D image onto a 3D transformation grid, which can greatly ameliorate the position of non-frontal face recognition.

In 2010, from the development of law enforcement through the forensic database to the beginning of social media boom, face recognition was used not only to crossed checking suspect and subsequently investigation and arrest by authorities. For example, the department of highway safety of motor vehicles (DHSMV) in Pinellas

county sheriff used face recognition technology. Social media such as facebook introduced in 2010 a facial recognition functionality in order to help identified people. The following year,2011, market another milestone of face image recognition in the implementation phase; Panama Government with the collaboration of the United states of America installed successfully a first major system in an airport, Panama's Tocumen airport. The purpose was to cut down on illicit activities since that airport was known as a passage for drug smugglers and organized crime.

High tech companies were not left out in the advancement of face recognition technology; Apple Inc. with its iPhone (x) introduced a security features based on face recognition with outstanding results: complete sold out at the opening of this model.

1.2 PROBLEM STATEMENT AND RELEVANCE

Simply using a search engine website like ''Google'' the term 'face recognition' yields more than 10.000 results. This implied that there are some huge potential applications using face recognition and it remain an ever-growing area unexploited.

The recognition of faces within a set of faces and verification methods used in security or access control applications involves frequently a pre-processing step of facial features which constitute a pivotal step in recognizing human faces. Face recognition remained therefore a up to date subject in image processing; pattern recognition, neural networks to name a few. limited to. As far computer vision research is concern facial recognition is not the only problem restricted to it; we could count the following examples in other application domains: human-machine interaction, video surveillance, security or law enforcement, photo cameras. (Table 1.1) It is important to think how to combine some methods to achieve better performance.

Table 1.1: Application of faces recognition

DISCIPLINE	APPLICATION DOMAIN
Industrial and Personal Security	Video Surveillance for home and industries Driver monitoring system and expression interpretation [6]
Biometrics	Identification of person, Ids, Passports, driver licenses, population census and

	registration, secure driving license, national border security[1] [7]
Inflow, outflow management	Access log or audit trails, Access verification, permission-based systems
Amusement and leisure	Photo Camera, Video game system [8]
Information Security	Access control security; User authentication, (trading, on line banking etc.); Data privacy: medical records

1.3 RESEARCH OBJECTIVES

Vast number of researchers has dedicated tremendous time and effort to developed of an automated face recognition system as stated in the history of face recognition; however, it is still difficult to achieve an acceptable result rate. Carry on from their achievement, the proposed research will concentrate on face recognition system for security purpose that will reach a higher result rate. This research tries to combine some methods include PCA, 2DCT and NN to find how to improve the facial image recognition performance.

1.4 LIMITATION

Face recognition remained up to date a valuable applications of computer image analysis. Building an automated system that equates human performance to recognize faces is not an easy task. Nonetheless there are different methods of identification (IRIS scans or fingerprints) may be more reliable but face recognition remains a major research area due to its non-invasive nature and is the primary method of identifying man [9]. Although other methods of identification (fingerprints or iris scanners) may be more accurate, facial recognition remains an important research topic because of its non-invasive nature and is the primary method of man to identifying the subject. Even though humans are skilled in the area of identifying known faces, it remained a challenge for the researchers when we have a huge amount of unknown faces. However, the machine like a computer can overcome human's limitations thanks to its limitless memory and high computational speed. The research will be carried out using a limited number of data that could eventually be

[1] Biometric registrations include persons of interest to partner countries, the United States or the international law enforcement community, as well as persons associated with or suspected of having committed terrorist activities, flagrant offenses or similar offenses, transnational criminal activities.

extended to a larger amount of data and hopefully produced the same or even a better result using PCA, 2DCT and NN.

1.5 THE LATEST DEVELOPMENT ON FACIAL RECOGNITION APPLICATION SINCE 2016

1.5.1 VIEW FROM THE GUARDIAN NEWS PAPER

Fig.1.1 US passport logo

- **March 2017**

The House committee has heard that facial recognition database that the FBI use in incontrollable
Published:27 Mar 2017.263
Facial recognition database used by FBI is incontrollable, related to passport control (Fig. 1.1), House committee hears[2]

January 2017

KFC China is using facial recognition technique to supply service to customers (Fig. 1.2)

Fig. 1.2 Customer face recognition

Fig. 1.3 Face selection

- **June 2016**

FBI is using vast public photo data and unsure facial recognition tech to identify criminals

Published:15 Jun 2016.
FBI using vast public photo data (Fig. 1.3) and doubtful facial recognition tech in order to identified criminals[3]

[2] "Facial recognition technology is a useful tool, people can be protected, their belongings, borders and a nation," said Jason Chaffetz, chairman of the committee. He added that it could be used in the private sector to protect financial transactions. and prevent identity theft.
[3]a GAO study finds that the FBI did not adequately disclose the effects of 411-m privacy-preserving photos and did not provide information on how often the software returns false positives

- **August 2016**

Digital business Facial recognition but one question:''a powerful ad tool or privacy nightmare?'' It appears as science fiction but tech that identifies VIP shoppers in a crowd (Fig. 1.4) could also be used for customer loyalty schemes
Published: 17 Aug 2016

.Fig. 1.4 Crowd picture

1.6. THE ISSUE OF FACE RECOGNITION

Faces recognition throughout the time has presented itself with some challenging issue that still existed today. Then, depending on the system requirements and methods, there are important factors to be taken into consideration such as: Illumination, expression, occlusion, pose variation, adaptability (to variable input image formats, color, length,) and portability. Apart some problem related to the environmental condition, there are also hardware constraints. In general, these problems and approaches are described in the next paragraph.

1.7 FACE RECOGNITION SCENARIOS

Basically, there are two mains approaches in face recognition with some sub-scenario.

1.7.1 FIRST APPROACH

Using the diagram in Figure 1.5 (Fig. 1.5) From one face we choose the template matching, not the geometric, then after processing it using neural network via a preselected feature, we projected it in the new subspace where we find the match of the face.

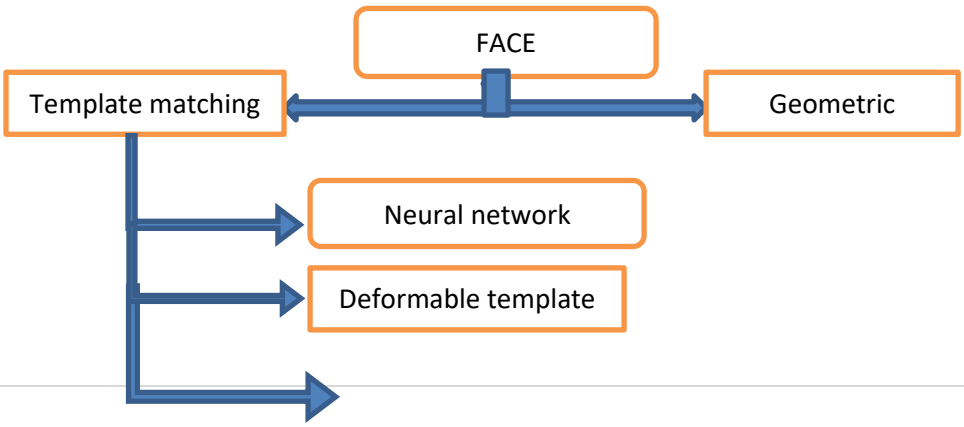


Fig. 1.5 Recognition Scenario

1.7.2 SECOND APPROACH

Some new approached have been listed in Table 1.2.

Table 1.2 (Wen-Yi Zhao: The advances in Face Processing –Face Recognition. ICIP 2003

APPROACH	Methods Representation
HOLISTIC METHODS	
Principal Components Analysis	Direct application of PCA
Eigen face	FLD using eigenspace
Fisherface/Subspace LDA	Two-class problem based on SVM
SVM,ICA	ICA-based feature analysis
Other representations	
LDA/FLA	FLD/LDA on raw images
PDBNN	Probalistic decision based Neural Network
HMM	Hidden Markov Models
FEATURE BASED METHODS Pure geometry methods Dynamic Link Architecture Convolution Neural Network	Graph matching methods, SOM (self-organizing map) learning based CNN (convolution neural network) methods [10]
HYBRID METHODS Modular eigenface Hybrid LFA Component-based	Eigenface and eigen modules Local and Global feature method Face region and components

1.8 SOME OF THE HINDRANCE OR PROBLEM IN FACES RECOGNITION

a) POSE

Pose variation is a problem in face recognition; the clear majority of study methods is center around frontal images. Whether it is face image representation, dimension

reduction algorithms, reconstruction technique to name a few, the testing phase uses frontal face technique. They are other constraint associated with face recognition application: we can cite for example video surveillance or security system where the input data is taken from uncontrolled environment.

b) ILLUMINATION

Illumination remained a major issue in automated face recognition system. The extraction of features elements on faces rely on intensity value between pixel to acquire much needed data to be processed, they are all have reliance on light variability; not only light source may vary but also its intensity. For a human, the color perception on a given object relied greatly on two factors: the nature of the surface and the light that shine upon it. Thus, the chromacity is a crucial factor in the face recognition. Zhao and al. have highlighted this problem by plotting the

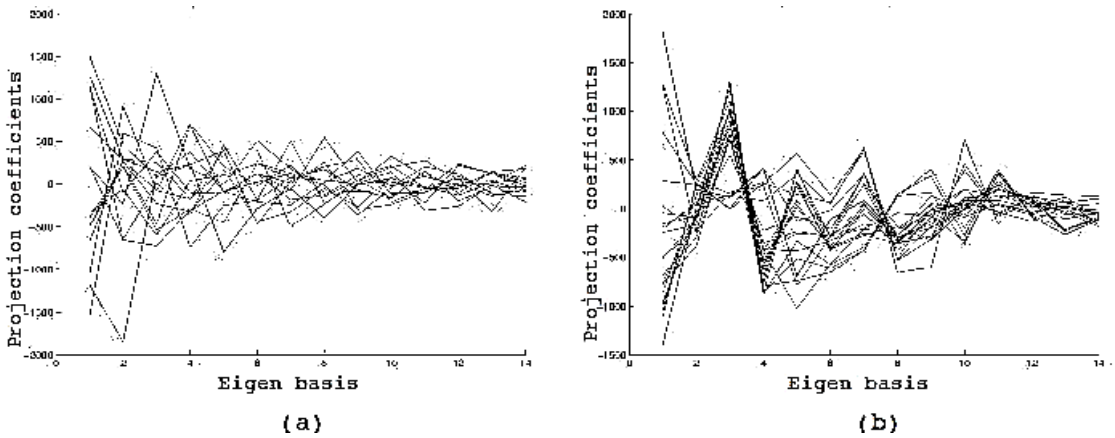


Fig. 1.6a and Fig. 1.6b : Variability due to class and illumination difference [10]

The figure above highlights the variation of Eigen space against the projection coefficient vectors because of the differences in class together with change due in illumination of the identical class.

c) OCCLUSION

Taking example of a face photograph from a security or surveillance camera where an object partially obstructs the full view of the face, therefore some parts of the face are hidden in the face recognition

The face recognition process relies strongly on the obtainability of a full and clear input face and not partial. Therefore, when some parts of the face are missing or

poorly visible and obstructed by object such as, hats, glasses beards, certain haircuts, etc, this may lead to a misclassification or bad classification. More often, most of the face recognition processes required a preprocessing phase that deal with this problem.

d) EXPRESSION

Among concerns in face image processing, there is face expression. However, facial expression is a minor problem; nonetheless a problem less acute as illumination or pose variation. The amount of expression variability due to illumination problems and pose are the impediment of obtaining accurate result for face recognition.

e) AGEING

Even though the latest development on the effect of aging in face recognition performance has been significant in the implementation phase using the state-of-the-art commercial off the shell (COTS) for face recognition systems, (*2017 IEEE Conference on computer Vision and pattern Recognition Workshop (CVPRW*), the facts remains that beyond very constrain life span periods (about 8.5 years) even with such a system, there is a considerable drop and loss in recognition rate . Ageing can be the object of a single research study on its own.

f) SPOOFING

Spoofing is defined as action of disguising a communication from an unknown source as being from a known, trusted source; applying it to face recognition is the act of mimic someone face using fraudulent means in order to gain access to emails, phone calls, and websites, spread malware through infected links or attachments, bypass network access controls, or redistribute traffic to conduct a denial-of-service attack. Even though spoofing can constitute a hindrance for accuracy in face recognition, it won't be part of the study of this report.

1.8.1 FACE IMAGE REPRESENTATION

Fundamentally, the software Matlab is a series of well-ordered written code used to command data stored in a working space environment. All these codes can be prolonged by others written commands inputted by a user. the backup of the files is

ensured in a file with the extension name (. Mat). Different lists are stored in a precise working environment. At the call of the instructions written on the list of commands [11] in the workspace, the process work well by processing the data.

At the end, the results are displayed under various forms such as: graphic form, surface shape, and image form

Matlab software gives us a robust tool to write, implement and test the unorthodox resolution, we will avoid throughout this thesis as much as possible the implementation of the matrix in favor of algorithmic expressions of the C11 form. Then we will use sequences of loops to develop the matrix expressions. She will help us help in understanding and developing techniques without having to resort to software libraries of matrix manipulation. When we reach the practical phase, the implementations in this thesis serve to understand the technical performance and veracity. Matlab is a processing images tool, so knowing what the image represents is a first start. Two spatial coordinates are used to define an image in space. By using a camera, the brightness is detected, and the points indicated have coordinates x and y. In general, the axis x and y represent respectively the various axes; both vertical and horizontal. [40]

Especially as we work on the scale of the image in gray, the values of the different points will be equivalent to the brightness of the point corresponding to the image seen by the lens of the camera. Then throughout this thesis, we will operate in an orthographic projection by ignoring the perspective where the coordinates of the x and y axes of an image and the coordinates of the real world are directly related. With reference to the coordinates of the x and y axes of the image, the brightness detected for example by a camera is transformed into a digital signal which is sent to the converter a / n (analog-digital converter) and recorded in the computer or other support. Finally, a matrix of points [41] composes an image of the different points that are elements of images or pixels. Consider for example, the set of pixel value of a given image (x) of $200 \mathcal{x} 250$(pixels value) in the Fig. 1.7 and Fig. 1.9

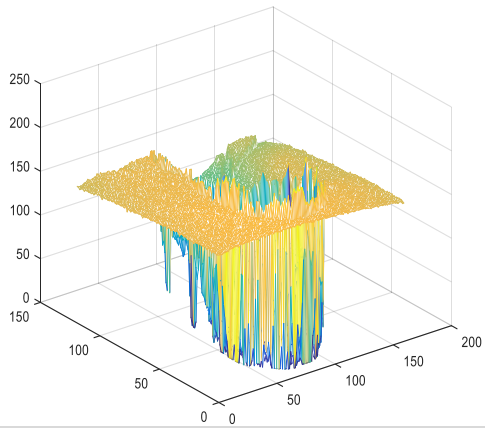


Fig. 1.7 Face image plus (relevant area)

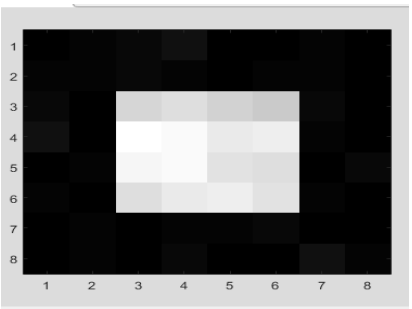


Fig.1.8 Image representation

Fig. 1.9 Surface Mesh plot of (Fig.1.7)

Fig.1.9.1 Another Image from Essex University data base

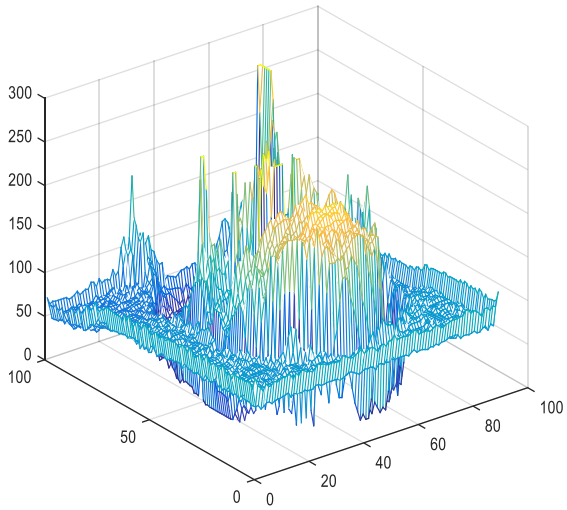


Fig.1.9.2 Surface Mesh plot of (Fig.1.9.1)

If the pixels have a larger value, then the square is brighter (here about 38 levels of brightness); the background is dark, and these pixels have a lower value (close to 0 brightness levels). These values were taken from the image of a bright square on a dark background (Figure 1.9). The square can be considered as a surface (or function) in Fig. 1.7 or an image in Fig. 1.8. It is particularly interesting to note that neither the background nor the square has a constant brightness because the noise has been added to the image. By removing the noise which is one of the advantages of using synthetic images, we can evaluate the performance of a computer vision technique on an image. We can see the difference with the image below in Fig.1.9.1 and the mesh representation in Fig.1.9.2.

1.9 IMAGE DATABASE

A database of face images (Fig. 1.9.3) is defined as a collection of data stored specially closed to the activities of one organization. They are organized to allowed easy retrieval access in case it is needed.

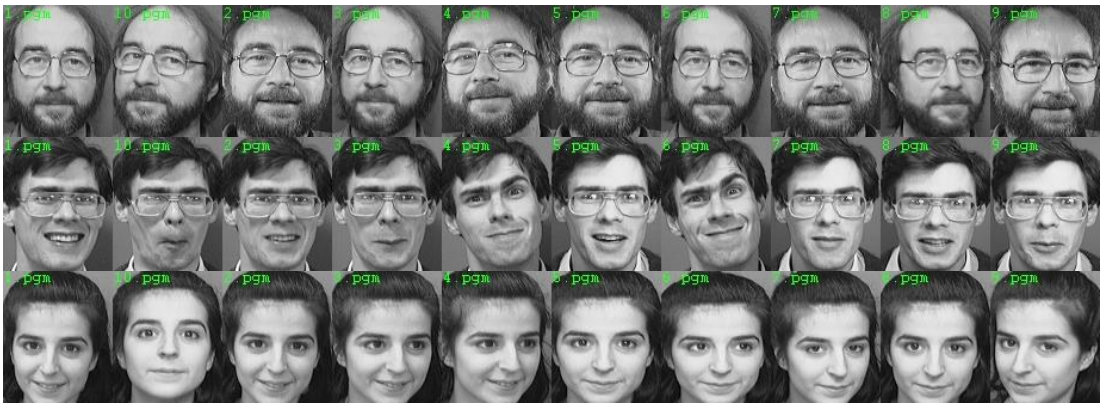


Fig. 1.9.3 Some images of the AT&T Data base [14]

Among a plethora of face image already store and ready to choose from, it has been a challenge to choose the one fitting for the current research. With great variance in term of size, qualities and availability, the data needed should have a degree of consistency throughout each image taken; that is the same physical Images environment for all picture taken. We have set our choice on two data bases for the following reasons: [12]

a. Manageability

The storage of data can be directly linked with metadata, so it becomes user friendly

b. Backup/Recovery

So it greatly simplifies the process by backing up the database which will back up every image. In an event of crash, solely one recovery procedure will be enough.

c. Stretchability

During the conversion of the image from one format to another, metadata can be extracted from it. More over the images are copied and resized, the quality also controls.

d. Flexibility

All data related to an image or set of images can logically exist together. Sets of images can be removed, updated or copied as easy as it is to write a query. Metadata can be easily attached to them and the images can be linked together.

1.9.1 FIRST DATA BASE

With a total amount of ten separate images of every of 40 distinct subjects, the first data base (Fig. 1.9.3) is used for the research purpose. The overall images of AT&T is m x n pixels and were taken against a dark homogeneous background (that make it avoid a preprocessing step) with the face image in an upright, frontal position (with tolerance for some side movement). The data base regroups ten separate images of each of 40 distinct subjects. [14]

Different times were necessary to take images, with the light variable, facial expressions face image used for study (open, nervous/ closed eyes, smiling / not smiling) and facial details (glasses, hair / no glasses, no hair). The face image is orderly arranged in 40 listings (one for each subject), which have names under the form sx,

1.9.2 SECOND DATA BASE

The second set is more interesting because the faces have different orientation position (Fig. 1.9.4); although it is color, we have changed it into grey during the preprocessing phase. The second set is from the University of Essex in the U.K.(United Kingdom) [13] . The Acquisition conditions is laid out as follow:

The subjects sit at fixed distance from the camera and are asked to speak, whilst a sequence of images is taken. The speech is used to introduce facial expression variation.

Database Description

Number of individuals: 153

Image resolution: 180 by 200 pixels (portrait format)

directories: female (20), male (113), male staff (20)

Contains images of male and female subjects in separate directories

Variation of individual's images

Backgrounds: the background is plain green

Head Scale: none

Head turn, tilt and slant: very minor variation in these attributes

Position of face in image: minor changes

Image lighting variation: none

Expression variation: considerable expression changes

Additional comment: there is no individual hairstyle variation as the images were taken in a single session.

