



**AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
CAIRO-EGYPT**

Electronics and Communications Engineering Department

Face Recognition Enhancement for Security Purposes

A thesis

**Submitted in partial fulfillment for the requirements of the degree of M.Sc in Electrical
Engineering**

Submitted by

Yasmin Nabil Mousa Mansour

**B.Sc Engineering in the Electronics and Communications Eng. Dept.
Modern Academy**

Supervised by

Prof. Dr. Abdelhalim Abdelnabi Zekry

**Professor in the Electronics and Communications Eng. Dept.
Faculty of Engineering - Ain Shams University ASU**

Dr. Nariman Abd-El-Salam

**Assistant Professor in the Electrical and Communication Eng. Dept.
Canadian International College CIC**

2015

Statement

This dissertation is submitted to Ain-Shams University in partial fulfillment for degree of Master of science in Electrical Engineering (Electronics and Communication Engineering).

The work included in this dissertation was carried out by the author at the Electronics and Communications Engineering Department, Ain Shams University.

No part of this dissertation has been submitted for a degree or a qualification at any other university or institute.

Name : Yasmin Nabil Mousa Mansour

Signature:

Date: /8 /2015

Table of Content

Statement	II
Table of Content.....	III
List of Figures	VII
List of Tables	IX
Abstract	X
Acknowledgments.....	XI
Abbreviations.....	XII
List of Symbols.....	XIII
Chapter 1 Introduction	2
1.1 Introduction to Face Recognition Systems	2
1.2 Face Recognition Challenges.....	2
1.3 Biometric Characteristics	3
1.4 Face Recognition Applications	3
1.5 Human Brain and Face Recognition Systems.....	4
1.6 Previous Work and Motivation	5
1.7 Thesis Organization	6
Chapter 2 Face Recognition Techniques Overview.....	8
2.1 Introduction to Face Recognition Techniques	8
2.2 Holistic Matching Approaches	8
2.2.1 Subspace Methods	8
2.2.2 Neural Network	9
2.2.3 Other Techniques	9
2.2.4 Advantages and Disadvantages of Holistic Approaches.....	10
2.3 Feature Based Approaches (Structural Matching Approaches or Geometric Approaches) .	10
2.3.1 Advantages and Disadvantages of Feature Based Approaches.....	11
2.4 Hybrid Approaches	11
Chapter 3 Principal Component Analysis	13
3.1 Introduction	13
3.2 Image Space and Face Space.....	13

3.3	The Variability Reduction Procedure.....	14
3.4	Characteristics of Principal Components	14
3.5	Overview of Eigenface Method.....	15
3.6	The Eigenface Concept.....	15
3.7	Principal Component Analysis Mathematical Model	17
Chapter 4	Artificial Neural Networks.....	23
4.1	Introduction	23
4.2	Problems Not Suited to a Neural Network Solution	23
4.3	Problems Suited to a Neural Network.....	23
4.3.1	Pattern Recognition.....	24
4.3.2	Classification.....	24
4.3.3	Prediction	24
4.3.4	Optimization.....	24
4.4	Biological Inspiration and Human Neuron.....	25
4.4.1	From Human Neurons to Artificial Neurons.....	26
4.5	Electronic Implementation of Artificial Neurons	26
4.6	Transfer Functions (Activation Functions)	28
4.6.1	Sigmoid Activation Function.....	28
4.6.2	Hyperbolic Tangent Activation Function	29
4.6.3	Linear Activation Function.....	29
4.6.4	Threshold Activation Function	30
4.7	General Network Architecture	30
4.7.1	Learning as Determination of Weights.....	31
4.7.2	Learning Paradigms	31
4.7.2.1	Unsupervised Training	31
4.7.2.2	Supervised Training.....	31
4.7.3	Learning Algorithms	32
4.8	Neural Networks Types	32
4.8.1	Feed-forward Neural Networks.....	32
4.8.1.1	Single-Layer Perceptron.....	32
4.8.1.2	Multi-Layer Perceptron.....	33
4.8.1.3	Adaptive Linear Neuron (ADALINE)	34
4.8.1.4	Radial Basis Functions.....	35
4.8.1.5	Kohonen Self-Organizing Maps.....	36
4.8.2	Recurrent Neural Networks.....	36
4.8.3	Stochastic Neural Networks	36
4.8.4	Modular Neural Networks.....	36
Chapter 5	Feed Forward Back Propagation Neural Networks.....	38
5.1	Introduction	38
5.2	Feed Forward Neural Network.....	38
5.2.1	Structure of a Feedforward Neural Network	39