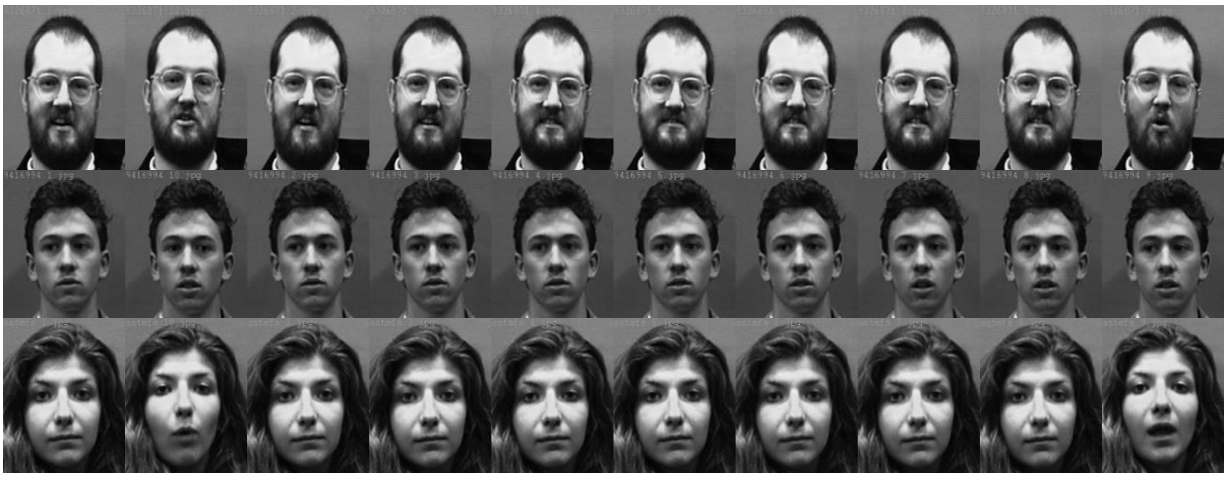


Fig. 1.9.4 Some images of University of Essex of the face data set [13]

CHAPTER 2- ARTIFICIAL NEURAL NETWORK -ANN

DEFINITION

An artificial neural network is a system processing information patterned on the operation of the human brain inspired from a biological neuron (Fig. 2)

The basic element that process the information is called neurons. These elements are interconnected altogether and work together to solve specifics problem.

2.1 BACKGROUND STUDIES

The artificial neural network has its history date far back in the 1940's [34] where Warren McCulloch and Walter Pitts created a threshold logic; which is a combination of algorithm and mathematic for computational model for neural network. This was neural network research's the land mark for where it was split into two distinct approaches. First of all, one method that focused on biological processes (Fig. 2) inside the brain and on the other hand focused on the applicability of neural networks to what is known as (AI) artificial intelligence.

2.2 THE BIOLOGICAL NEURON

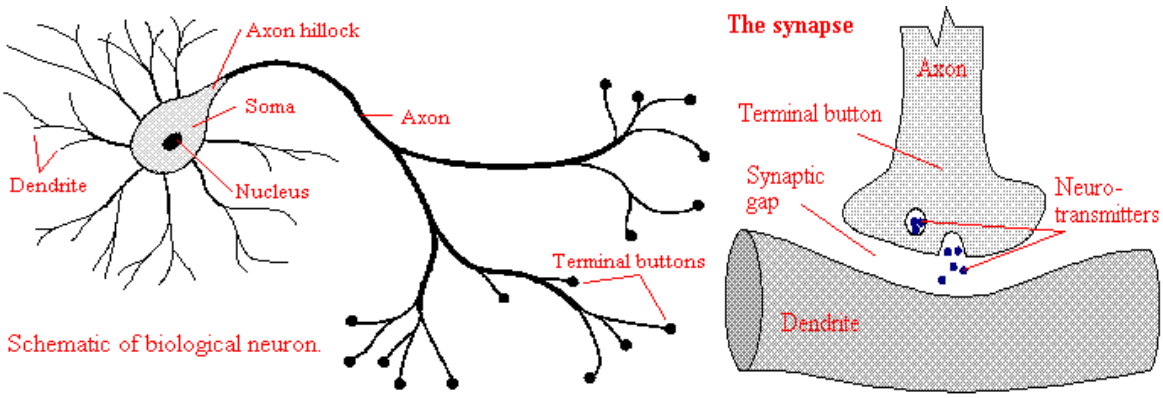


Fig. 2 Biological neuron [16]

About 100 billion neurons or interconnected nerve cells make up the average brain. Along the brain's pathways, chemical and electrical signals are transmitted by neurons forming cells. Thousands of neighboring neurons connect the dendritic tree. When one of these neurons is triggered, a positive or negative charge is received by one of the dendrites. The forces of all received charges are added using the time sum. The aggregated entry is then transmitted to the soma (cell body). In the treatment of incoming and outgoing data, the soma and the closed nucleus do not play an important role. They necessarily maintain the functioning of the neuron continuously. The axonal hill is the part of the soma that relates to the signal. Uniformity is critical in an analog device such as a brain where small errors can snowball and where error correction is more difficult than in a system. digital.

If the aggregated entry is greater than the threshold value of the axon of the hill, the neuron triggers and an output signal is transmitted in the axon. The force of exit is constant, whether the entrance is just above the threshold or a hundred times greater. The force of exit is not affected by the many divisions of the axon; he reached each terminal button with the same intensity he had at the top of the axon.

2.3 THE MATHEMATICAL APPROACH OF THE NEURON

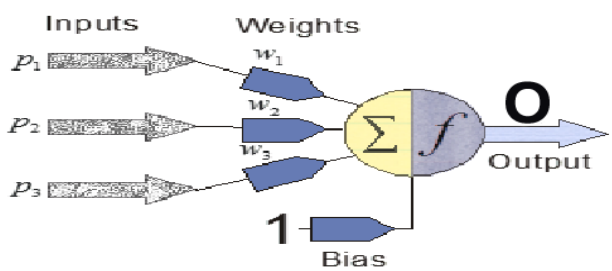


Fig. 2.1 Artificial neuron

$$\mathbf{O} = f(p_1 w_1 + p_2 w_2 + p_3 w_3 + b)$$
$$= f(\sum p_i w_i + b) \qquad (2.1)$$

with $p_1, p_2, and\ p_3$ representing all inputs data, they can be single value or set of vectors. The weights $w_1, w_2, w_3,$ are the strength between the node of the sum and the input or amplitude of the connection. The letter b is the adjustable bias value. The processing function is f and the letter a is the output of the overall result as in equation (2.1)

An artificial neuron (Fig. 2.1) within the network is a single unit that process the information; the output is produced from one to more inputs solely when the sum of the weights in the input plus the ones of the threshold makes this neuron a very flexible and strong one.

2.4 THE NEURAL NETWORK ARCHITECTURE

2.4.1 ONE LAYER OF NEURONS

The Network is a feedforward with one layer (Fig. 2.2) having a number of *R* input elements and a number of *a number of H* neurons follows.

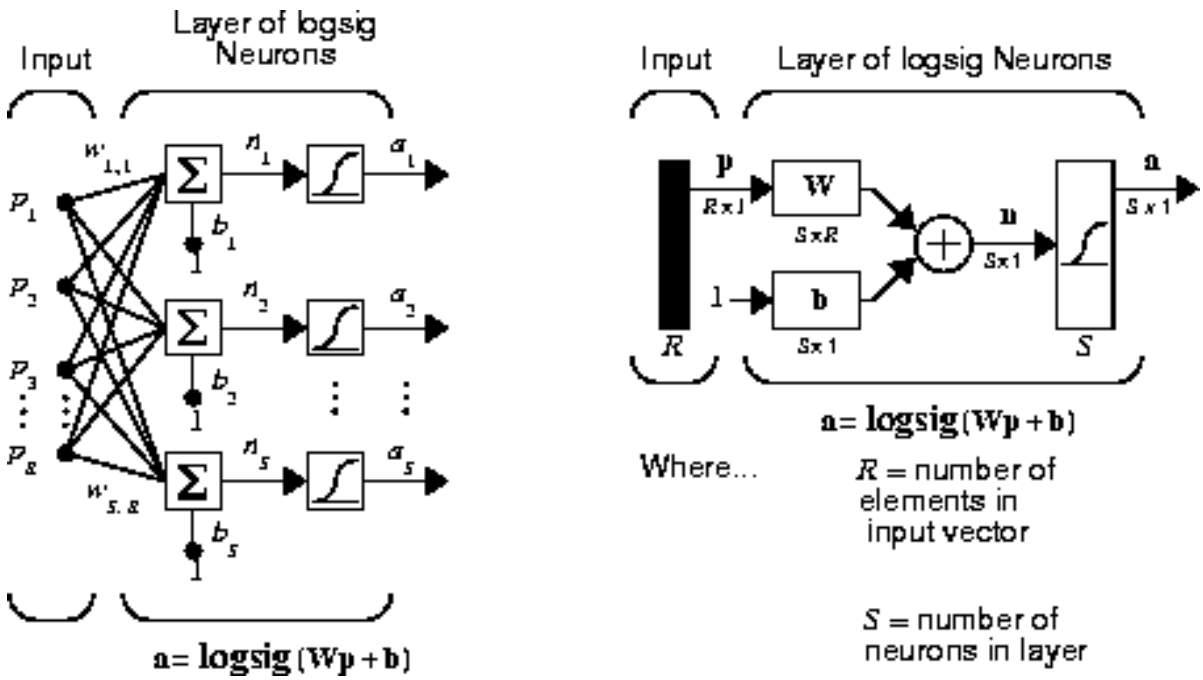


Fig. 2.2 One Layer network [17]

The network above with ith neuron has a summer that regroups its weighted inputs plus its bias to constitute its own scalar output n (i). In Fig. 2.2, each unit of the input vector p is connected to each neuron input through the weight matrix W. The different n (i) taken together form a net input vector of unit H N. At the end, the outputs of the neuronal layer are a vector of column A. The expression of a is shown above in the figure.

H and R' value are hardly the same. A layer is not forced to have the number of its inputs equivalent to the number of its neurons.

. The elements in the input vector that are fed in the network is in fact the principal component of the faces class calculated earlier.

$$W = (PCA; class) = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,R} \\ w_{1,2} & w_{2,2} & \dots & w_{2,R} \\ \vdots & \vdots & \dots & \vdots \\ w_{S,1} & w_{S,2} & \dots & w_{S,R} \end{bmatrix} \quad (2.2)$$

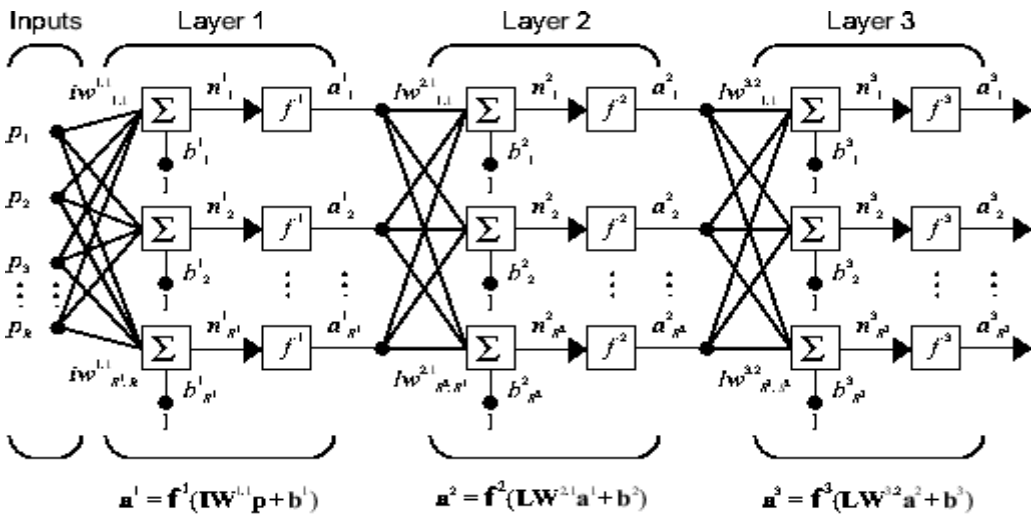
On the elements of the W matrix, the line indices indicate the target neuron of the weight, and the column indices indicate which source is the input for that weight. This means that, the indices in $w_{1,2}$ meaning that $w_{1,2}$ is the signal strength of the second input element to the first (and only) neuron as in the equation.2.2. We do not

summarize ourselves in this experience of this one-layer experience; It would be a multiple layer of neurons. (Fig. 2.3).

2.5 MULTIPLE LAYER OF NETWORK

A series of layers form the feedforward network. The different layers have a connection of the previous layer. The first layer has a connection from the network input followed by the last layer that is the network output.

The network in Figure 2.3 has multiple layers. A W-weight matrix, a b-polarization vector, and an output vector a define each layer. The number of the layer is highlighted by exposing to the variable of interest in order to differentiate the weights, the output vectors, and so on, for each of these layers in the figures. The notation of the layer in the three-layer network is then highlighted, and in the different equations, below of Fig. 2.3



$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1 + \mathbf{b}^2) \qquad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

..

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

Fig. 2.3 Multiple Layer of neurons [18]

2.6 HOW THE NEURAL NETWORK WORK?

2.6.1 LEARNING PROCESS

There are two major categories in which learning methods in neural networks can be classified:

- **Supervised learning** which method involves an external element: a teacher, so that each output classes is assigned its expected result for this experiment we are going to use the supervised learning because it includes error-correction learning.
- **Unsupervised learning** relying only on local information, and also called self-organization, non-supervised learning does not involve any external teachers. In the sense that it self-organizes from the data presented to the network and also detect emerging collective properties. Non-supervised learning paradigms are competitive learning and Hebbian.

The chosen Network is a feedforward neural network using gradient descent backpropagation algorithm. The weights are modifying according to the inputs value thank to the delta rule. The Fig. 2.4 represents a delta rule mechanism. It updates the weight according to the following mechanism:

The neural network uses the shift in the value of weights and biases during the training process to optimize network performance; and it is defined by the mean square error between the network outputs y and the target outputs t which is mse which will be defined below; also, the so-called default performance function for Feedforward networks is the mean quadratic error. The network performance function net.performFcn. The mse is defined as follows

$$F = mse = \frac{1}{N}\sum_{I=1}^{N}\left(t_j - y_j\right)^2 \tag{2.3}$$

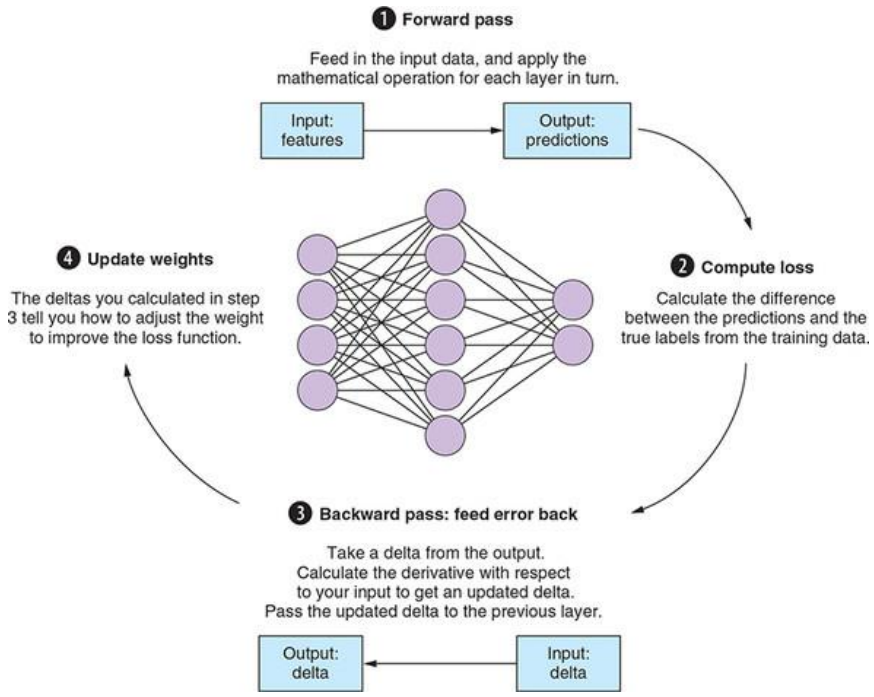


Fig. 2.4 Delta Rule [41]

Using the function names trainscg, the various weighting values are updated as well as the bias according to the gradient method conjugated by the network learning function : The training carry out is roll up according to trainscg parameters, shown in Table 2.1 with their default values:

Table 2.1 initialization parameters

net.trainParam.epochs	1000	Number maximum of epochs to train
net.trainParam.show	26	Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine	False	Generate command-line output
net.trainParam.showWindow	True	Show training GUI
net.trainParam.goal	0	Result goal
net.trainParam.time	Inf	Maximum time to train in seconds
net.trainParam.min_grad	1e-6	Minimum performance gradient
net.trainParam.max_fail	7	Maximum validation failures
net.trainParam.sigma	4.0e-5	Determine change in weight for second derivative approximation

net.trainParam.lambda	6.0e-7	parameter to regulate Hesse's indetermination

The function *''trainscg''* following the Table 2.1 above train the network if net input, its weight, and transfer functions possess derivative functions. Having the performance called perf, respectively aligned to bias and weight variable x, we can used backpropagation to calculate the derivative. The connection weights throughout the network is adjusted progressively, in a backward mode through the hidden layer and to the input layer till the appropriate output is found. In that way the network is taught how to produce the correct output for a given input. Exactly as in the functions 'traincgb', 'traincgf' and 'traincgp', the totality of the algorithm of the gradient conjugates to a scale is indeed based on the conjugated directions. When any of these conditions will occur, the training will stop; they are:

- The maximum number of epochs set out in the initial condition is achieve.
- Time chosen is exceeded by the maximum selected.
- Performance is lowered to the goal.
- The below mark min_grad is reached by the gradient's performance
- The max_fail times chosen for the max validation performance has exceeded whereas where validation is used, the previous time it has diminish.

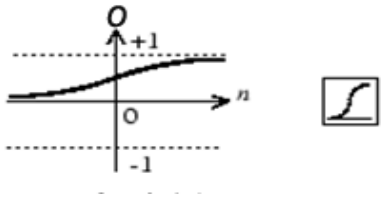
2.7 TRANSFER FUNCTIONS

The transfer function constitutes one of the pivotal elements within a neural network algorithm. Each designer can specify its own type of transfer function.

Among several types, there are two transfer functions which are commonly used and are highlighted in the following.

2.7.1 a) THE SIGMOID TRANSFER FUNCTION

The sigmoid transfer function (Fig. 2.5) shown below takes the input, where all the values can be between minus infinite and plus infinite and overwrites the output in the range 0 to 1.



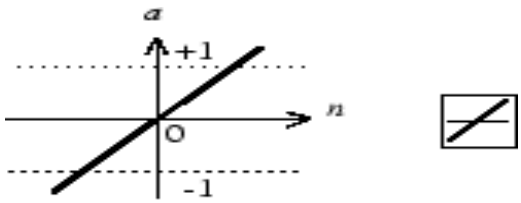
O= logsig(n)

Fig. 2.5 transfer function (log-sigmoid)

Thanks to its differentiable properties, the log-sigmoid transfer function is usually used in the hidden layers of multilayer networks.

2.7.1 b) THE PURELIN TRANSFER FUNCTION

The following figure (Fig. 2.6) illustrates the linear transfer function.



a=purelin(n)

Fig. 2.6 Transfer Function (Linear)

Linear transfer function is used in the final layer of multilayer networks that are used as function approximators. This is shown in multilayer neural networks and backpropagation training.

On the right side of the transfer function graph, the symbol in the square represented above is the associated transfer function. These small images replace the general function (*f)* when we draw diagram blocks (see Fig. 2.2) to mentioned the specific transfer function utilize in the network.

2.8 IS THERE ANY PARTICULAR METHOD TO CHOOSE THE AMOUNT OF HIDDEN NEURON AND INITIAL VALUE OF WEIGHT AND BIASES?

According to Matlab help book, given enough memory, and by using Levenberg-Marquardt training algorithm for small and medium size networks we can obtain optimum value for the network, or use a genetic algorithm to optimize the result of recognition with a correct amount of hidden neuron and biases value. However, in our case we require a large network; that is why we are using *trainscg*

2.9 SOLUTION OF LIMITATION USING TRAINSCG

Multilayer networks (Fig. 2.3) can perform typically almost any linear or nonlinear calculus of algorithm, and they can mimic any function quite well. Nevertheless, while the network being trained might theoretically be able to perform correctly, backpropagation that we are using, and its variations do not always find a solution. In facial recognition, there is complexity when compared to the error surface of a linear array. This is why the error surface of a nonlinear network like the one we will use for facial recognition is also complexes.

In multilayer networks, nonlinear transfer functions introduce many local minima into the error surface (Figure 2.7). This cause a problem. Depending on the initial start conditions, at the time of the gradient descent on the error surface, the network solution could be trapped in one of these local minima (Fig. 2.7). Being trapped in a local minimum may be good or bad depending on the proximity of the local minimum to the global minimum and the level of error required.

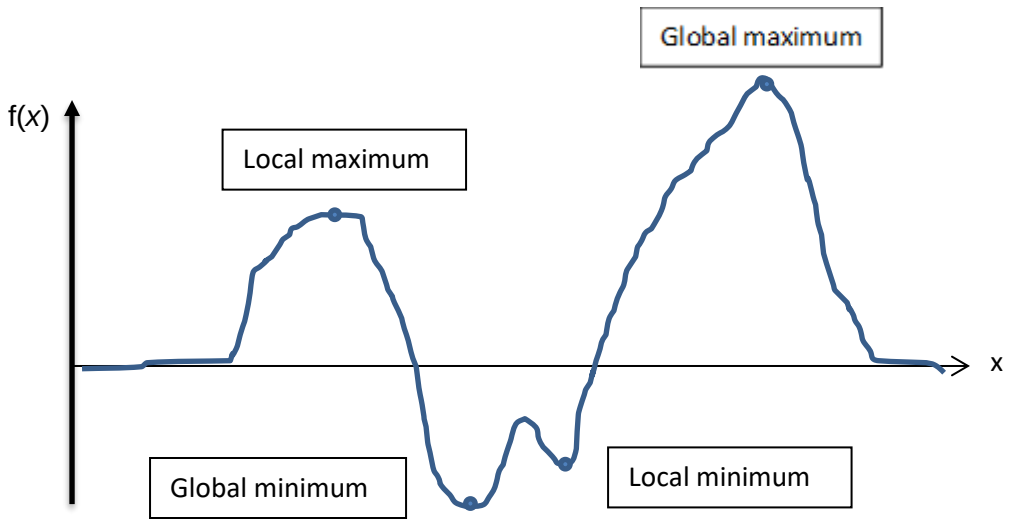


Fig. 2.7 Curb function with local minima and global maximum

However, the use back propagation does not always guarantee optimal weight search for the solution; with enough given neurons, any function can be implemented

with a multilayered repropagation network. While trying to solve this problem, we can reset the network and recycle it several times to ensure the best solution.

The number of neurons in their hidden layers poses a big problem of sensitivity on the networks. Two extremes emerge: on the one hand, if the number of neurons is important, this can lead to an excessive adjustment while the adjustment curve can seriously vary between the points or all the training points are well equipped if the number of neurons is low, this can lead to a under fit.

2.9.1 CHOICE OF THE TRAIN FUNCTION (trainscg)

In Scaled Conjugate Gradient each of the conjugate gradient algorithms that I have discussed so far requires a line search at each iteration. This line search is computationally expensive, since it requires that the network response to all training inputs be computed several times for each search. The scaled conjugate gradient algorithm (SCG), developed by Moller [15], was designed to avoid the time-consuming line search. This algorithm is too complex to explain in a few lines, but the basic idea is to combine the model-trust region approach (used in the Levenberg-Marquardt algorithm described later), with the conjugate gradient approach. See the final code for a detailed algorithm.

In the code (at the appendix of the report), we reinitialize our previous network and retrain it using the scaled conjugate gradient algorithm. The training parameters for *trainscg* are epochs, show, goal, time, min_grad, max_fail, sigma, lambda (indicated in the table2.1) . We have previously discussed the first six parameters. The parameter sigma determines the change in the weight for the second derivative approximation. The parameter lambda regulates the indefiniteness of the Hessian. The parameters show and epoch are set to 100.

The *trainscg* routine may require more iterations to converge than the other conjugate gradient algorithms, but the number of computations in each iteration is significantly reduced because no line search is performed that is my reason of choosing scale conjugation gradient backpropagation

CHAPTER 3- GENERAL METHODOLOGY

3.1 FLOWCHART DIAGRAM

The overall study follows the methodology lay out in the flowchart below on Fig.3 Each section will be explained; from 1 to 8

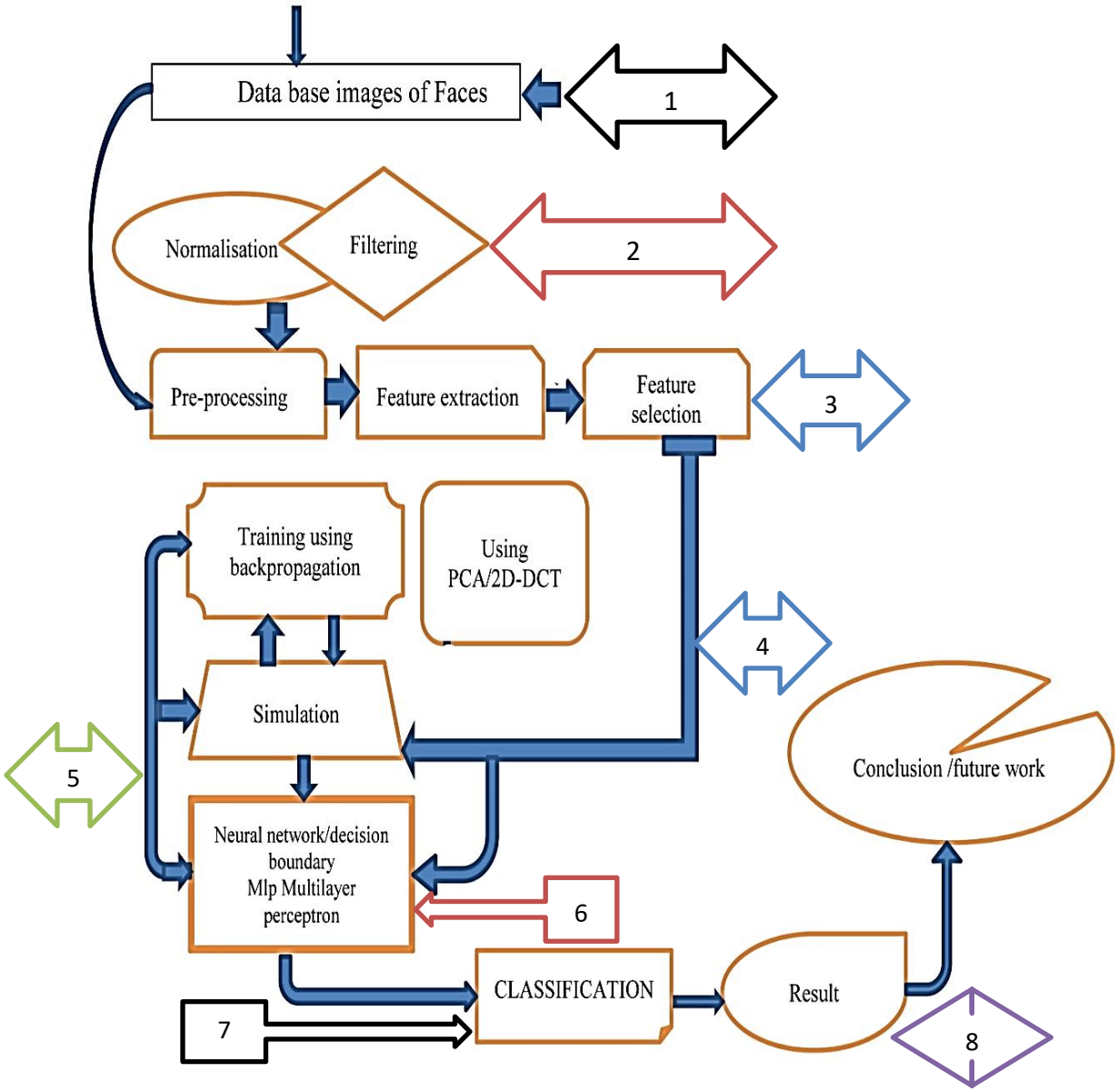


Fig. 3 Flowchart Diagram

3.1.1 Flowchart Description

a) 1 Data base image of faces are selected amount several existing and already available information on the web; with their authorization there are used for the sake of the recherche. Once selected, the different face image within the data base is then normalize.

b) 2 The normalization consists of readjusting the image value so that the new values are rescaling to some size variable.

c) 3 Using algorithm, we extract from each face image the feature that best describe each face in respect of all the face within the data base image.

d) 4 Then using the features selected, we simulate using a built neural network model to test the recognition rate

e) 5 After adjusting the relevant element within the network we classified all the face image and establish the result in a table.

3.2- FEATURE EXTRACTION

3.2.1 THE OVERVIEW

Usually, we have three type of features extraction implemented when we are dealing with image processing: low level feature extraction and high level feature extraction technique and the third one that we called point characteristics feature selection.

Finding shapes and object in computer image can be obtained using high-level feature extraction. The extraction of features components is one great step to achieve the recognition of human face. The extraction processes involve the extraction of specific part of the face such as: eyes, ears, and the nose or the mouth which constitute face features. We can search for them based on their form; for example, the white zone around the eyes has a form like ellipsoidal shape; the mouth can be sketched as two lines in shape, also are the top of the eyes, and the nose a vertical line crossing the horizontal. In other words, we can view them as objects and applied low-level functions to define the entire face. Feature extraction process have similarities in how we perceive the world around us.

Some basic shapes with a geometric shape such as triangle, square, or circle are represented in children's books. Any complex image structure can be broken down into simple shapes. Several applications use shape layout to perform analysis. If we take a facial image for example, the structure and position of the characteristic elements of a human face are such that: we obviously look for the mouth under the nose and the eyes are above (and on

each side) of the nose. In general, using feature extraction technique, we search for the invariability element so that the result of extraction does not change according to the rule set foremost (or specified). Among other things, no matter the value of any parameter that could change the appearance of shapes, the various techniques must obey to find robust and reliable forms. Non-variance is what we seek at the level of enlightenment; as a basic invariant, we look to find a shape no matter. In a general principle if there is a background and its form, then this form can be extracted. When we use a computer and the lighting conditions are bad, for example in absolute darkness not only can nothing be seen and also any recognition technique will fail. When the lighting is sufficient, the position remains the most important parameter, then wherever a shape appears it is towards this form that we are heading. In general, if we consider that the object or camera has an unknown orientation, this is called rotation or orientation invariance. This requires an invariance of size or scale. As nature always tends to want to roll the balls under our feet, we will look for properties of invariance in our form extraction techniques, taking into account that there are noises in our images. Indeed, since we are looking for shapes, note that there could be several in the image. If one is above the other, it will close or hide the other, so that the entire shape of an object will not be visible. The detection of a form means to know its existence in an image; which is easier than extraction; because extraction requires that we have a descriptive form of the image such as its size and position.

3.3 LOW LEVEL FEATURE EXTRACTION

One of the functions that can be automatically extracted from an image without any shape information (such as spatial relations) is the low-level operations.) Simply, a one-time operation such as threshold formation in any form of extraction can be considered as an operation. form of extraction of low-level features. Of course, these approaches can be used to extract high-level features where shapes are found in images. Even if we use the portraits of the creators, we can see that there are very

basic techniques and techniques as well as the most popular approaches. This is the first low-level feature we can find. This is the first low-level feature we can find. The first-order detectors are identical to the first-order differentiation, and of course the second-order edge recognition operators are identical in a slightly higher degree of differentiation. This is called detection and outline, which are intended to produce a line drawing, such as that of a face in Figure 3.1 (a) and (d), something that looks like a cartoonist's sketch [19]. All of this represents techniques for extracting localized features, in this case the curvature, and the most advanced approaches are to capture localized areas or areas of interest. Another form of edge detection is phase congruence. However, a wedge ID can be considered in which the points at which the lines are fixed edges are captured with a strong curvature, as is the case with respect to the lizard head in Fig. 3.1 (b) and (e). This is considered a lower-level feature that can be automatically extracted from the image at any time.

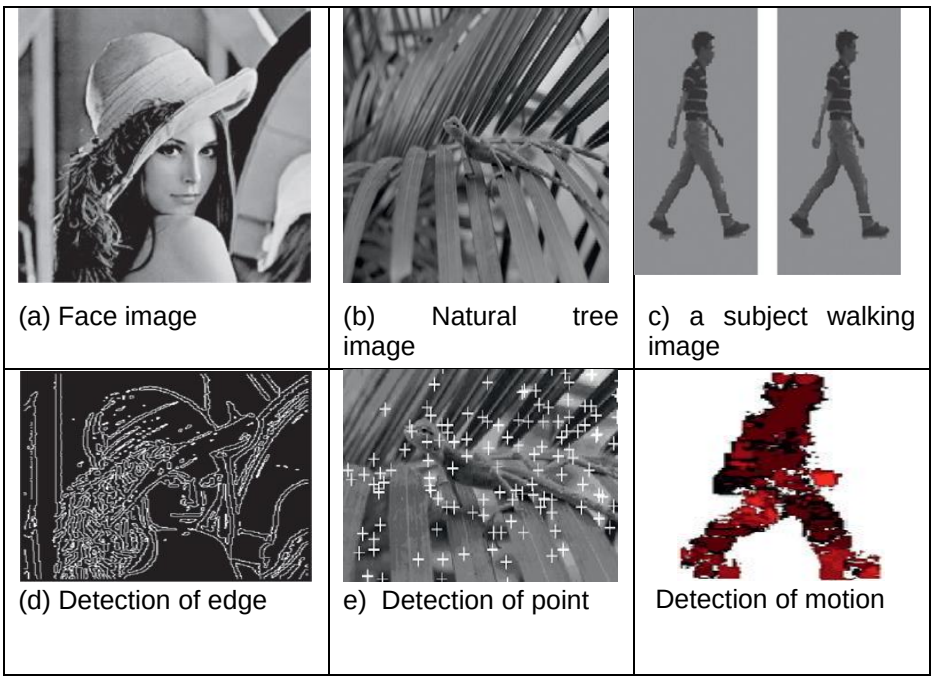


Fig. 3.1 Feature extraction technique

Importance of using different detection methods (including corresponding regions) by optical flow, by differentiation, closure problem, polishing stress and differential approach,Hornand Schunk method; correlation. These approaches are summarized in Table 3.3 below

Table 3.3 Feature extraction levels

Main Topic	Subtopics	Main Points

First-order edge detection	How can we detect an edge? By using first order differentiation and more sophisticated first order operators	Extraction difference, polishing, Sobel, and Prewitt, Canny. The frequency domain analysis are the basis of the operators
Second order edge detection	First-and second order operations differencing are linked; the ingredient of a second-order operator; inclusion of filtering and better operations	The use of Laplacian, zero-crossing detection; Laplacian of Gaussian, difference of Gaussian; are the main points of second-order differencing
Other edge operators	Others methods and result aspects; detect different operators	Alternative noise models *Spacek*; plus another edge models, *Petrou and Susan*
Phase congruency	To extract features, the use of inverse Fourier transform, edge and feature detection phase for alternative form.	Frequency domain analysis; The detection of a range of features such as wavelets photometric invariance.
Localized feature extraction [20]	The achievement in finding localized low-level features, could be done by stretching from curvature to patches; and the model of curvature and computation achieve: By edge detection, by change in intensity by correlation	For the planar curvature, and corners; we can use curvature estimation by: saliency operators modern feature detectors, change in edge direction, value change, or scale space.
Estimation of optical flow [21]	For the nature and movement of optical flow, he differential approach is used for the estimation of the optical flow,	Done using extraction by differencing; optical flow, aperture problem, constraint polishing, correlation; differential approach, the method of Hornand Schunk

3.4 HIGH LEVEL FEATURE EXTRACTION

3.4.1 DEFINITION

The search for forms has been correspondingly covered in the previous chapter and at the same time the extraction of lower-level features [27]. The target model, ie the characteristic, must be known a priori. In the case where the exact shape is unknown, or the perturbation of this form is impossible to parameterize, it is very difficult to model a distinct shape with good accuracy or to provide a lens model if necessary. The shape should be set in that it is flexible only in terms of parameters that define the shape or parameters that highlight the appearance of a pattern.

Then, we will perform operations that can tend to the desired result or adapt their results to different data. This leads to using formulations in flexible form. Several techniques in this case 4 is used to find flexible shapes in images. These are summarized in Table 3.4 and can be distinguished by the mapping function used to indicate the extent of correspondence between the image data and a shape. We get a deformable model, such that we can adapt the extracted image data if the shape is flexible or deformable.

Table 3.4 Feature extraction (high level technique)

Main Topic	**Subtopics**	**Main Points**
Pixels Operations	Using the level of a pixel, how can we detect features? Are there any frame and benefits	Thresholding. Differencing.

	of this method? There is necessity for type of shape.	
Template Matching	The inconvenient and advantages shape extraction by matching. It must be efficiently implemented.	Using Fourier implementation, and matching the Template. Occlusion and noise .
Low-level features	Collecting low-level features for object extraction. Frequency-based and parts-based methods. Detecting distributions of measures.	Wavelets and Haar wavelets. SIFT and SURF descriptions and Histogram of oriented gradients.
Hough transform	For conic sections feature extraction is applied by matching Hough transforms. Also for arbitrary shapes. Invariant formulations. The advantages reside in efficacity and speed of implementation.	Using Evidences gathering we extract features. we used Hough transforms for lines, circles, and ellipses Generalized and invariant

CHAPTER 4

FACE IMAGE RECOGNITION BASED PCA AND NEURAL NETWORK

Principal component analysis is defined as a statistical technique that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. PCA is based on factorization techniques contracted in linear algebra. From the previous chapter (chapter3), it is a high-level feature extraction technique.

PCA is a technique based on operation that t*akes* a set of data, analyses and transforms it into the new value with given statistical properties. All the data element are highlighted when the statistical properties are chosen in the transformation process. Thus, the new transformed data can be used for processing by defining important component of the value data. The sole aim to used principal component analysis method is to perform dimensionality reduction while conserving as much of the variance in the high-dimensional space as possible. For the work undertaken the principal component will serve as features.

4.1 CREATION OF FACE IMAGE DATA

On the section 1.8.1) above, we have highlighted a face image representation in matlab , Generally, face image data can be represented by a number of set of m vectors, for a set of m number of face image we have:

Let's called it

$$X=\{ x_1, x_2, x_3, \dots \dots x_m\} \tag{4.1}$$

where each vector x_i has n elements of features, i.e

$$x_1 = [x_{i,1}, x_{i,2} \dots \dots . x_{i,n}] \tag{4.2}$$

each vector x_i is the principal component value.

Now because we have a class of face where each contain several images, we can then put together those by taking of each vector element. That will be the feature column vector k for the set X(image) by:

$$C_{X,k} = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ x_{3,k} \\ \vdots \\ x_{m,k} \end{bmatrix} \qquad (4.3)$$

The sub index X nay seem unnecessary now. For *k* ranging from 1 to n; but, this will help us differentiate feature of the first set and of the reconditioned data. Then we put together all the feature in the feature matrix by considering each vector $C_{X,k}$ to be column in a matrix, i.e

$$C_{X,k} = \left[C_{X,1}\ C_{X,2}\ \ C_{X,3}\ \ \ldots . C_{X,k}\ \right] \qquad (4.4)$$

Using PCA, we will transform the vector $C_{X,k}$ into a set of new vectors with better classification capabilities. The maximum change in the measurements by the covariance is dealt with by the PCA and ensures that the significant data is accounted for.

4.2. COVARIANCE MATRIX- MATHEMATICAL DERIVATION

In general, we can determine whether there is a relationship between two datasets, so covariance measures the linear dependence between two random variables in order to compute the covariance

The covariance between the characteristics can be defined by taking the component of each single vector if the data of the face image is defined in the previous section; With only two components (which is not the case, it is made here for the purpose only understanding, face image has more than one dimension, that is, If $X_i = \{x_{i,1}, x_{i,2}\}$ then the result covariance is

$$\sigma_{X,1,2} = E[(c_{X,1} - \mu_{X,1})(c_{X,2} - \mu_{X,2})] \qquad (4.5)$$

E[ ] denotes the expectation which is loosely the average value of the elements of the vector ,moreover the multiplication is considered to be element by element and. The denomination $\mu_{X,k}$, *k* as a column vector deducted by multiplying the scalar value $E[c_{x,k,}]$ by a unitary vector. Which mean that, $\mu_{X,k}$ is a vector that has the mean value on each element. Thus, according to Equation (4.5), we first subtract the mean value for each feature and we continue by computing the mean of the multiplication of each element. Using the matrix form we expressed the covariance as:

$$\sigma_{X,1,2} = \frac{1}{m}\left(\left(c_{X,1} - \mu_{X,1}\right)^T \left(c_{X,2} - \mu_{X,2}\right)\right) \tag{4.6}$$

where T represents the matrix transpose. Quite often, features are represented as rows, so you can find the transpose positioned on the second factor rather than on the first. Sine the covariance is symmetric and therefore $\sigma_{X,1,2} = \sigma_{X,2,1}$

In addition to Equation (4.5) and (4.6), by developing the products in Equation (4.6) This is therefore written in simple form as

$$\sigma_{X,1,2} = \sigma_{X,2,1} = E[c_{x,1}, c_{x,2}] - E[c_{x,1}]E[c_{x,2}] \tag{4.7}$$

with $$E\left[c_{x,1}, c_{x,2}\right] = \frac{1}{m}\left(c_{X,1}^T c_{X,2}\right) \tag{4.8}$$

Equations (4.5), (4.6), and (4.8) are another manner to compute the covariance. They are obtained by representing products and averages in algebraic similitude definitions.

The covariance value is classified from zero (indicating no relationship) to extreme positive and negative values that pertained strong dependencies. The Cauchy Schwarz inequality is used to obtained the maximum and minimum value and they are expressed by

$$|\sigma_{X,1,2}| \leq \sigma_{X,1}\sigma_{X,2} \tag{4.9}$$

‘‘ **||**’’ denotes the absolute value

and $\sigma_{X,1}^2 = \sigma_{X,2,1} = E\left[c_{x,1}, c_{x,2}\right] - E\left[c_{x,1}\right]E\left[c_{x,2}\right]$ defines the variance of $c_{x,1}$. It is worth noted that the variance is a measure of dispersion; thus, this inequality

indicates that the covariance will be huge if the data has large ranges. When the sets are totally dependent, then $|\sigma_{X,1,2}| = \sigma_{X,1}\sigma_{X,2}$. The covariance measures also a linear relationship.

Our face image has more than one dimension. To define the covariance, we considered every pair of components. These components are represented in matrix called covariance matrix and defined as:

$$\sum_X = \begin{bmatrix} \sigma_{X,1,1} & \sigma_{X,1,2} & \cdots & \sigma_{X,1,n} \\ \sigma_{X,2,1} & \sigma_{X,2,2} & \cdots & \sigma_{X,2,n} \\ \vdots & \vdots & \cdots & \vdots \\ \sigma_{X,n,1} & \sigma_{X,n,2} & \cdots & \sigma_{X,n,n} \end{bmatrix} \tag{4.10}$$

According to Eq.4.5 the value in the element *(i,j)* in the covariance matrix is given

By $$\sigma_{X,i,j} = E[(c_{X,i} - \mu_{X,j})(c_{X,j} - \mu_{X,j})] \tag{4.11}$$

Taking into consideration the Eq. 4.6 the covariance matrix can be expressed as:

$$\sum_X = \frac{1}{m}\left((c_X - \mu_X)^T (c_X - \mu_X)\right) \tag{4.12}$$

Here, μ_X is the matrix that has columns $\mu_{X,i}$ .By observing the definition of the covariance given in the last section, we note that the covariance matrix is symmetric; the diagonal of the covariance matrix defines the variance of the feature and that given symmetry in the definition of the covariance.