5.2.1.1	Input Layer	39
5.2.1.2	Output Layer	39
5.2.1.3	Hidden Layer	39
5.2.1.3.1	Number of Neurons in the Hidden Layer, Underfitting and Overfitting Problems	40
5.2.2	Preventing Underfitting and Overfitting	40
5.2.3	Selecting Learning Parameters for the Back Propagation Training Algorithm	41
5.2.4	Back Propagation Algorithm.....	42
5.2.5	Back Propagation Algorithm Mathematics.....	43
5.2.5.1	Network Training as a Function Optimization Problem.....	44
5.2.5.2	Computing the Jacobian	45
5.2.5.3	Approximating the Hessian.....	45
5.2.5.4	Solving the Levenberg-Marquardt Equation.....	45
5.2.5.5	General Levenberg-Marquardt Algorithm	46
Chapter 6	<i>Simulation and Results</i>.....	48
6.1	Introduction	48
6.2	The Proposed System Flow Diagram	48
6.2.1	Face Data Base Description	48
6.2.2	Preparing Database	50
6.2.3	Dimensionality Reduction and Feature Extraction using Principal Component Analysis.....	50
6.2.4	Neural Network Design and Architecture	51
6.2.4.1	Levenberg-Marquardt Training Definition.....	51
6.2.4.2	The Choice of the Number of Neurons in Hidden Layer	51
6.2.4.3	Activation Functions for the Hidden and Output Layers	52
6.2.4.4	Back Propagation Weight/Bias Learning Function (Learngdm)	52
6.2.4.5	The Used Learning Rate and Momentum	52
6.2.4.6	The Performance Function	52
6.2.5	Training Neural Network with Levenberg-Marquardt Training.....	52
6.2.5.1	Supervised Learning.....	53
6.2.5.2	Binary Classification Problem Description	53
6.2.5.3	Initializing Weights and Biases.....	55
6.2.5.4	Network Training Window	56
6.2.6	Neural Network Testing.....	57
6.2.7	Network Outputs and Performance Plots for the 80:20 Case	58
6.2.7.1	Network Training Window in 80:20 case.....	58
6.2.7.2	Target, Network Output, and MSE Values and Plots for the Testing Set in 80:20 Case	59
6.2.7.3	Target Values and Plot for all Cases, Network Output, MSE Values and Plots for the whole Dataset in 80:20 Case	62
6.2.7.4	Training Performance Plot in 80:20 Case.....	69
6.2.7.5	Training State Plots in 80:20 Case.....	69
6.2.8	Network Outputs and Performance Plots for the 90:10 Case	70
6.2.8.1	Network Training Window in 90:10 Case	70
6.2.8.2	Network Output, and MSE Plots for the whole Dataset in 90:10 Case.....	71
6.2.8.3	Training Performance Plot in 90:10 Case.....	72
6.2.8.4	Training State Plots in 90:10 Case.....	73
6.2.9	Network Outputs and Performance Plots for the 70:30 Case	73
6.2.9.1	Network Training Window in 70:30 Case	73
6.2.9.2	Network Output, and MSE Plots for the whole Dataset in 70:30 Case.....	74

6.2.9.3	Training Performance Plot in 70:30 Case.....	75
6.2.9.4	Training State Plots in 70:30 Case.....	76
6.2.10	Network Outputs and Performance Plots for the 60:40 Case	76
6.2.10.1	Network Training Window in 60:40 Case`	76
6.2.10.2	Network Output, and MSE Plots for the whole Dataset in 60:40 Case.....	77
6.2.10.3	Training Performance Plot in 60:40 Case.....	78
6.2.10.4	Training State Plots in 60:40 Case.....	79
6.2.11	Network Outputs and Performance Plots for the 50:50 Case	79
6.2.11.1	Network Training Window in 50:50 Case	79
6.2.11.2	Network Output, and MSE Plots for the whole Dataset in 50:50 Case.....	80
6.2.11.3	Training Performance Plot in 50:50 Case.....	81
6.2.11.4	Training State Plots in 50:50 Case.....	82
6.2.12	Simulation Results for all Experiments after Classification	82
Chapter 7	Conclusion and Future Work	86
7.1	Conclusion.....	86
7.2	Future Work	87
7.2.1	Smart Environment	87
7.2.2	Wearable Systems	88
References.....		89
References.....		90