4.3 FACE IMAGE TRANSFORMATION

The purpose is to find a combination of vector that could regroup every single vector defined in the set X into another feature vector for the set Y, so that there is diagonality in the element in Y. The linearity of the transformation is expressed as:

$$c_Y = c_X W^T \tag{4.13}$$

Or re-writing in matrix form, we have:

$$\begin{bmatrix} y_{1,1} & y_{1,2} & \dots & y_{1,n} \\ y_{2,1} & y_{2,2} & \dots & y_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ y_{m,1} & y_{m,2} & \cdots & y_{m,n} \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ x_{m,1} & x_{m,2} & \cdots & x_{m,n} \end{bmatrix} \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{m,1} & w_{m,2} & \cdots & w_{m,n} \end{bmatrix} \tag{4.14}$$

Remember that

$$c_Y^T = W c_X^T \tag{4.15}$$

That is Eq.4.14 can be write as:

$$\begin{bmatrix} y_{1,1} & y_{2,1} & \dots & y_{m,1} \\ y_{1,2} & y_{2,2} & \dots & y_{m,2} \\ \vdots & \vdots & \cdots & \vdots \\ y_{1,n} & y_{2,n} & \cdots & y_{m,n} \end{bmatrix} = \begin{bmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} \\ \vdots & \vdots & \cdots & \vdots \\ w_{n,1} & w_{n,2} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{2,1} & \dots & x_{m,1} \\ x_{2,1} & x_{2,2} & \dots & x_{m,2} \\ \vdots & \vdots & \cdots & \vdots \\ x_{1,n} & x_{2,n} & \cdots & x_{m,n} \end{bmatrix} \tag{4.16}$$

In order to obtain the covariance of the features in Y based on the features in X, we substitute $c_Y \; and \; c_Y^T$ and the definition of the covariance matrix as:

$$\sum_Y = \frac{1}{m}[(W c_X^T - E[W c_X^T])(\, c_X W^T - E[c_X W^T])] \tag{4.17}$$

Eq.4.17 by factorization gives:

$$\sum_Y = W \Sigma_X W^T \tag{4.18}$$

This is greatly simplified since the inverse of the transformation is equal to its transpose, i.e., Now to map Y into X, we use the inverse of the transformation.

$$W^{-1} = W^T \tag{4.19}$$

Taking the Eq. 4.19 and 4.20 we have:

$$\sum_X = W^{-1} \Sigma_Y (\, W^T)^{-1} \tag{4.21}$$

Since the covariance is symmetric,

$$\sum_X = \Sigma_X^T$$

$$W^{-1}\Sigma_Y(W^T)^{-1} = (W^{-1})^T\Sigma_Y((W^T)^{-1})^T \tag{4.22}$$

Eq.4.22 could be true only if the inverse of W is equivalent to its transpose

that is: $$W^T c_Y^T = c_X^T \tag{4.23}$$

4.3.1 EIGEN VALUE

By considering the eq. in 4.20 we can rewrite eq. 4.19 as:

$$\sum_X W^T = W^T \Sigma_Y \tag{4.24}$$

In matrix form, we can rewrite

$$W^T\Sigma_Y = \begin{bmatrix} w_{1,1} & w_{2,1} & \cdots & w_{n,1} \\ w_{1,2} & w_{2,2} & \cdots & w_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ w_{1,n} & w_{2,n} & \cdots & w_{n,n} \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

$$= \lambda_1 \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \lambda_2 \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \ldots\ldots\ldots\ldots\ldots + \lambda_n \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix} \tag{4.25}$$

The name dubbed as λ is the diagonal elements of the covariance; same for the other side we have:

$$\Sigma_X W^T = \Sigma_X \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \Sigma_X \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \ldots\ldots\ldots\ldots + \Sigma_X \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix}$$

$$= \lambda_1 \begin{bmatrix} w_{1,1} \\ w_{1,2} \\ \vdots \\ w_{1,n} \end{bmatrix} + \lambda_2 \begin{bmatrix} w_{2,1} \\ w_{2,2} \\ \vdots \\ w_{2,n} \end{bmatrix} + \ldots\ldots\ldots\ldots\ldots + \lambda_n \begin{bmatrix} w_{n,1} \\ w_{n,2} \\ \vdots \\ w_{n,n} \end{bmatrix} \tag{4.26}$$

From the equation above 4.26 and 4.25 we derive W by solving the

equation $$\Sigma_X W_i = \lambda_i w_i \quad (4.27)$$

Equation.4.27 can be rewritten as:

$$\Sigma_X W_i - \lambda_i w_i = 0 \quad (4.28)$$

4.3.2 FINDING THE PRINCIPAL COMPONENTS OF CLASS FACE IMAGES

In resume, we can summaries PCA based feature extraction by reducing the PCA approach into 6 steps in order to analyze a set a data:

1.-The whole dataset imply of d-dimensional samples ignoring the class labels is taken.

2.-We calculate the d -dimensional mean vector (i.e. the means for each dimension of the complete dataset)

3.-Then we Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set.

The scatter matrix is calculated by the following equation:

$$S = \sum_{k=1}^{n} (x_k - m)^T (x_k - m)$$

where

$$m = \frac{1}{n} \sum_{k=1}^{n} (x_k)$$

4.-Follow by the computation of eigenvectors $(e_1, e_2, e_3, \ldots . e_k)$ and corresponding eigenvalues $(\lambda_1; \lambda_{2\,;\ldots\ldots} \lambda_d)$

5.-Sort the eigenvectors by decreasing eigenvalues and select k eigenvectors with the largest eigenvalues to form a *d*×*k* dimensional matrix *W* (where every column represents an eigenvector)

6.-Use this *d* ×*k* eigenvector matrix to transform the samples onto the new subspace.

For the face image recognition, to find the principal components, we choose the features that have large values of λi. Knowing that λi measures the variance, and features that have large range of values will have large variance. Afterwards, we

lessen the dimensionality of the new feature vectors by tuning to zero components with low λi values.

4.4 PRACTICAL APPLICATION EXPERIMENT FOR PCA

4.4.1 FLOWCHART

The flowchart in Fig.4 detailed the working mechanism method of writing the code that will generate the result of the recognition rate. Each block indicated a subsystem that is put altogether in a sequential manner from the top till the bottom.

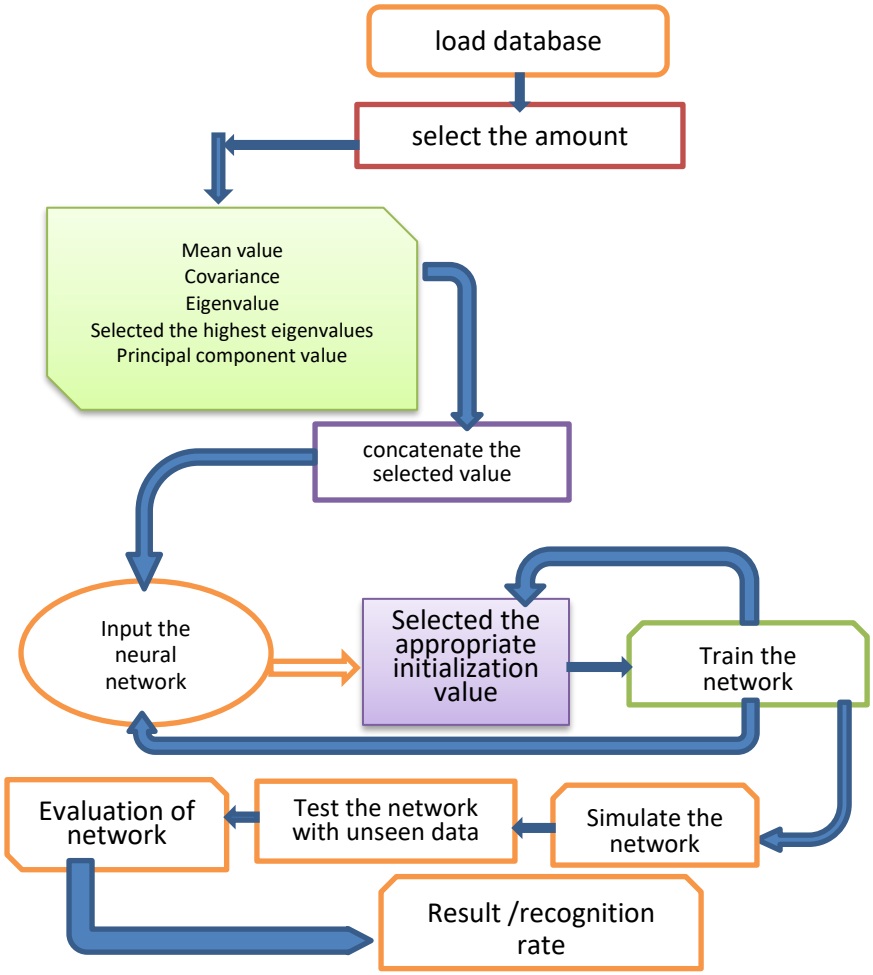


Fig. 4 Algorithm use to write the program

4.5 EXPERIMENTAL PHASE FOR FACE RECOGNITION

4.5.1 SEQUENTIAL WORKING PROCEDURE

Face recognition is classified as a pattern recognition problem. A set of s input vectors constituting the total feature vector are grouped as columns in a matrix.

Feedforward neural networks are also known as Multi-layered Network of Neurons (MLN) will be used. These networks of models are called feedforward because the signal data only travels forward in the neural network, from the input nodes through the hidden layers (single or many layers) and finally through the output nodes.

4.5.2 JUSTIFICATION OF THE VALIDATION , TRAINING SETS THE DATA.

According to a research work called:*"A scaling law for the validation-set training-set size ratio" by Isabelle Guyon AT&T Bell Laboratories, Berkeley, California*
It is determined that the ratio of the validation set size over the training set size scales like the square root of two complexity parameters: the complexity of the second level of inference (minimizing the validation error) over the complexity of the first level of inference (minimizing the error rate on the training set)

A set of S target vectors so that they indicate the classes to which the input vectors are assigned, here we have two sets; one of 40 classes for the first data and another of 100 classes for the second data. There are two approaches to creating the target vectors. The blocks 1,2,3, and 4 in the flowchart diagram Fig. 3 of chapter 3 are computed using the algorithm of principal component analysis method; the block 5 has a workflow as follow:

- Using the neural network pattern recognition tool GUI, nprtool(Fig. 4.1). This

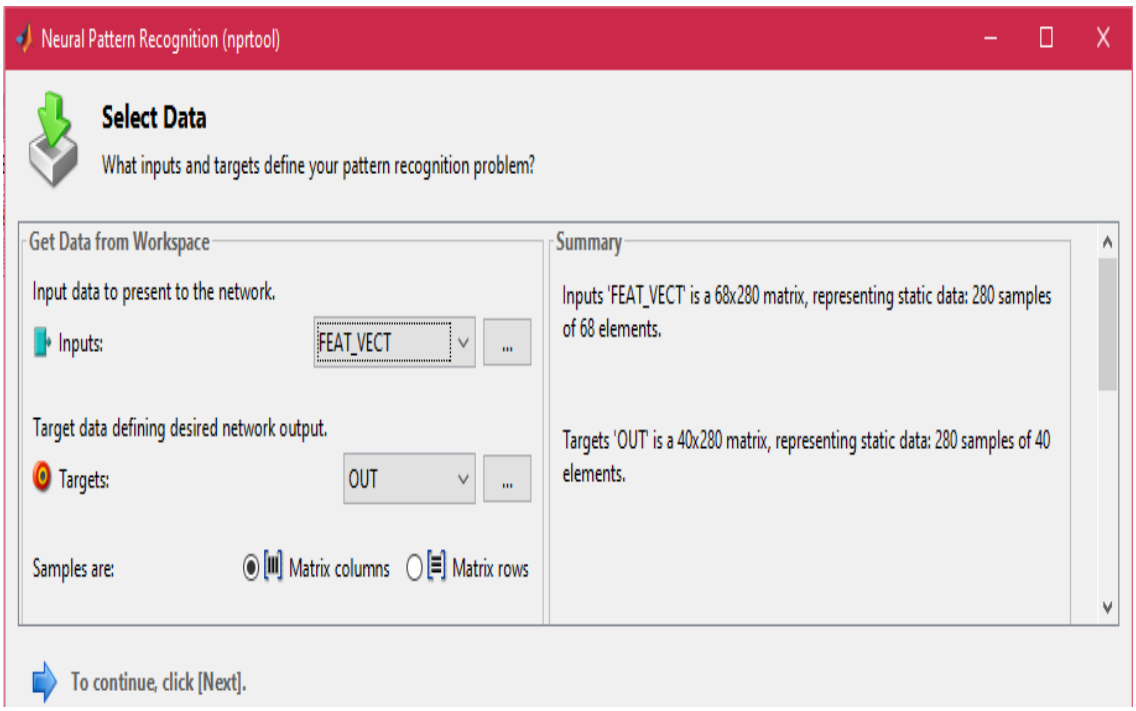


Fig. 4.1 Nprpool window input/target vectors

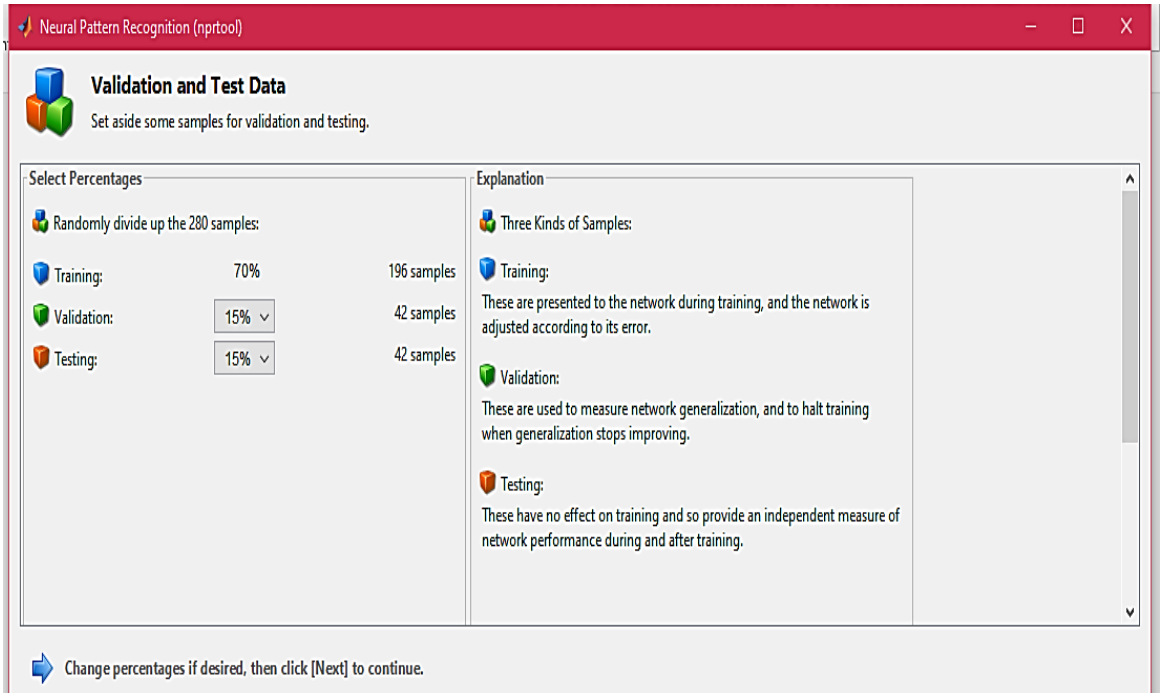


Fig. 4.2 Division selection of sample

first data set consists of 85 to 100 elements input vectors and 40-elements target vectors. There are more than 85 elements in each target vector, because there are (40) forty associated with each input vector.

-First, we use init to reset the initial network weights and biases to new values and train again until we obtained better result.

- Then the number of hidden neurons is Increased. (Fig. 4.3)

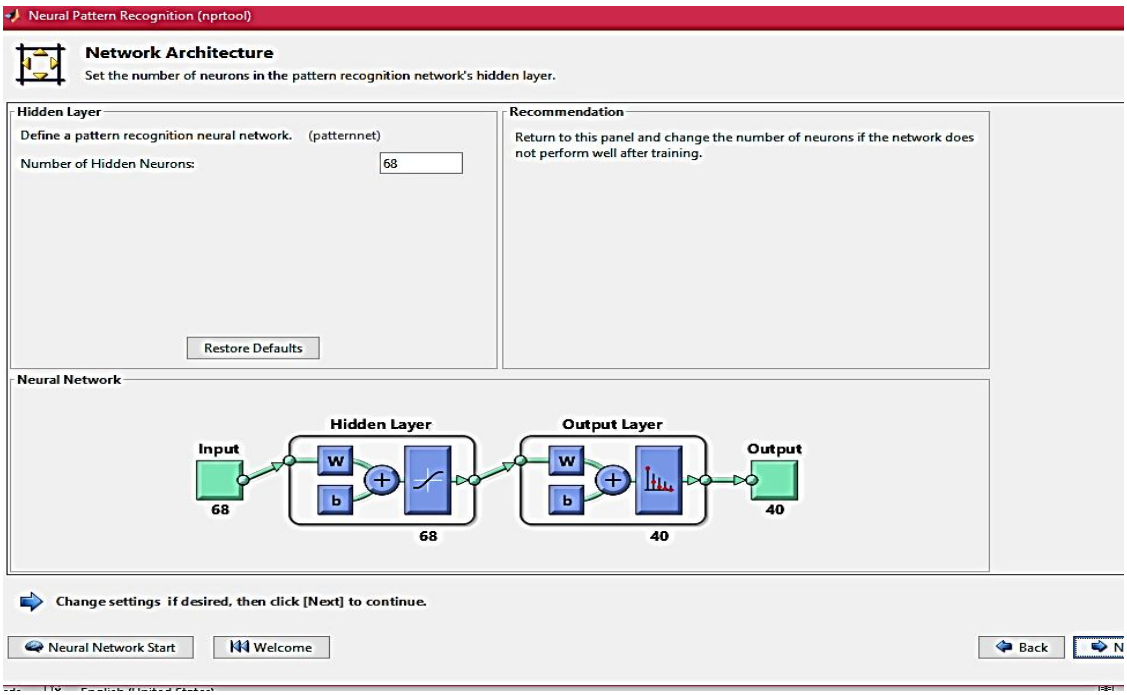


Fig. 4.3 Choose the hidden neuron amount

-We simulate

- The number of training vectors is increase; using our indice value we can increase or decrease the number of training vector to see if there is any improvement of performance.
- Testing the final network result is shown in the Table 5.9 and Table 5.10

4.5.3 COMPREHENTION EXAMPLE

When there are only two classes a path can be used; for each scalar target value 1 or 0, we define it by specifying the class to which the corresponding entry belongs. For example, we can define the problem of classification or exclusivity of two classes as follows:

inputs = [0 1 0 1; 0 0 1 1];

targets = [0 1 0 1; 1 0 1 0];

The vectors representing the target vectors have H elements, where for each target vector, it is divided into 2; one element is 1 and the others are 0. All the elements pose a base problem where inputs should be classified into H different classes. As an example, the lines highlighted show how to define a classification problem that is divides the corners of a 5-by-5-by-5 square into three classes:

- The first input vector representing the origin in one class.
- The last input vector representing the corner farthest from the origin in a second class.
- The third class representing other points.
- inputs = [0 0 0 0 5 5 5 5; 0 0 5 5 0 0 5 5; 0 5 0 5 0 5 0 5];
- targets = [1 0 0 0 0 0 0 0; 0 1 1 1 1 1 1 0; 0 0 0 0 0 0 0 1];

Either format can be used to resolve classification problems involving only two classes. The formation of the targets consists of either 1/0 scalar elements or two-element vectors, one of which is 1 and the other element is 0.

a-As shown in (Fig. 4.1) selection of target data and input vectors.

b-Random input vectors and target vectors are divided into three data sets as follows (Fig. 4.2):

70% are used for network formation and

15% of original data is set for validation and test data • 15% are Used to validate the fact that the network is generalized and to stop the formation before over fitting (Fig. 4.2)

• 15% are used at the end for a separate test of network generalization,

c-The default number of hidden neurons is set to 10. The standard network used for pattern recognition is a two-layer feedforward network-layer feedforward network, with a sigmoid transfer function in the hidden layer, and a softmax transfer function in the output layer. The default number of hidden neurons is set to 10. We are going to come back and increase this number of hidden neuron if the network has a poor performance as you expect (Fig. 4.4). The number of output neurons is set to number of classe; here it is 40 and 100 for the two data sets, which equate to the number of elements in the target vector (the number of categories).

1) Then we train the network using the ''*traing*'' algorithm (Fig. 4.4); we repeat it till we get satisfactory answer for the result.

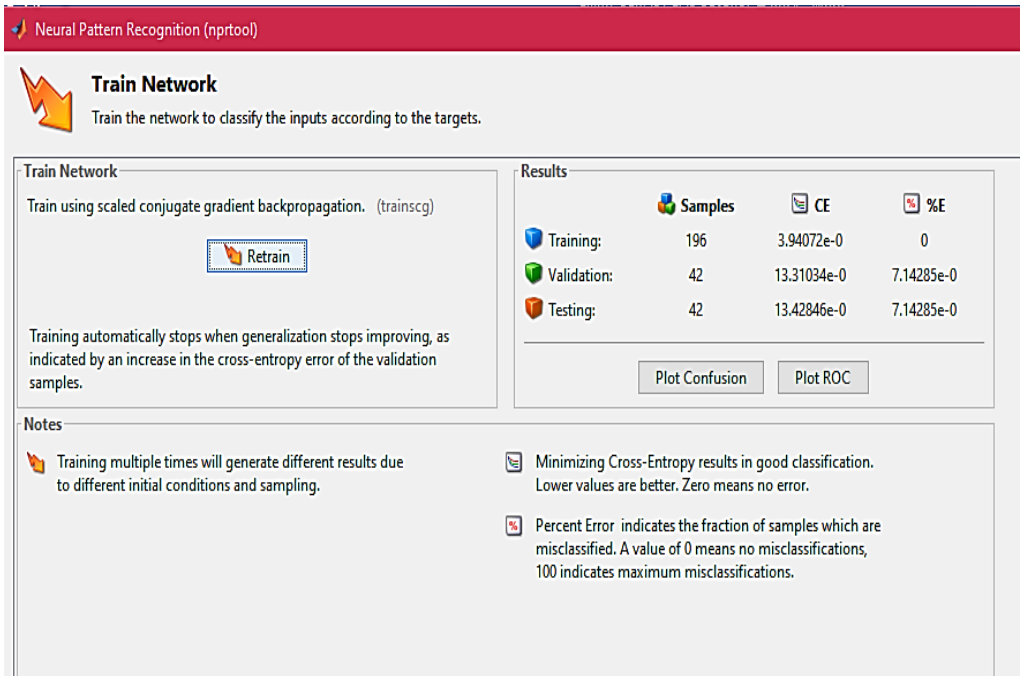


Fig. 4.4Training the network

The Fig. 4.5 indicates the iteration, when the validation performance has attained certain minimum. We continued the training for more than 6 iterations before we stopped. The test curve and the validations look very similar. The test curve hadn't increased significantly before the validation curve increased, then the overfitting has not occurred.

Fig. 4.5-Network performance using PCA

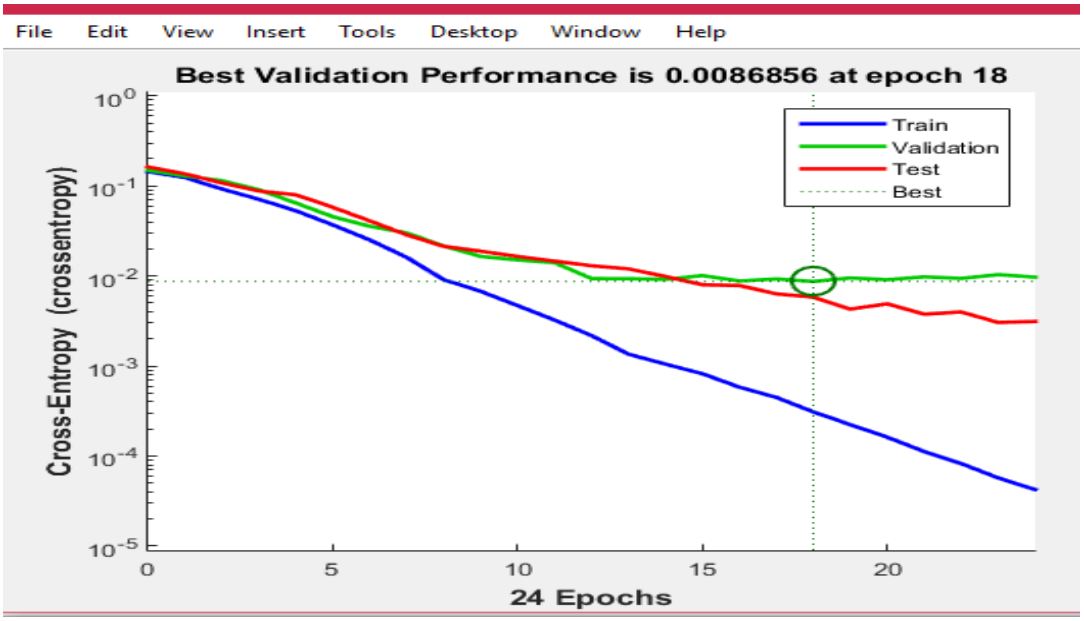


Table 5.10: Using PCA for ATT&T data base

Coef. of Principal					Total faces: 40	
Components (used to varied the amount of value components) indice	features vectors Amount	Performance respond computation times		Percentage of recognition success With unknown images	Numbers of image per classes :10	
		Training(s)	Testing MSE (s)		Training	Testing
0.85	*57*	$3.7x10^{-7}$	$8.22x10^{-3}$	***90.8%***	*6x30*	*4x30*
		$9.34x10^{-7}$	$6.94x10^{-3}$	***93.3%***	*7x30*	*3x30*
0.86	*62*	$4.85x10^{-7}$	$8.29x10^{-3}$	***89.9%***	*6x30*	*4x30*
		$6.72x10^{-7}$	$6.87x10^{-3}$	***91.1%***	*7x30*	*3x30*
0.87	*68*	$6.11x10^{-7}$	$9.73x10^{-3}$	***89.5%***	*6x30*	*4x30*
		$5.14x10^{-7}$	$6.77x10^{-3}$	***92.2%***	*7x30*	*3x30*
0.88	*75*	$6.85x10^{-7}$	$7.63x10^{-3}$	***90.8%***	*6x30*	*4x30*
		$5.92x10^{-7}$	$8.83x10^{-3}$	***90%***	*7x30*	*3x30*
0.9	*90*	$3.53x10^{-7}$	$1.51x10^{-2}$	***88.9%***	*6x30*	*4x30*
		$5.13x10^{-7}$	$1.06x10^{-2}$	***92,2%***	*7x30*	*3x30*

4.6 TABLE RESULT FOR PCA BASED FACE RECOGNITION

The overall result is indicated in the Table 5.9 for the first data base and Table 5.10 for the second database.

Table 5.9 using PCA only for database from Essex University

Coeff of PCA (used to vary the amount of value components) indices	feature s vectors Amount	Performance respond computation times		percentage of recognition success With unknown images	Total image face 20X 40 Numbers of image per classes :20	
		Training(cs)	Testing MSE (cs)		Training	Testing
0.85	*16*	$8.9x10^{-6}$	$14.20x10^{-2}$	***96.8%***	*15x40*	*5x40*
		$11.32x10^{-6}$	$8.88x10^{-2}$	***98.3%***	*16x40*	*4x40*
0.86	*25*	$10.75x10^{-6}$	$12.41x10^{-2}$	***95.9%***	*8x40*	*12x40*
		$9.75x10^{-6}$	$11.26x10^{-2}$	***94.4%***	*10x40*	*5x40*
0.87	*26*	$11.30x10^{-6}$	$14.75x10^{-2}$	***97.5%***	*8x70*	*12x40*
		$10.19x10^{-6}$	$11.98x10^{-2}$	***96.5%***	*10x70*	*5x40*
0.88	*27*	$9.88x10^{-6}$	$13.05x10^{-2}$	***98.00%***	*15x40*	*5x40*
		$11.82x10^{-6}$	$14.85x10^{-2}$	***98,2%***	*9x40*	*11x30*
0.9	*34*	$6.56x10^{-6}$	$8.06x10^{-2}$	***97.3%***	*8x40*	*12x40*
		$8.20x10^{-6}$	$9.089x10^{-2}$	***98,2%***	*10x40*	*5x40*

In Fig. 4.6, there are three blocks of colors. The confusion matrix highlights the percentage of correct and incorrect classifications. Correct classifications are the green squares on the matrices diagonal. Incorrect classifications form the red squares. The blue square is the overall performance of correct and incorrect data. The dark square gives performance of correct and misclassified data of each class

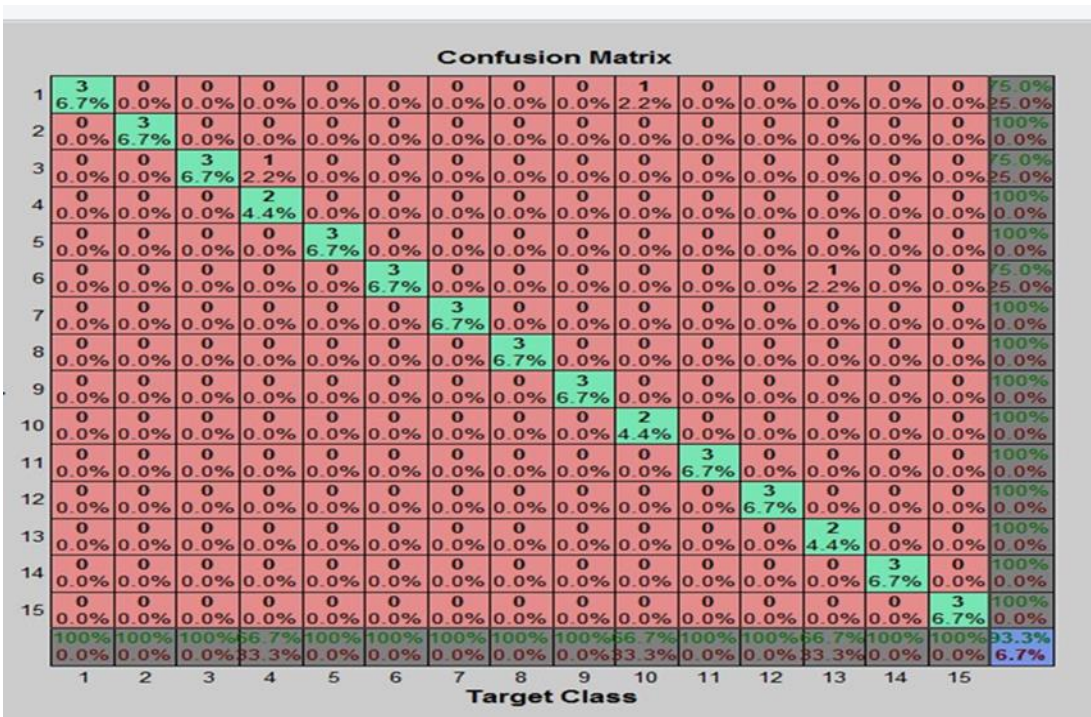
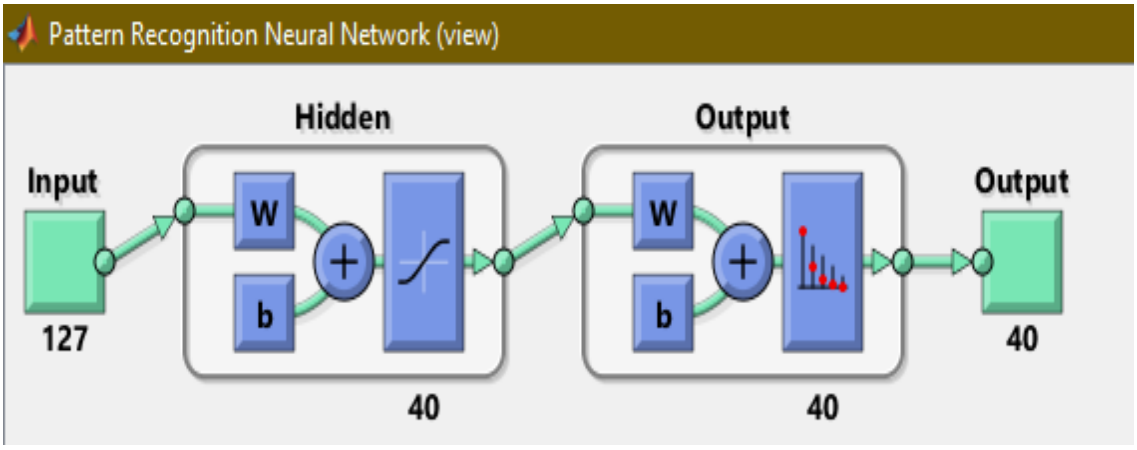


Fig. 4.6 Confusion matrix for PCA using the university of Essex data set

4.7. TABLE ANALYSIS

The Table 5.9 and 5.10 give us a result for the two-sample database. The first column indicated the coefficient uses in the program to increase or decrease the number of principal component number; here consider as the feature amount vector. We notice that the response time is relatively equal for all the number of features selected; however, the higher recognition rate recorded is obtained when the combination of features value and training data are at the highest. That means the network should be train with a significant amount of value to tune the neurons to the optimum value.

4.8 GRAPHICAL DIAGRAM OF THE NEURAL NETWORK



4.9 CONCLUSION FOR PCA

PCA is a very versatile technique in the sense that the dimensionality reduction of the global data is achieved but also the selection of the most significant value of the transform data. In the example of face recognition above we are dealing with multiple dimension; although we could still have a good result as in the Table 5.9 and Table 5.10, more than 91.85%, the technique may fail when we are dealing with many sample components. This is largely due from the fact in the input space the PCA assumes that the input data is real and continuous, and it approximates normality in the input space.

We could therefore associate the PCA with another features extraction method to boost the recognition rate because it will be independent from each other. This is going to be the object of the next chapter

CHAPTER 5

FACE RECOGNITION USING TWO-DIMENSIONAL DISCRETE COSINE TRANSFORM

5.1 DEFINITION

The DCT is a technique of sinusoids signal transformation with varying magnitudes and frequencies. The DCT2 function computes the two-dimensional discrete cosine transform (DCT) of an image from a spatial representation into a frequency representation. We could transform an image into its frequency components and discarded the higher frequency coefficients because the energy transformation of any image is concentrated in the lower frequencies part; by doing that the image is represented using a reduced amount of data components needed; at the same time without compromising in image quality. A discrete cosine transforms (DCT) expresses a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies. (Fig. 5.2)

Fig. 5.1 Face image picture

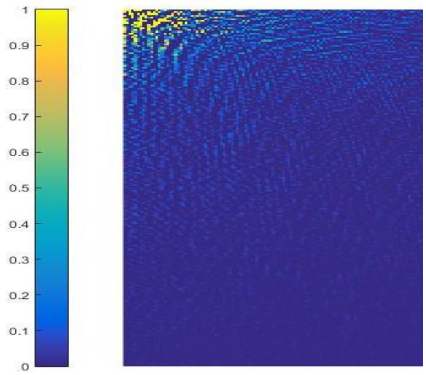


Fig. 5.2 DCT2 Image representation

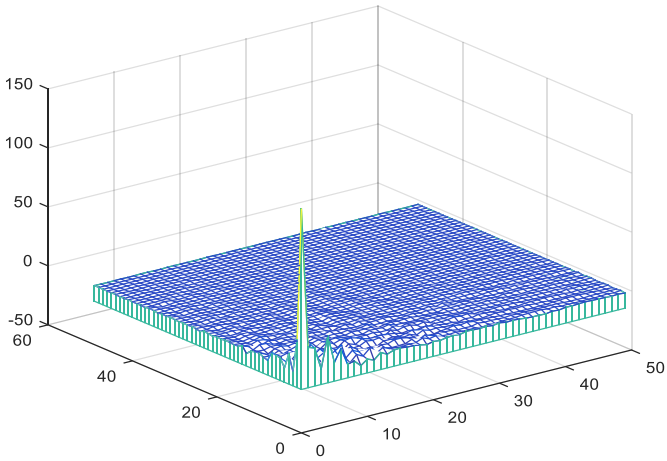


Fig. 5.3 Mesh plot 2DCT representation

One of the reason we combine PCA and DCT in that dissertation is that DCT is strongly correlated to Markov processes, the DCT can approach the compaction efficiency of the Karhunen-Loève transform or PCA .(which is optimal in the decorrelation sense) DCT is often used in signal and image processing, especially for lossy data compression, because it has a strong “Energy Compaction” property: the majority of the signal information after the transformation is concentrated in a low-frequency components of the DCT(Fig. 5.3) , approaching the KLT for signals based on certain limits of Markov processes.Through the DCT transform of an image ,we have a set of numbers called coefficients where a useful coefficient is determined by its variance over a set of images (Fig. 5.4). Therefore, some coefficient can be

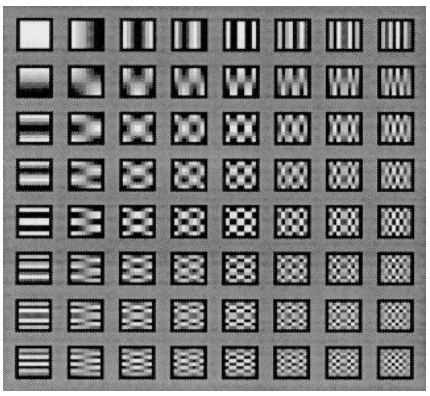


Fig. 5.4. 2DCT repartition value

removed without affecting the picture quality if those coefficients have few variances over a set. And during a process called quantization which step is at the heart of compression, the sole frequencies that still available after the selection are used to retrieve the image. This lead to the reconstructed image having some distortion or blurred appearance.

5.2 DCT VERSUS DFT

The DCT is like the discrete Fourier transform in the sense that it transforms a signal or image from the spatial domain into a frequency domain; the discrete cosine transformation is a variant of the Fourier transform (Fourier series). The normal idea behind the two transformations is that any (periodic) function can be broken down into sine and cosine functions, the frequencies being multiples of the given frequency.

In comparison, the discrete cosine transforms (DCT) transformation is a true transformation that transforms a sequence of real data points into your real spectrum, thereby bypassing the question of redundancy. The Discrete Fourier Transform (DFT) transforms a complex signal into its complex domain. However, for the real value of the signal in most applications, almost half of the data does not make sense. In the time domain, the imaginary part of the signal is all NULL; in the frequency domain, the real part of the spectrum is even symmetrical and imaginary part strange (mathematical explanation in paragraph 5.2.

Also, as DCT is derived from DFT, all the desirable properties of DFT (such as the fast algorithm) are obtained, that is, a DCT is a Fourier-related transform similar to the Discrete Fourier Transform (DFT), but only with real numbers. The DCTS are typically related to the coefficients of the Fourier series of a periodically and symmetrically extended sequence, while DFTs are related to the Fourier series coefficients of a periodically extended sequence. Working on real data with even symmetry, since the Fourier transform of a real and even function is real and uniform, the DCTs are equivalent to about twice as long DFT, while in some variants, the input data and output are offset by half.

5.2.1 PRINCIPLE OF DCT

For example, a simplified JPEG compressor is done using the following step:

a Reduce a face image till into blocks of 8x8 pixels

b Applied an 8x8 2DCT on each block

c The resulting coefficients is quantize (i.e. remove irrelevant to reduce the file size – good to note that JPEG does this by dividing the coefficients by a quantization matrix to get long runs of zeros)

d Using a lossless method (Huffman, Arithmetic coding, etc) let’s compress the quantized coefficients

5.2.2 APPLICATION OF DCT

The DCT-II and DCT are for the most part useful in the processing of images and signals, in this case for lossy compression. They are important for many applications in computer science and engineering, lossy audio compression, because DCTs have a strong property of "energy compaction"; in typical applications. Most of the low-

frequency component signal information from the DCT (eg MP3), to the images (JPEG, for example) (where small high frequency components can be removed), to the spectral methods for the numerical solution of the differential partial equations. Since we only need a small amount of cosine functions to estimate a typical signal, the use of cosine rather than sine functions is of paramount importance for compression, whereas for differential equations, cosines express a particular choice of boundary conditions. There are eight standard DCT variants, four of which are common. The most common variant of the discrete cosine transform is Type II DCT, often referred to simply as "DCT".

As explained above, this results from the implicit boundary conditions in the cosine functions.

5.2.3 MATHEMATICAL DERIVATION OF 1 AND 2DCT.

A three-dimensional signal is the representation or description of a signal of a graphic image. It can be described as follows: the amplitude of the signal considered as the Z axis is the value of the pixel a (X, Y), while the two dimensions of the screen form the X and Y axes. Visually express in a two-dimensional array where each numerical value of the pixel at that location.

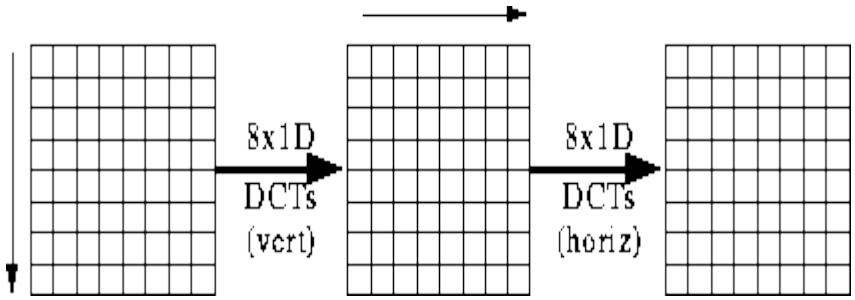


Fig. 5.5 Pixel projection representation (1)

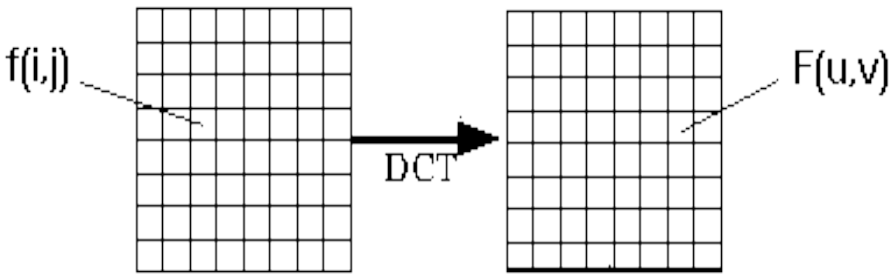
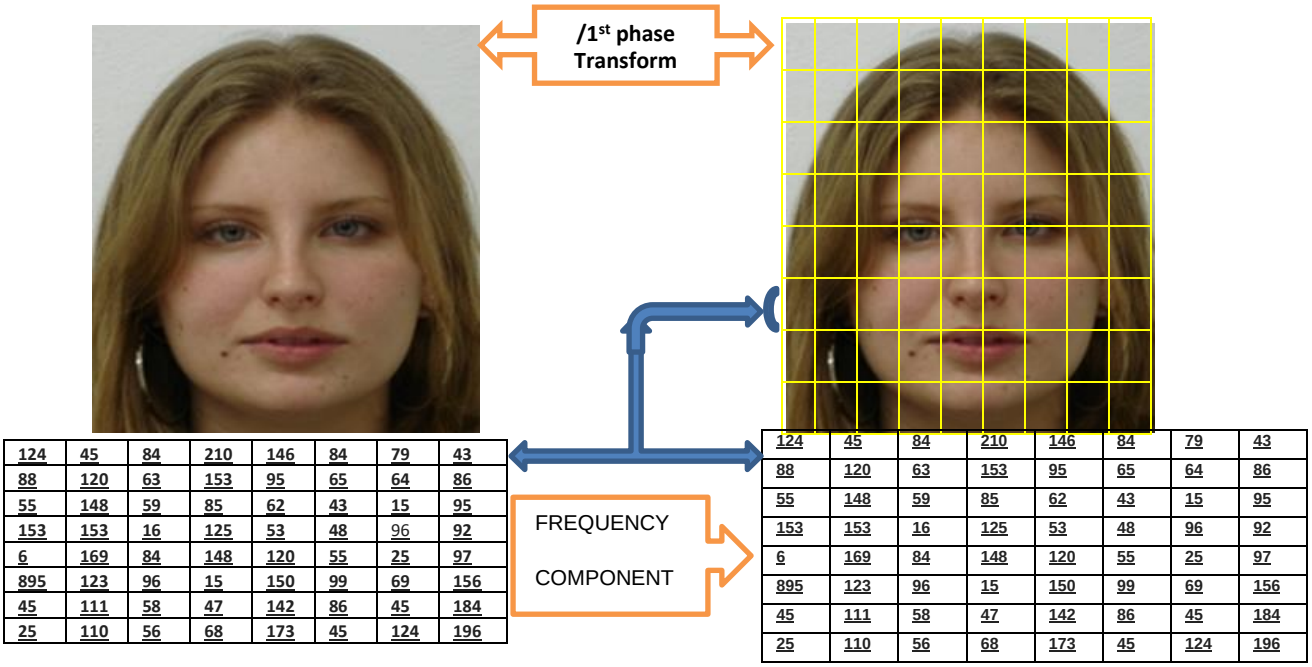


Fig. 5.6 Pixel projection representation (2)

As the particularity of a two-dimensional DCT matrix are rather complex, we will make it easy to understand the problem by first considering the derivation and intentions of a one-dimensional DCT matrix. The discrete cosine transforms (DCT) helps separate the image into parts (or spectral sub-bands) of different importance (with respect to the image's visual quality).