List of Figures

Figure 2.1 a, b Feature extraction based on face regions	10
Figure 2.2 Different face landmarks	10
Figure 3.1 Image space and face cluster approximated by 3D space	13
Figure 3.2 PCA Algorithm	17
Figure 3.3 (a) The average of two faces (b) The average image for the face database	18
Figure 3.4 Eigenfaces generated from the training set	19
Figure 4.1 Neuron cell	25
Figure 4.2 Artificial neuron with respect to a human neuron	26
Figure 4.3 A Detailed schematic of simplistic artificial neuron	27
Figure 4.4 Sigmoid Function	28
Figure 4.5 Hyperbolic tangent function	29
Figure 4.6 Linear function	29
Figure 4.7 Threshold function	30
Figure 4.8 The General neural network architecture	30
Figure 4.9 Single layered perceptron neural network architecture	33
Figure 4.10 Multi-layered Perceptron neural network architecture	34
Figure 4.11 Adaptive Linear Neuron	35
Figure 4.12 Architecture of radial basis function neural network	35
Figure 5.1 Feed forward neural network	39
Figure 5.2 A 3-Dimensional approximation of network error in weight space	41
Figure 5.3 Sloshing problem	42
Figure 5.4 Back Propagation of multilayered ANN	43
Figure 6.1 The proposed face recognition system	48
Figure 6.2 ORL face database	49
Figure 6.3 Supervised Learning	53
Figure 6.4 The used dataset	54
Figure 6.5 Target states values for 8 patterns with 8 images for training neural network	55
Figure 6.6 Network training window example	56
Figure 6.7 Network training window for 80:20 case	59
Figure 6.8 Target values plot for the testing set	61
Figure 6.9 Network output plot for the testing set	61
Figure 6.10 Mean Square error performance values for 16 test images in the 80:20 case	62
Figure 6.11 Target values plot for the whole dataset for all cases	66
Figure 6.12 Network output plot for the whole dataset for the 80:20 case	66
Figure 6.13 Mean square error performance after testing the whole dataset for 80:20 case	68
Figure 6.14 The Network Training Performance, MSE values through 32 epochs for 80:20 case	69
Figure 6.15 Training State plots; gradient and damping factor plots vs. number of epochs for 80:20 case	70
Figure 6.16 Network training window for 90:10 case	71
Figure 6.17 Network output for 90:10 case	71
Figure 6.18 Mean square error performance after testing the whole dataset for 90:10 case	72
Figure 6.19 The Network Training Performance, MSE values through 22 epochs for 90:10 case	72
Figure 6.20 Training State plots; gradient and damping factor plots vs. number of epochs for 90:10 case	73
Figure 6.21 Network training window for 70:30 case	74

<i>Figure 6.22 Network output for 70:30 case.....</i>	<i>74</i>
<i>Figure 6.23 Mean square error performance after testing the whole dataset for 70:30 case</i>	<i>75</i>
<i>Figure 6.24 The Network Training Performance, MSE values through 15 epochs for 70:30 case</i>	<i>75</i>
<i>Figure 6.25 Training State plots; gradient and damping factor plots vs. number of epochs for 70:30 case</i>	<i>76</i>
<i>Figure 6.26 Network training window for 60:40 case</i>	<i>77</i>
<i>Figure 6.27 Network output for 60:40 case.....</i>	<i>77</i>
<i>Figure 6.28 Mean square error performance after testing the whole dataset for 60:40 case</i>	<i>78</i>
<i>Figure 6.29 The Network Training Performance, MSE values through 21 epochs for 60:40 case</i>	<i>78</i>
<i>Figure 6.30 Training State plots; gradient and damping factor plots vs. number of epochs for 60:40 case</i>	<i>79</i>
<i>Figure 6.31 Network training window for 50:50 case</i>	<i>80</i>
<i>Figure 6.32 Network output for 50:50 case.....</i>	<i>80</i>
<i>Figure 6.33 Mean square error performance after testing the whole dataset for 50:50 case</i>	<i>81</i>
<i>Figure 6.34 The Network Training Performance, MSE values through 320 epochs for 50:50 case</i>	<i>81</i>
<i>Figure 6.35 Training State plots; gradient and damping factor plots vs. number of epochs for 50:50 case</i>	<i>82</i>

List of Tables

<i>Table 5-1 Summary of back propagation neural network parameters.....</i>	<i>42</i>
<i>Table 6-1 Criteria's that can be used to stop network training.</i>	<i>57</i>
<i>Table 6-2 Target values of the testing set.....</i>	<i>59</i>
<i>Table 6-3 Network output values for the testing set</i>	<i>60</i>
<i>Table 6-4 Mean square error performance values for the testing set</i>	<i>61</i>
<i>Table 6-5 Target values for the whole data set for all cases</i>	<i>62</i>
<i>Table 6-6 Network output values for the whole data set in the 80:20 case</i>	<i>64</i>
<i>Table 6-7 Mean square error performance values for the whole data set.....</i>	<i>67</i>
<i>Table 6-8 For 90% Training, 10% Testing, Number of training images =72 , Number of testing images=8.....</i>	<i>83</i>
<i>Table 6-9 For 80% Training, 20% Testing, Number of training images =64 , Number of testing images=16.....</i>	<i>83</i>
<i>Table 6-10 For 70% Training, 30% Testing, Number of training images=56, Number of testing images=24.....</i>	<i>83</i>
<i>Table 6-11 For 60% Training, 40% Testing, Number of training images=48, Number of testing images=32</i>	<i>83</i>
<i>Table 6-12 For 50% Training, 50% Testing, Number of training images=40, Number of testing images=40</i>	<i>84</i>