124	45	84	210	146	84	79	43
88	120	63	153	95	65	64	86
55	148	59	85	62	43	15	95
153	153	16	125	53	48	96	92
6	169	84	148	120	55	25	97
895	123	96	15	150	99	69	156
45	111	58	47	142	86	45	184
25	110	56	68	173	45	124	196

124	45	84	210	146	84	79	43
88	120	63	153	95	65	64	86
55	148	59	85	62	43	15	95
153	153	16	125	53	48	96	92
6	169	84	148	120	55	25	97
895	123	96	15	150	99	69	156
45	111	58	47	142	86	45	184
25	110	56	68	173	45	124	196

Fig. 5.7 DCT transformation

The image is fragmented into 8x8 groups, each containing 64 pixels. (Fig. 5.7) one of these 8x8 groups are magnified in this figure, highlighted the values of the single pixels, a single byte value between 0 and 255.

5.3 THE EQUATION OF DCT

5.3.1 DCT-1

As shown in the proximity image, the representation shown in Figure 5.7 is an operation that aims to transform a signal from one type of representation to another. In the signal conversion, that is, spatial information, into digital data ,that is, "frequency or spectral" information, so that all the information contained in the image can exist in a quantitative form and then used for compression. In the previous

passage we saw that the signal for an image can be considered as a three-dimensional signal. The amplitude of the signal, the Z axis, is the value of the pixel at (X, Y), while the X and Y axes of the image signal are the two dimensions of the screen. The complexity of the specificity of a two-dimensional DCT matrix forces us to simplify the problem by first considering the derivation and the intentions of a one-dimensional matrix.

The function F: $R^n \rightarrow R^n$

Here R denotes the set of real numbers; or equivalently on n x n matrix.

For a 1D ,the general equation (for *N* data elements) DCT is express by the equation that follow:

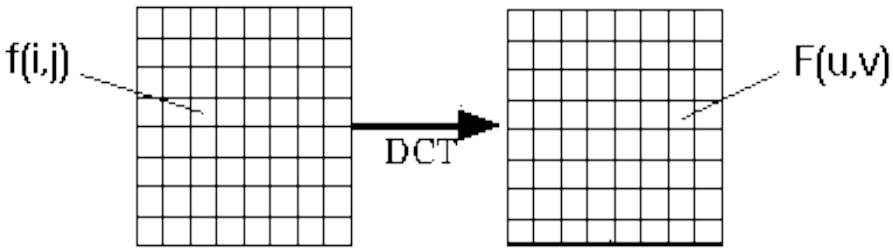


Fig. 5.8 Pixel projection representation (2)

The DCT equation calculate the i,jth entry of the DCT of an image.(Fig. 5.8)

$$F(u) = \left(\frac{2}{N}\right)^{\frac{1}{2}} \sum_{i=0}^{N-1} \Lambda(i).cos\left[\frac{\pi u}{2.N}(2i+1)\right] f(i) \qquad (5.1)$$

and the matching ***inverse*** 1D DCT transform is simple $F^{-1}(u)$, i.e.:

where:

$$\Lambda(i)=\begin{Bmatrix} \frac{1}{\sqrt{2}} \; for\; \xi = 0 \\ 1 \;\; otherwise \end{Bmatrix}$$

5.3.2 DCT-II

5.3.2.1 DESCRIPTION

In Section 3.1 the description of the one-dimensional DCT is applicable for two-dimensional images. The creation of two-dimensional cosine basis functions from which compound sample waveforms are obtained by multiplying a vertically oriented

set of the same functions. It follows a logical sequence according to which the set of basic functions represents the functions the vertical frequencies.

By convention, the term DC for the vertical basic functions is at the top and the term DC of the horizontal basic functions is on the left. So, the left upper element of a two-dimensional DCT matrix contains a value that is almost always very large (Fig 5.8) which is of great importance to us, because we want to recover a significant element to be used in conjunction with the PCA for image recognition.

For a 2DCT of an image, the equation is given by:

$$D_{(i,j)} = \frac{1}{\sqrt{2N}} C_{(i)} C_{(j)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} p(x,y) \cos\left[\frac{(2x+1)i\pi}{2N}\right] cos\left[\frac{(2y+1)j\pi}{2N}\right] \quad (5.2)$$

$$C_{(u)} \begin{Bmatrix} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{Bmatrix} \quad (5.3)$$

The matrix p now represents the term p (x, y) of the element x, y^{th} of the image. The DCT is performed on N which is the size of the block. An input (i, j^{th}) of the transformed image from the pixel values of the original image matrix is calculated and for the block, a standard 8x8 block is used for the compression = 8 and x and y go from 0 to 7.

So D (i, j) corresponds to

$$D_{(i,j)} = \frac{1}{4} C_{(i)} C_{(j)} \sum_{x=0}^{7} \sum_{y=0}^{7} p(x,y) \cos\left[\frac{(2x+1)i\pi}{16}\right] cos\left[\frac{(2y+1)j\pi}{16}\right] \quad (5.4)$$

Since the DCT performs calculation the cosine function, the final matrix resulting to the equation will be dependent on the horizontal, diagonal and vertical frequency.

Taking back (2) and the condition set out below:

$$T_{i,j} = \begin{cases} \frac{1}{\sqrt{N}} & if\ \ i = 0 \\ \sqrt{\frac{2}{N}} cos\left[\frac{(2j+1)i\pi}{2N}\right] & if\ i > 0 \end{cases}$$

Taking for example our face image in Fig. 6.6 and extracted the 8x8 block, the matrix calculated is:

$$T=\begin{bmatrix} .3456 & .3581 & .3684 & .3521 & -.9845 & -.6345 & .8746 & .4456 \\ .3989 & .3659 & .0975 & .0052 & -.8971 & -.4785 & -.7894 & -.6989 \\ .4056 & .1453 & -.6524 & .1653 & .1248 & .2147 & .3694 & .4056 \\ .3896 & -.5479 & .2658 & -.4794 & -.7518 & -.1239 & -.1298 & .9896 \\ .2356 & -.2563 & -.5269 & -.1974 & -.3454 & .0125 & .4981 & -.5563 \\ .1235 & -.4521 & .7986 & -.2541 & .3979 & .3698 & .3536 & .3521 \\ .0895 & -.7413 & -.1201 & .4967 & -.4056 & .1477 & -.4719 & -.0413 \\ .0845 & -.1986 & -.4521 & -.1267 & .3916 & -.1987 & .4859 & -.1986 \end{bmatrix}$$

All the entries equal to $\frac{1}{\sqrt{8}}$ as expected from equation (4) has the first row $(j = 1)$. T is an orthogonal matrix, so the columns of t form an orthonormal set. DCT.

The orthogonality of T is important as the inverse of T is T' ´When doing the inverse which is relatively easy to calculate.

For our feature extraction process, we select the upper left corner of the DCT result when calculated since it is in that part that the energy is concentrated and where we have significant value that best described the face image as we see in the Fig. 6.2 and Fig. 6.3; that is the respective face and the DCT value.

$$\mathbf{U=}\begin{bmatrix} 154 & 123 & 123 & 123 & 123 & 123 & 123 & 136 \\ 192 & 180 & 136 & 154 & 154 & 154 & 136 & 110 \\ 254 & 198 & 154 & 154 & 180 & 154 & 123 & 123 \\ 239 & 180 & 136 & 180 & 180 & 166 & 123 & 123 \\ 180 & 154 & 136 & 167 & 166 & 149 & 136 & 136 \\ 128 & 136 & 123 & 136 & 154 & 180 & 198 & 154 \\ 123 & 105 & 110 & 149 & 136 & 136 & 180 & 166 \\ 110 & 136 & 123 & 123 & 123 & 136 & 154 & 136 \end{bmatrix}$$

Fig. 5.9 Picture image data

$$\mathbf{DCT(U)=}\begin{bmatrix} 162.3 & 40.6 & 20.0 & 72.3 & 30.3 & 12.5 & -19.7 & -11.5 \\ 30.5 & 108.4 & 10.5 & 32.3 & 27.7 & -15.5 & 18.4 & -2.0 \\ -94.1 & -60.1 & 12.3 & -43.4 & -31.3 & 6.1 & -3.3 & 7.1 \\ -38.6 & -83.4 & -5.4 & -22.2 & -13.5 & 15.5 & -1.3 & 3.5 \\ -31.3 & 17.9 & -5.5 & -12.4 & 14.3 & -6.0 & 11.5 & -6.0 \\ -0.9 & -11.8 & 12.8 & 0.2 & 28.1 & 12.6 & 8.4 & 2.9 \\ 4.6 & -2.4 & 12.2 & 6.6 & -18.7 & -12.8 & 7.7 & 12.0 \\ -10.0 & 11.2 & 7.8 & -16.3 & 21.5 & 0.0 & 5.9 & 10.7 \end{bmatrix}$$

Fig. 5.9.1 DCT values of image data

5.4 PRACTICAL APPLICATION EXPERIMENT FOR DCT

5.4.1 FLOWCHART

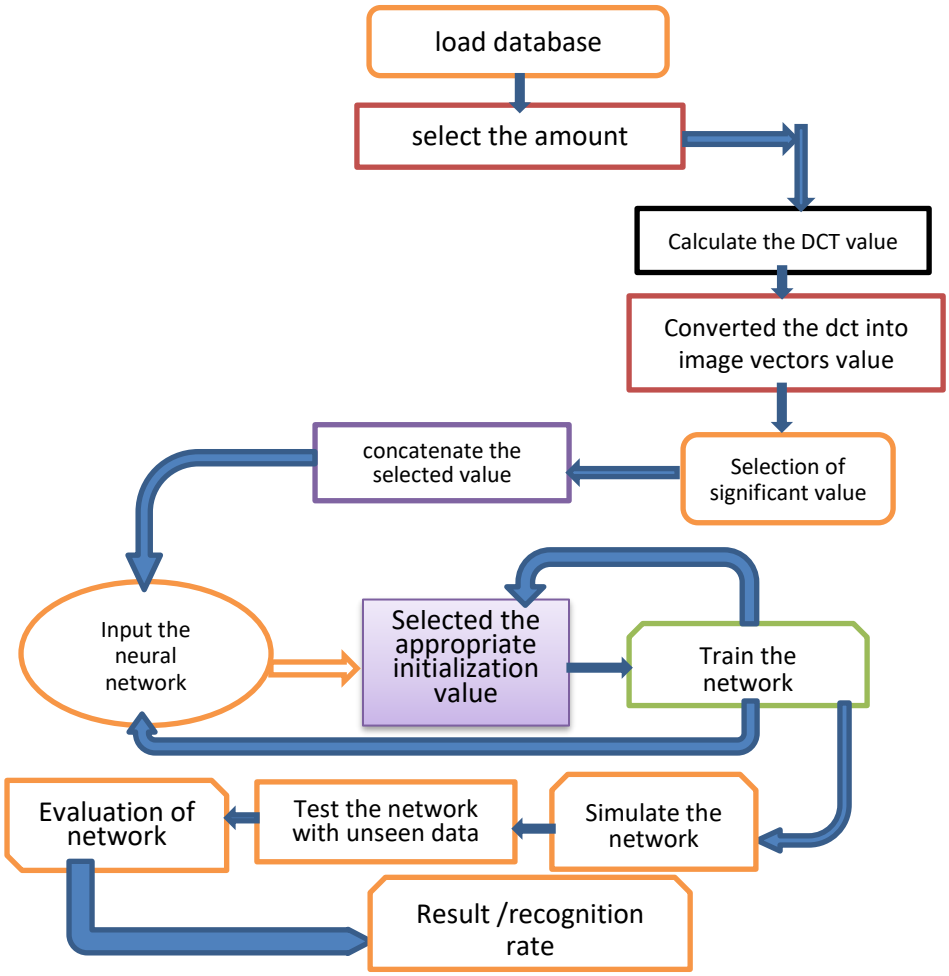


Fig.5.9.2 Algorithm use to write the program

5.5 EXPERIMENTAL PHASE FOR FACE RECOGNITION

5.5.1 SEQUENTIAL WORKING PROCEDURE

As in the PCA, the technique of 2DCT transform the original signal into its frequency components and discard other less significant components. By that we reduce the original signal into a number of input vectors constituting the total feature vector which are grouped as columns in a matrix. The selection of features is taken

from the left higher part of the original image data (Fig. 5.2). Then another set of S target vectors so that they indicate the classes to which the input vectors are assigned, here we have two sets; one of 40 classes for the first data set and another of 100 classes for the second data set.

The Fig. 5.5, Fig. 5.6, Fig. 5.7 are all identical in the working method when using the 2DCT method. Except the confusion matrix in Fig. 5.9.3

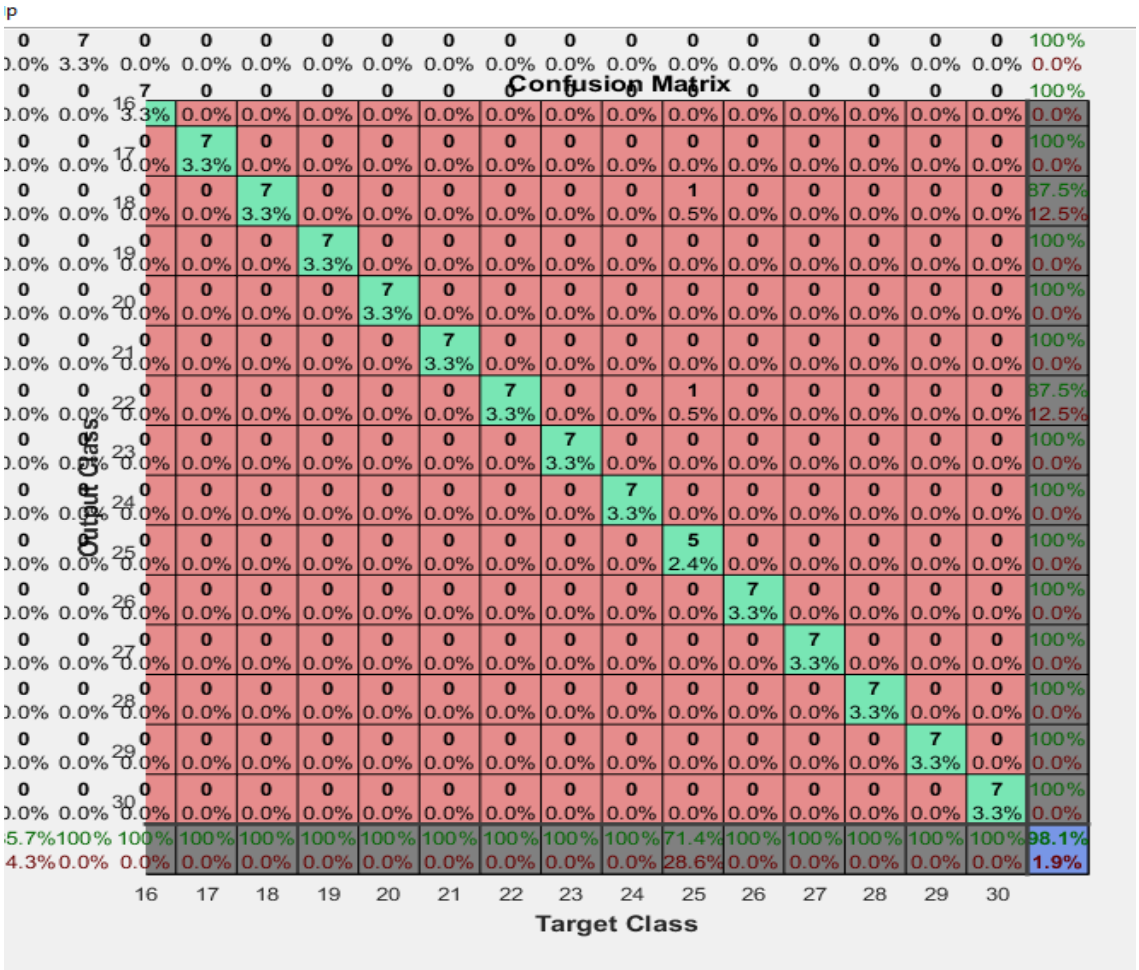


Fig. 5.9.3: Zoom Confusion matrix plot for 2DCT (with a zoom view, for clarity) for AT &T database

5.5.2 DESCRIPTION OF FIGURE

In Fig. 5.9.3, there are three blocks of colors. The confusion matrix shows the percentage of correct and incorrect classifications for the 2DCT. Correct classifications are the green squares on the matrices diagonal. Incorrect

classifications form the red squares. The blue square is the overall performance of correct and incorrect data. The dark square gives performance of correct and misclassified data of each class

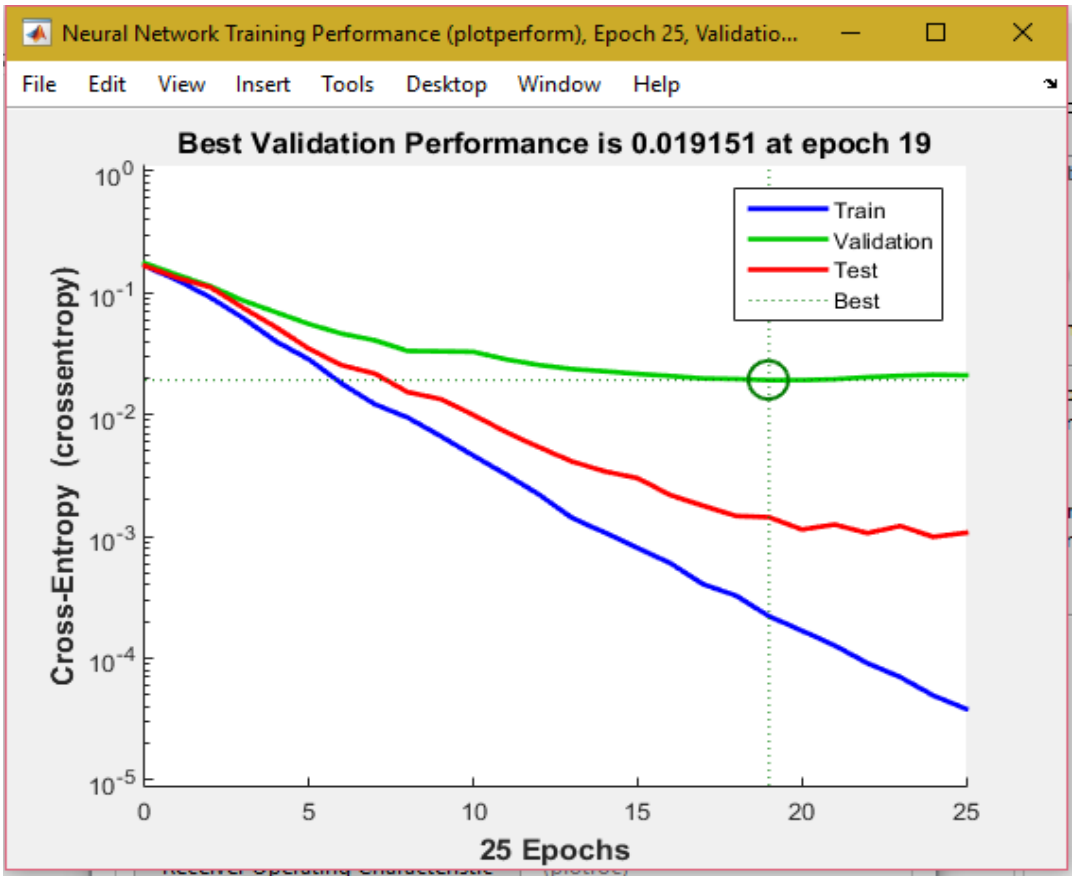


Fig. 5.9.4 Network performance for the 2DCT for AT&T database

The Fig. 5.9.4 indicates the iteration at which the validation performance meets a minimum. The training continued for 5 more iterations before the training stopped. The validation and test curves are very identical. It is possible that some overfitting might have occurred specially If the test curve had increased significantly before the validation curve increased.

5.6 TABLE RESULT FOR 2DCT BASED FACE RECOGNITION

The overall result is indicated in the Table 5.11 for the first data base AT& T database

Table 5.11 using 2DCT only for the AT&T database

2DCT value mxm total pixels number	Number of Epoch	Performance response time s	% centage recognition with unknown image	Total Number of faces images.10x40	
				Training	Testing
3x3	*31*	*1*	*81.2*	*7x40*	*3x40*
4x4	*32*	*1*	*90*	*7x40*	*3x40*
5x5	*33*	*0.8*	*92.61*	*7x40*	*3x40*
6x6	*35*	*0.7*	*90.26*	*7x40*	*3x40*
6x6	*30*	*1*	*88.67*	*6x40*	*4x40*
7x7	*31*	*0.9*	*87.03*	*5x40*	*5x40*
7x7	*32*	*0.7*	*93.75*	*7x40*	*3x40*
8X8	*32*	*0.8*	*90.25*	*6x40*	*4x40*
9x9	*31*	*1.0*	*91.4*	*6x40*	*4x40*
10X10	*32*	*1.0*	*90.97*	*6x40*	*4x40*

The Table 5.11 gives us a result for the one-sample database. The first column indicated the most significant frequency component used; here consider as the feature amount vector. We notice that the response time is relatively equal for all the number of features selected; however, the higher recognition rate recorded is obtained when the combination of features value and training data are at the highest. 93.75 That means that there is a tradeoff of 2DCT value to be selected as to avoid overfitting and less recognition rate like in the last row where 10x10 is the 2DCT value.

5.7. COMBINED PCA AND 2DCT METHOD

5.8 FLOWCHART OF COMBINED METHOD PCA AND 2DCT

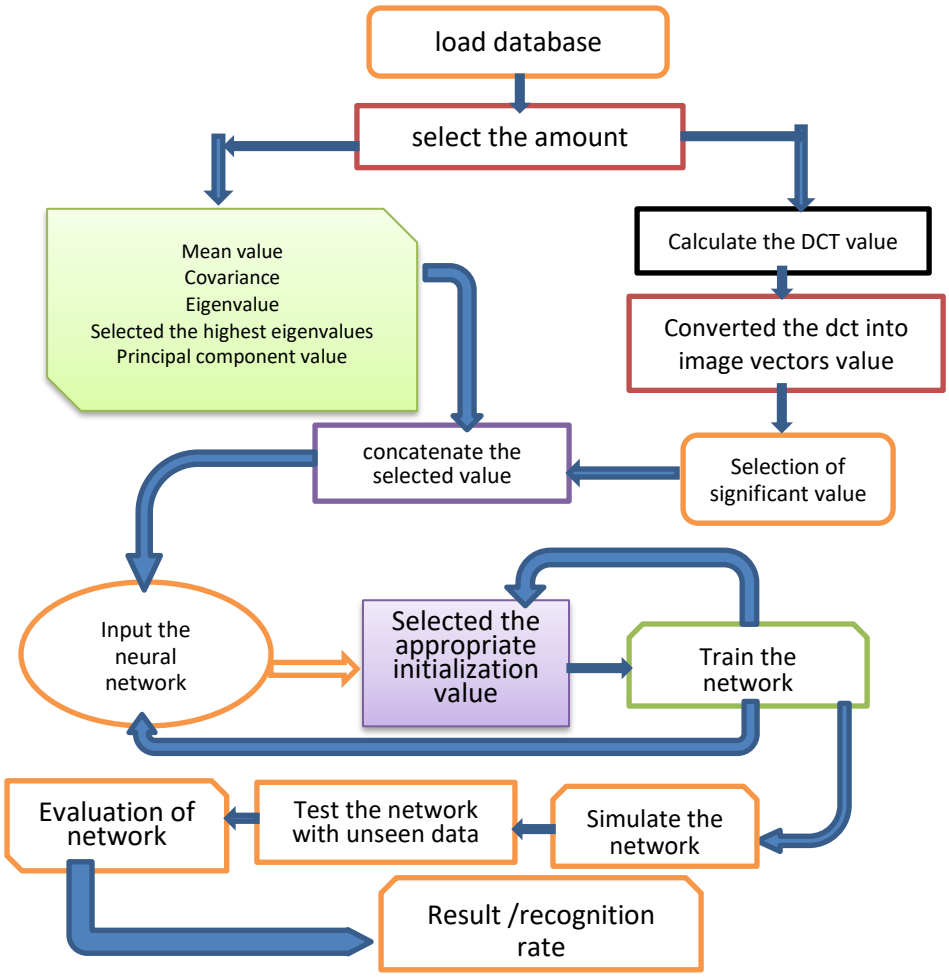


Fig.5.9.5 Algorithm use to write the program

5.8.1 TABLE RESULT USING 2DCT AND PCA

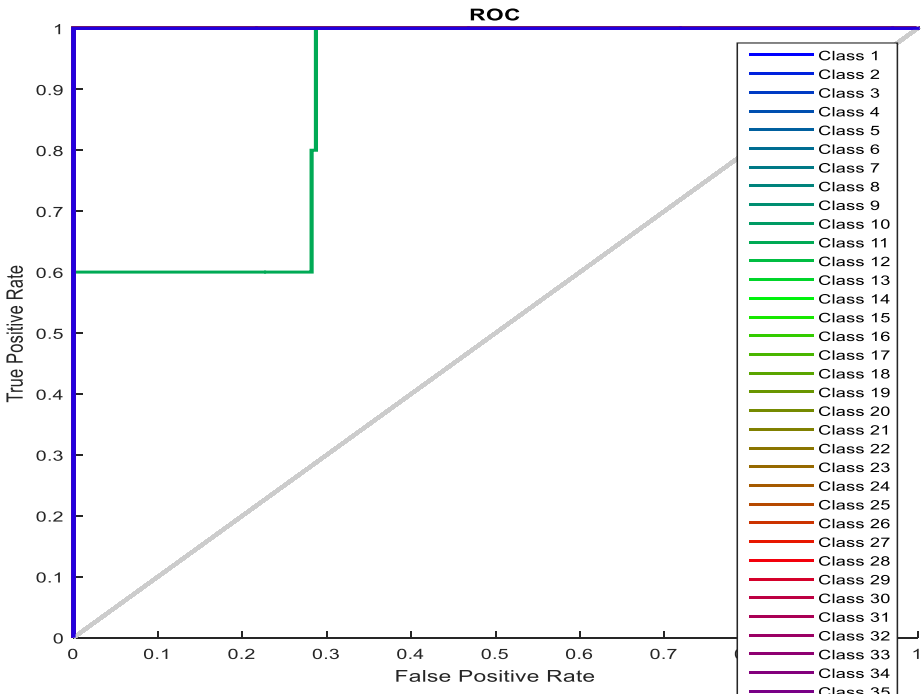
The Table (Table 5.12) is the visual representation of the result when we use the two features extractors. The overall recognition rate has substantially improved from the previous experiment: either the 2DCT or the PCA.

The first column is the area delimited where the concentration of the most significant value of the 2DCT is found, (see Fig.5.2). The second column are the chosen value of the Principal component analysis of the same image. Both of them are combined to give us the recognition rate indicated in the fourth column.

Table 5.12 Using 2DCT and PCA for AT&T database

2DCT Square array of most significant(R value)	*Pca Number*	*Performance response Computation time s*	*percentage recognition rate (%)*	*Total image faces 40X10*	
				Training	*Testing*
10x10	*44*	*3*	*93*	*7x40*	*3x40*
8x8	*44*	*2*	*92.8*	*7x40*	*3x40*
9x9	*44*	*2*	*91*	*7x40*	*3x40*
12x12	*44*	*5*	*93*	*7x40*	*3x40*
10x10	*44*	*6*	*92.5*	*8x40*	*2x40*
10x10	*44*	*4*	*92.6*	*6x40*	*4x40*
10x10	*57*	*6*	*89.5*	*6x40*	*4x40*
10x10	*68*	*6*	*91*	*8x40*	*2x40*
7x7	*82*	*4*	*96.5*	*7x40*	*3x40*
7x7	*44*	*4*	*94.9*	*6x40*	*4x40*
6x6	*44*	*4*	*94.8*	*7x40*	*3x40*
5x5	*17*	*3*	*97.5*	*7x40*	*3x40*
5x5	*17*	*3*	*94.86*	*6x40*	*4x40*

5.9 Result Graph using the second data base from the University of Essex



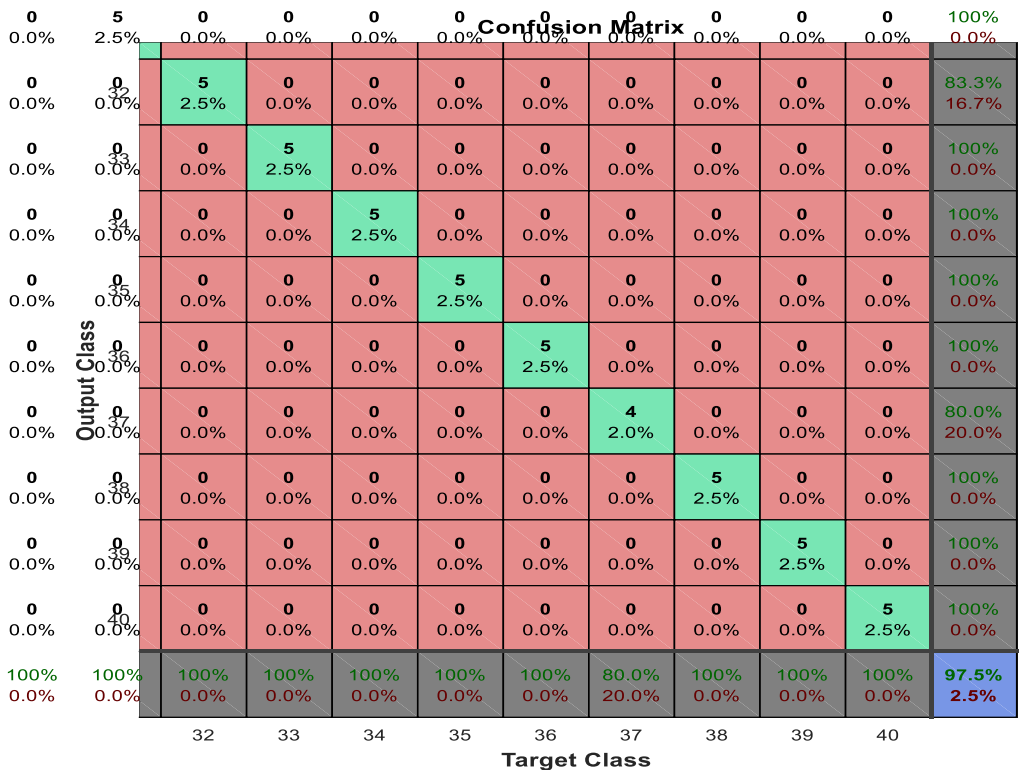


Fig. 5.9.4: Zoom Confusion matrix plot for 2DCT (with a zoom view, for clarity) for University of Essex database for different value of RX R

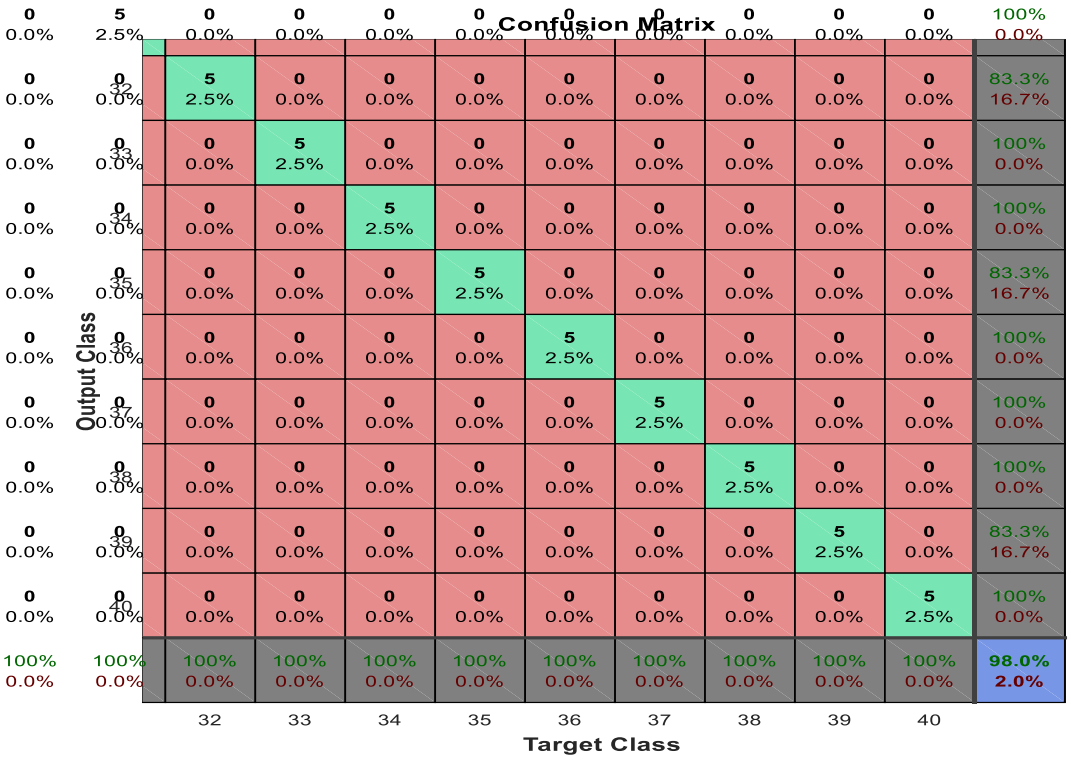


Fig. 5.9.5: Zoom Confusion matrix plot for 2DCT (with a zoom view, for clarity) for University of Essex database for different value of RX R

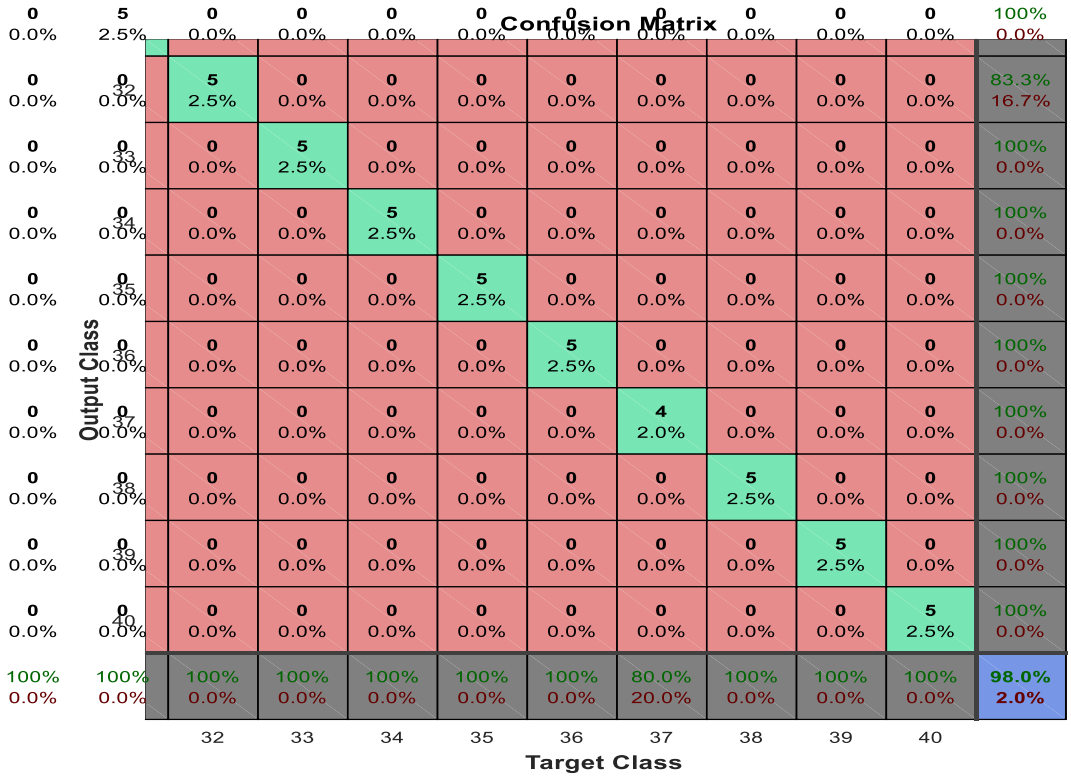


Fig. 5.9.4: Zoom Confusion matrix plot for 2DCT (with a zoom view, for clarity) for University of Essex database for different value of 7x7 R value

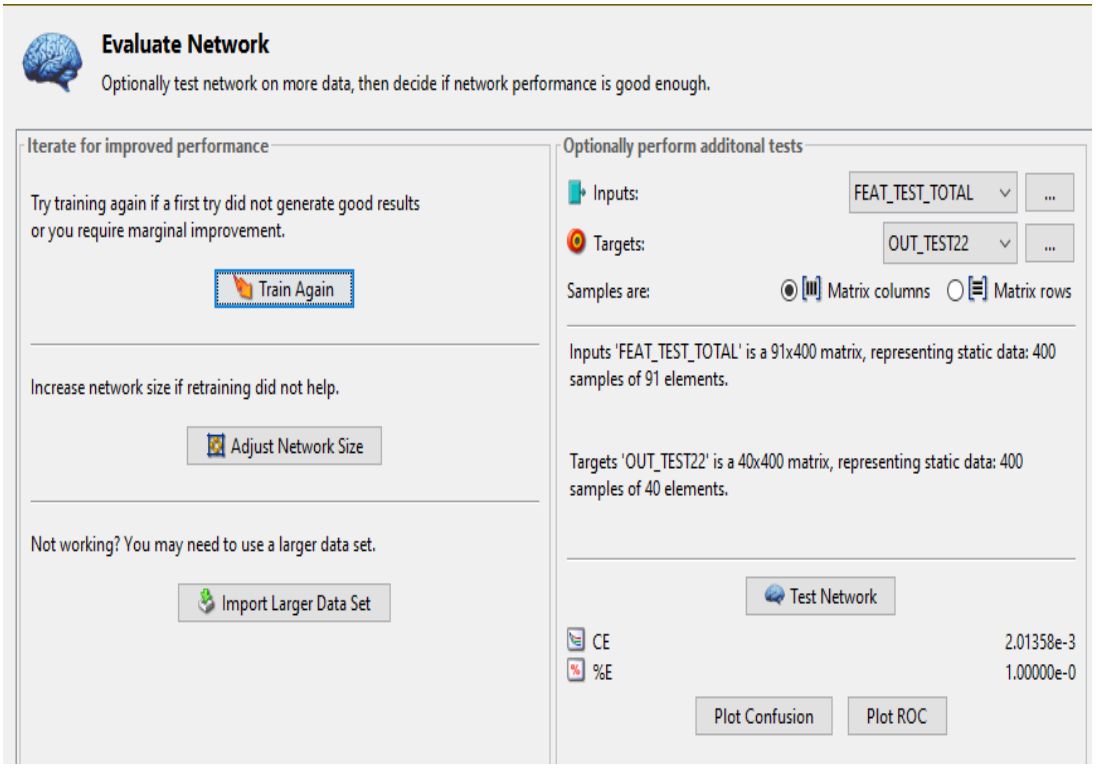


Fig 5.4.5 Training the Network using the Essex data base for Dct and Pca

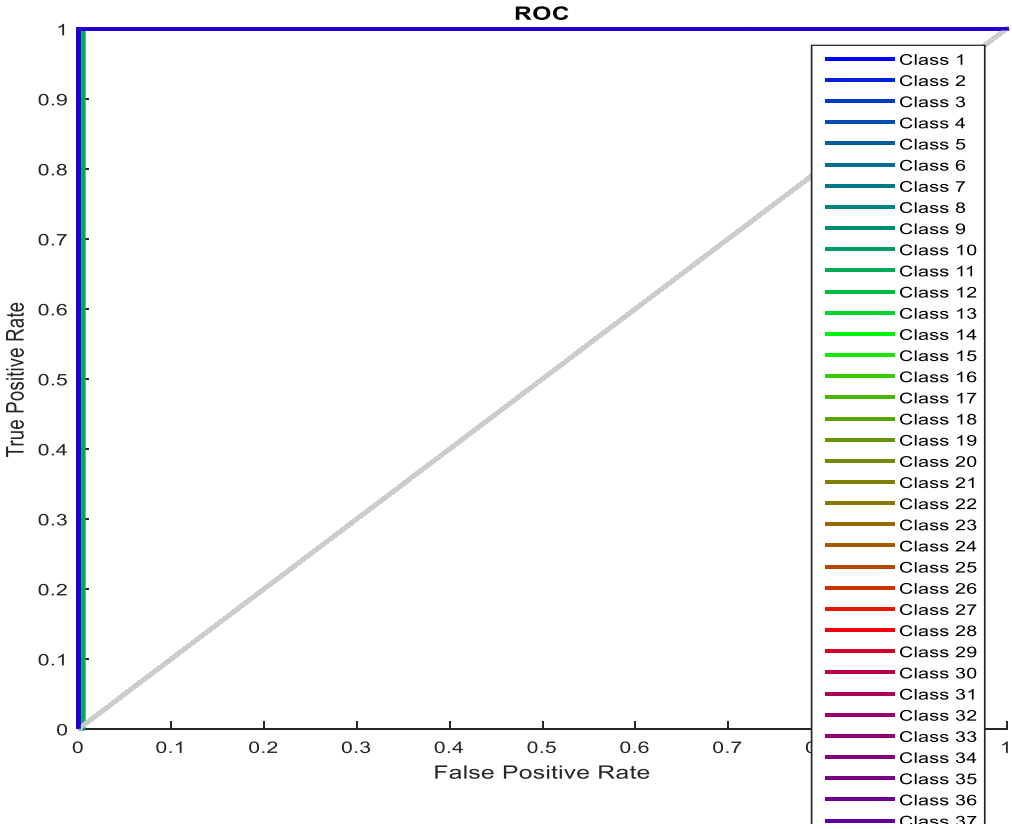
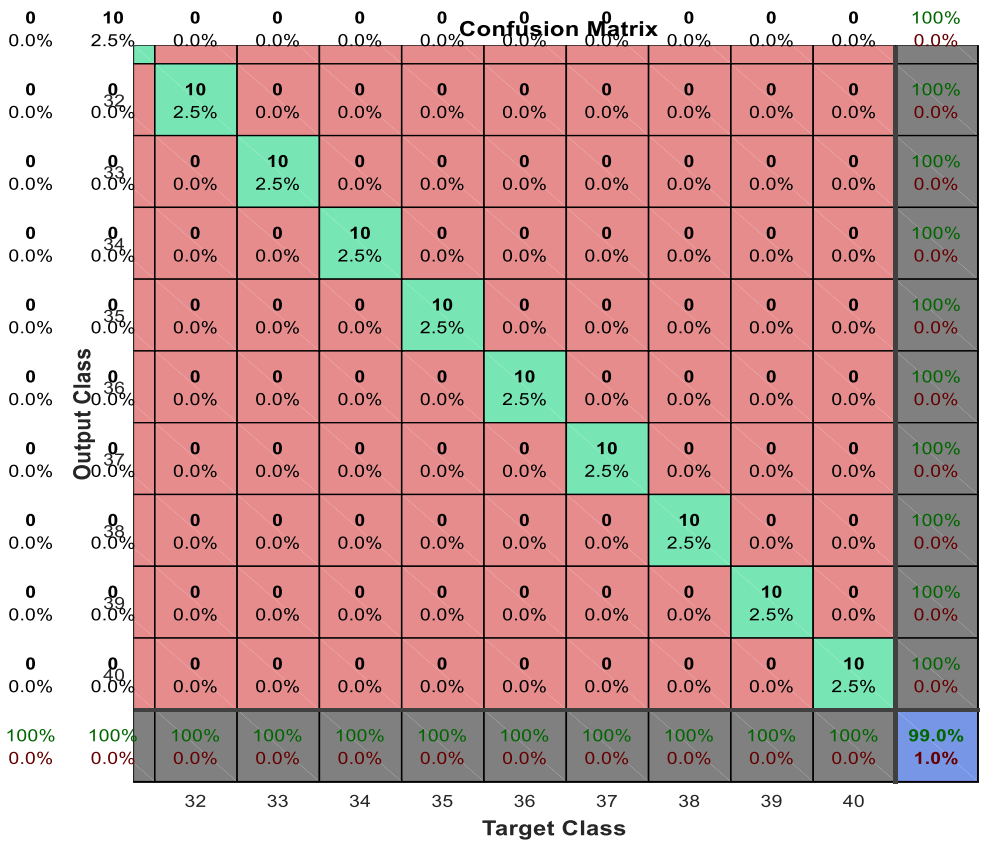


Table 5.13 Using 2DCT and PCA for the University of Essex database

2DCT Square array of most significant(R value)	*Pca Amount*	*Performance response Computation time cs*	*percentage recognition rate (%)*	*Total image faces 40X20*	
				Training	*Testing*
10x10	*27*	*1.01*	*99*	*12x40*	*8x40*
8x8	*24*	*1.01*	*99*	*15x40*	*5x40*
9x9	*25*	*1.02*	*98.99*	*12x40*	*8x40*
6x6	*22*	*1.01*	*98*	*15x40*	*5x40*
7x7	*23*	*1.01*	*98.7*	*15x40*	*5x40*


Confusion Matrix
Output Class
Target Class
32 33 34 35 36 37 38 39 40
100% 99.0% 1.0%

CHAPTER 6

GENERAL CONCLUSION AND FUTURE WORK

Faces images identification carried out here has given encouraging results; more than 98.6% recognition rate for first data base, and more than 99% for the second database using a combined method: PCA and 2Dct . We envisage more research in this area that could focus on testing a larger number of variables in the pictures and to be able to track and identified faces automatically. Another area to consider would be further research on the most relevant method to use for the minimization of feature extraction of images that are placed in the database on one hand and the consideration of ageing in the recognition rate as the people gets old, also introduced a method to prevent the spoofing on face image recognition using computer detection.

Furthermore, for face recognition it is not a paramount importance to matches face to face; we can extract small most relevant features like we did with 2DCT and still recognize the unknown faces; and it is where Eigen faces and principal component analysis are combined with other extraction method such 2D-DCT or others statistical extraction method to achieve even high accuracy result.

BIBLIOGRAPHY

[1] Bledsoe, W. W. 1968. Semiautomatic Facial Recognition, Technical Report SRI Project 6693, Stanford Research Institute, Menlo Park, California.

[2] Bledsoe, W. W. 1966a. Man-Machine Facial Recognition: Report on a Large-Scale Experiment, Technical Report PRI 22, Panoramic Research, Inc., Palo Alto, California.

[3] Goldstein, A. J., Harmon, L. D., & Lesk, A. B. (1971). Identification of human faces. Proceedings of the IEEE.

[4] T,Kenade. Picture Processing System by Computer Complex and Recognition of Human faces. PhD thesis, Kyoto University, November 1973.

[5] Sirovich, L. & Kirby, M. (1987). A low-dimensional procedure for the characterization of human faces. Journal of Optical Society of America.

[6] Babu, T. et. al (2019). Sleep Detection and Alert System for Automobiles, Advances in Manufacturing Technology.

[7] Fingerprints and Other Biometrics, https://www.fbi.gov/services/cjis/fingerprints-and-other-biometrics, access on 2019/10/02.

[8] McWhertor, M. (2009). Sony Spills More PS3 Motion Controller Details To Devs, https://kotaku.com/sony-spills-more-ps3-motion-controller-details-to-devs-30981868, access on 2019/10/02.

[9] Fischler M. and Elschlager. R. (1973). The representation and matching of pictorial structures. IEEE Transactions on Computers.

[10] Nabney I.T. (2002). NETLAB: Algorithms for Pattern Recognition, Springer.

[11] Higham D.J., Higham N.J. (2016). MATLAB Guide 2nd Edition. Third edition, SIAM-Society for Industrial and Applied Mathematics.

[12] Lowe, D.G. (2004). Distinctive image features from scale-invariant key points. International Journal of Computer Vision.

[13] Collection of Facial Images: Faces94, https://cswww.essex.ac.uk/mv/allfaces/faces94.html, access on 2019/10/02

[14] The Database of Faces (AT&T), https://git-disl.github.io/GTDLBench/datasets/att_face_dataset/, access on 2019/10/02

[15] Moller, M.F. (1993) A scaled conjugate gradient algorithm for fast supervised learning. Neural Networks.

[16] Nixon M.S. and Aguado A.S. (2013). Feature Extraction & Image Processing for Computer Vision. 3rd Edtion, Elsevier Ltd.

[17] Aggarwal C.C. (2018). Neural Networks and Deep Learning, Springer.

[18] Neural Network Toolbox, http://matlab.izmiran.ru/help/toolbox/nnet/model26.html, access on 2019/10/16.

[19] Hsu R. (2001). Face modeling for recognition. Proceedings 2001 International Conference on Image Processing

[20] Candes E.J., Li X. (2011). Robust Principal Component Analysis. Journal of the ACM

[21] Fleet D., Weiss Y. (2006) Optical Flow Estimation, Handbook of Mathematical Models in Computer Vision. Springer

[22] de Andrade N.N.G., Martin A., Monteleone S. (2013) "All the better to see you with, my dear": Facial recognition and privacy in online social networks. IEEE Security & Privacy.

[23] Chellappa, R. (1992) Digital Image Processing, second ed. IEEE Computer Society Press, Los Alamitos, CA.

[24] Cheng, C.H., Wang, P.S P. (2009). Handbook of Pattern Recognition and Computer Vision, fourth ed. World Scientific, Singapore.

[25] Cornsweet,T.N. (1970). Visual Perception. Academic Press

[26] Davies, E.R. (2005). Machine Vision: Theory, Algorithms and Practicalities, third ed. Morgan Kaufmann.

[27] Fisher, R.B., Dawson-Howe, K., Fitzgibbon, A., Robertson, C. (2005). Dictionary of Computer Vision and Image Processing. Wiley.

[28] Forsyth, D., Ponce, J. (2002). Computer Vision: A Modern Approach. Prentice Hall, Upper Saddle River, NJ.

[29] Parker, J.R. (2010). Algorithms for Image Processing and Computer Vision, second ed. Wiley, Indianapolis, IN.

[30] Petrou, M., Petrou, C. (2010). Image Processing—The Fundamentals, second ed. Wiley- Blackwell, London.

[31] Pratt, W.K. (2001). Digital Image Processing: PIKS Inside, third ed. Wiley, Chichester.

[32] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P. (2007). Numerical Recipes: The Art of Scientific Computing, third ed. Cambridge University Press, Cambridge.

[33] Prince, S.J.D. (2012). Computer Vision Models, Learning, and Inference. Cambridge University Press, Cambridge.

[34] Ratliff, F. (1965). Mach Bands: Quantitative Studies on Neural Networks in the Retina. Holden-Day Inc., San Francisco, CA.

[35] Rosenfeld, A., Kak, A.C. (1982). Digital Picture Processing, vols. 1 and 2, second ed. Academic Press, Orlando, FL.

[36] Russ, J.C. (2006). The Image Processing Handbook, sixth ed. CRC Press (Taylor & Francis), Boca Raton, FL.
[37]Schwarz, S.H. (2004). Visual Perception, third ed. McGraw-Hill, New York, NY.
[38] Shapiro, L.G., Stockman, G.C. (2001). Computer Vision. Prentice Hall, Upper Saddle River, NJ.
[39] Sonka, M., Hllavac, V., Boyle, R. (2007). Image Processing, Analysis and Machine Vision, third ed. Brooks/Cole, London.
[40] Szeliski, R. (2011). Computer Vision: Algorithms and Applications. Springer-Verlag, London.
[41] Pumperla M., Ferguson K. (2019). Deep Learning and the Game of Go, Manning Publications

APPENDIX 1. ETHICAL FORM APPROVED

UNISA university of south africa

UNISA SOE ETHICS REVIEW COMMITTEE

Date: 08/05/2018

ERC Reference # : 2018/CSET_SOE/NM/001
Name : Mr Nguela Mirabeau
Student #: 53571274
Staff #: N/A

Dear Mr Nguela Mirabeau

Decision: Ethics Approval from 08/05/2018 to 08/05/2021

Researcher(s): Name Mr Nguela Mirabeau
Address: 05 Jacoba street, Troyeville, Johannesburg, 2094.
E-mail address: mirabeaun2002@yahoo.com, telephone #: 011 056 8218

Supervisor (s): Name: Zenghui Wang
E-mail address: wangz@unisa.ac.za, telephone # 011 471 3513

Working title of research:

Face image recognition using 2 dimensional cosine transform (DCT) and principal component analysis (PCA)

Qualification: Masters

Thank you for the application for research ethics clearance by the Unisa SOE Ethics Review Committee for the above mentioned research. Ethics approval is granted for 3 years.

RECEIVED
2018 -05- 09
OFFICE OF THE EXECUTIVE DEAN
College of Science, Engineering and Technology

RECEIVED
OFFICE OF THE DIRECTOR
2018 -05- 08
School of Science, Engineering and Technology

Open Rubric

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

*The **low risk application** was **reviewed** by the SOE Ethics Review Committee on 08/05/2018 in compliance with the Unisa Policy on Research Ethics and the Standard Operating Procedure on Research Ethics Risk Assessment. The decision was approved on 08/05/2018.*

The proposed research may now commence with the provisions that:

1. The researcher(s) will ensure that the research project adheres to the values and principles expressed in the UNISA Policy on Research Ethics.
2. Any adverse circumstance arising in the undertaking of the research project that is relevant to the ethicality of the study should be communicated in writing to the SOE Committee.
3. The researcher(s) will conduct the study according to the methods and procedures set out in the approved application.
4. Any changes that can affect the study-related risks for the research participants, particularly in terms of assurances made with regards to the protection of participants' privacy and the confidentiality of the data, should be reported to the Committee in writing, accompanied by a progress report.
5. The researcher will ensure that the research project adheres to any applicable national legislation, professional codes of conduct, institutional guidelines and scientific standards relevant to the specific field of study. Adherence to the following South African legislation is important, if applicable: Protection of Personal Information Act, no 4 of 2013; Children's act no 38 of 2005 and the National Health Act, no 61 of 2003.
6. Only de-identified research data may be used for secondary research purposes in future on condition that the research objectives are similar to those of the original research. Secondary use of identifiable human research data require additional ethics clearance.
7. No field work activities may continue after the expiry date 08/05/2021. Submission of a completed research ethics progress report will constitute an application for renewal of Ethics Research Committee approval.
8. Field work activities may only commence from the date on this ethics certificate.
9. [Permission to conduct research involving UNISA employees, students and data should be obtained from the Research Permissions Subcommittee (RPSC) prior to commencing field work.] AND/OR
10. [Permission to conduct this research should be obtained from the [company, CE organisation, DoE, etc name] prior to commencing field work.]

University of South Africa
Preller Street, Muckleneuk Ridge, City of Tshwane
PO Box 392 UNISA 0003 South Africa
Telephone: +27 12 429 3111 Facsimile: +27 12 429 4150
www.unisa.ac.za

Add any other conditions if relevant.

Note:

The reference number ***2018/CSET_SOE/NM/001*** *should be clearly indicated on all forms of communication with the intended research participants, as well as with the Committee.*

Yours sincerely,

Signature.....

Dr T Sithebe

Chair of SOE ERC

E-mail: sithet@unisa.ac.za

Tel: (011) 429-3864

Signature.....

Prof BB Mamba

Executive Dean : CSET

E-mail: mambabb@unisa.ac.za

Tel: (011) 670-9230

APPENDIX 2. EXTRACTION OF FEATURES

```
% RESEARCH ON FACE IMAGE RECOGNITION USING MATLAB AND PCA
UNDER THE
% SUPERVISION OF PROF. WANG ZENGHUI UNIVERSITY OF SOUTH
AFRICA--UNISA
%College of Science, Engineering and Technology
% University of South Africa

%clear all;% clear all previous program
clc;
img_size=92*112;for the first data base
img size=180x200 for the second database
ni=40;
np=10;%total image per classes for first class
np=20 %total image per classe for the second class
F= 0:10:ni*np-1;
 nk=7;%number of image per classe allocates for the training
  %the remaining (np-nk) are the testimg images
count=1;
ct=0;
for  np=1:np

for ind_ice =0.85
    sum_image =zeros(img_size,ni*np);

 for i=1:ni)]for j=1:np
    %convert into double precision value im2double
temp=im2double(imread(strcat('C:\Users\MIRABEAU
NGUELA\Desktop\att_faces\s',num2str(i),'\',num2str(j),'.pgm')));% ROOT OF THE
image used for the experiment
```

```
temp=temp(:);%momentary storage place, for treatment
sum_image(:,np*(i-1)+j)=temp;
end
 end
 %------------------------PREPROCESSING STEPS
 %evaluating the common features :Calculating the average of training set
 aver_tr=sum(sum_image,2)/size(sum_image,2);
% image1=aver_tr(:,1);    %normalising the faces vectors
 sum_image_normal=sum_image-aver_tr*ones(1,size(sum_image,2));
 %covariance matrix through dimentionality reduction
  % calculating the eigen-vectors  and eigen-values
 [eig_vect_min,BETA]=eig(Covariance_Mat);%formula to stored the eigen value
 BETA=diag(BETA);
 for col=1:size(eig_vect_min,2)
     eig_vect_min(:,col)=eig_vect_min(:,col)./(sqrt(BETA(col)));
 end
%projecting the eigen-vectors into its original dimensionality
eig_vect =(sum_image_normal*eig_vect_min);
%Set the ‘v’ for the gradient descent
%  the eig vectors with maximum variance are selected
lsum= sum(BETA);
lsum_extract=0;
nf=0;
while (lsum_extract/lsum= ind_ice)
nf=nf+1;
lsum_extract=sum(max_eig(1:nf));% extract the max value
end
max_ind=index(1:nf);% extraction of the max value
%principle components or features extraction of images selected
PCA=eig_vect(:,max_ind).'*sum_image_normal;
end
```

APPENDIX 3. SAMPLE CODE FOR SIMULATION

```
function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
% Generated by Neural Network Toolbox function genFunction, 09-Sep-2019
19:41:54.
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 91xQ matrix, input #1
% and returns:
%   y = 40xQ matrix, output #1
% where Q is the number of samples.
%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====
% Input 1
x1_step1_xoffset = [-36.6648396088447;-31.8912100721877;-28.2510180957323;-.
.
.
.
0.12124095607379393 0.83369204972303823 -0.98589968474649825 -
0.81735456548846841 0.99736976921419263 0.93663294221660065 -
0.55975647965284936 0.20088495065301148 -1.0342141601219734
0.90786595933649428 0.10536583240467876 1.0684891214158907
0.54258972901435121 0.28976947803415803 0.97125638686600047 -
0.71051230315037284 0.95371718440209596 -1.0737011183492637 -
0.84299602742090252 1.0221318499747565];

% ===== SIMULATION ========
% Dimensions
Q = size(x1,2); % samples
% Input 1
xp1 = mapminmax_apply(x1,x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);
```

```
% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);
% Layer 2
a2 = softmax_apply(repmat(b2,1,Q) + LW2_1*a1);
% Output 1
y1 = a2;
end
% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings_gain,settings_xoffset,settings_ymin)
y = bsxfun(@minus,x,settings_xoffset);
y = bsxfun(@times,y,settings_gain);
y = bsxfun(@plus,y,settings_ymin);
end
% Competitive Soft Transfer Function
function a = softmax_apply(n)
nmax = max(n,[],1);
n = bsxfun(@minus,n,nmax);
numer = exp(n);
denom = sum(numer,1);
denom(denom == 0) = 1;
a = bsxfun(@rdivide,numer,denom);
end
% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
a = 2 ./ (1 + exp(-2*n)) - 1;
end
```

APPENDIX 4. SAMPLE CODE FOR TESTING

(same as simulation)

```
function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 12-Sep-2019 06:37:37.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 127xQ matrix, input #1
% and returns:
%   y = 40xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1_xoffset = [-36.6648396088447;-31.8912100721877;-28.2510180957323;-17.3738414452941;-23.3419417154327;-21.5523348532537;-22.4989377149703;-12.0353109504117;-15.2292374915382;-11.7421696444349;-9.64313482779055;-13.7161286572204;-9.69448150187986;-11.2558559014862;-10.4310414293245;-8.43890610271155;-7.87720448377305;-10.0917510566148;-7.68012023737059;-7.17736053717668;-8.16009728601905;-10.736464955725;-8.60553931326045;-9.32661758149192;-7.26362135630781;-7.89805892080765;-5.51978292675836;26.9526521768051;-8.07660368792682;-15.5223047661257;-7.24213980567644;-7.39879188338681;-7.69260054945994;-1.50158056414544;-5.73580001709785;-9.65421054278993;-6.43609771638785;-11.517155473424;-10.7771761668891;-7.96563089847553;-7.00028926876469;-3.57176755123874;-
```

```
1.0220994278829334];

% ===== SIMULATION ========

% Dimensions
Q = size(x1,2); % samples

% Input 1
xp1 = mapminmax_apply(x1,x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = softmax_apply(repmat(b2,1,Q) + LW2_1*a1);

% Output 1
y1 = a2;
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings_gain,settings_xoffset,settings_ymin)
y = bsxfun(@minus,x,settings_xoffset);
y = bsxfun(@times,y,settings_gain);
y = bsxfun(@plus,y,settings_ymin);
end

% Competitive Soft Transfer Function
function a = softmax_apply(n)
```

```
nmax = max(n,[],1);
n = bsxfun(@minus,n,nmax);
numer = exp(n);
denom = sum(numer,1);
denom(denom == 0) = 1;
a = bsxfun(@rdivide,numer,denom);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
a = 2 ./ (1 + exp(-2*n)) - 1;
end
```

APPENDIX 5. FNAL SCRIPT USED TO SOLVE FACE RECOGNITION PROBLEM

```
% Application to solve a Pattern Recognition Problem with a Neural Network

% This script assumes these variables are defined:
%
%   FEAT_TRAIN1 - input data.%from the first appendix
%   OUT_TRAIN1 - target data.%those value are defined

x = FEAT_TRAIN1;
t = OUT_TRAIN1;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. SuiTable   in low memory situations.
trainFcn = 'trainscg';  % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = 15;
net = patternnet(hiddenLayerSize);

net.input.processFcns = {'removeconstantrows','mapminmax'};% For a list of all
processing functions type: help nnprocess

net.output.processFcns = {'removeconstantrows','mapminmax'};% Choose Input and
Output Pre/Post-Processing Functions
```

```
% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'crossentropy';  % Cross-Entropy

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotconfusion', 'plotroc'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
```

```
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network[30],[31],],[32]
view(net)

% Plots
% Uncomment these lines to enable various plots.
%Fig.ure, plotperform(tr);performance plot
%Fig.ure, plottrainstate(tr);training plot
%Fig.ure, ploterrhist(e)
%Fig.ure, plotconfusion(t,y) ; confusion matrix
%Fig.ure, plotroc(t,y)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
```

```
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
% This script assumes these variables are defined:
%
%   FEAT_TRAIN1 - input data.
%   OUT_TRAIN11 - target data.

x = FEAT_TRAIN1;
t = OUT_TRAIN11;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable   in low memory situations.
trainFcn = 'trainscg';  % Scaled conjugate gradient backpropagation.

% Create a Pattern Recognition Network
hiddenLayerSize = 30;
net = patternnet(hiddenLayerSize);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};
```

```
 % Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand';  % Divide data randomly
net.divideMode = 'sample';  % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
 % Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'crossentropy';  % Cross-Entropy
 % Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotconfusion', 'plotroc'};
 % Train the Network
[net,tr] = train(net,x,t);
 % Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)
tind = vec2ind(t);
yind = vec2ind(y);
percentErrors = sum(tind ~= yind)/numel(tind);
 % Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)
```

```
% View the Network
view(net)
 % Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotconfusion(t,y)
%figure, plotroc(t,y)
 % Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)% if false skip this go to the next
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end%end of the total algorithm
```