Abstract

With the fast evolution in security fields in ministries and large firms, face recognition systems are needed to identify and recognize people. Given an input face image and a database of face images of known individuals, face recognition systems aims to verify or determine the identity of the person that in the input image.

Face Recognition is a task that human vision system seems to perform almost effortlessly. Yet the goal of building computer-based systems with comparable capabilities has proven to be difficult.

Many techniques for face recognition have been proposed over the last 30 years, and some were even proposed earlier. Some techniques extract local features from the face image (structural matching techniques), while others uses the face image as a whole (holistic matching techniques). Some techniques are a hybrid of both.

Subspace techniques which belong to the holistic category are the most successful techniques for face recognition. These techniques aim to reduce the dimensions of the face images by projecting the high-dimensional images to a lower-dimensional subspace, where each image is represented by a linear combination of vectors in the subspace.

In this work, a fully face recognition system to classify faces has been implemented. Principal Component Analysis (PCA) technique has been used as one of the holistic matching techniques for dimensionality reduction and feature extraction. Using PCA, features are extracted from the whole face, then an empirical method to choose the most principal “important” features from the face has been developed, this empirical method helps the classifier to recognize faces more easily and more efficiently. The mathematical model of PCA technique is proposed in this thesis, and its equations has been performed and implemented using Matlab Programming.

After the feature extraction phase, a Feed Forward Neural Network is used for classification, specifically, a Multi Linear Perceptron (MLP) networks is used, and a Back Propagation (BP) algorithm is used as a learning algorithm.

An empirical method has been developed to choose the number of hidden neurons of the hidden layer to avoid both underfitting and overfitting problems. The mathematical model of the back propagation algorithm is proposed in this thesis, and has been performed using the neural network tool box available in Matlab. The Olivetti Research Laboratory (ORL) database is used for performance evaluation.

Acknowledgments

I would like to praise Allah for all his graces and for helping me finishing this thesis before the end of my registration period.

I gratefully acknowledge my indebtedness to Prof. Dr. Abdel-Halim Abdel-Nabi Zekry who initiated and supervised this work. I would like to thank him for his guidance and support throughout the work. He conducted a lot of fruitful discussion and has been a source of inspiration and encouragement. Special thanks to Dr. Nariman Abdel-Salam for her support and encouragement to complete this work in an appropriate way. Special thanks to Dr. Ahmed Salah Eldin Mohammed for his support and continues generous help and advices.

Finally, these acknowledgements would be incomplete without expressing my deep love and gratitude to my parents and my sister for their patience and encouragement and providing me a special environment to finish it.

Yasmin Nabil

Abbreviations

ADALINE	Adaptive Linear Neuron
ANN	Artificial Neural Network
BP	Back Propagation
BPNN	Back Propagation Neural Network
CNN	Convolutional Neural Network
DCT	Discrete Cosine Transform
DLA	Dynamic Link Architecture
E	Eigen Space
EBGM	Elastic Bunch Graph Matching
FLD	Fisher Discriminant Analysis
HCI	Human-computer interaction
ICA	Independent Component Analysis
IR	Infra Red
KLT	Karhunen-Loeve Transform
Learngdm	Back propagation weight/bias learning function
LDA	Linear discriminant Analysis
LM	Levenberg Marquardt Algorithm
MLP	Multi Layer Perceptron
MSE	Mean Square Error
NN	Neural Network
Newff	Network design function
ORL	Olivetti Research Laboratory
PCA	Principal Component Analysis
R	A value to indicate the relationship between output and target
RBF	Radial Basis Function
RGB	Red-Green-Blue Model
RNG	Random Number Generator
RNN	Recurrent Neural Network
SLP	Single Layer Perceptron
SOM	Self Organizing Map
SVM	Support Vector Machine
Trainlm	Levenberg Marquardt Training Algorithm
TSP	Traveling Sales Man Problem

List of Symbols

$I(x,y)$	Image of two dimensional m by n array intensity values
m	Number of rows in I image
n	Number of columns in I image
R^N	The $m*n$ dimensional space
M	Number of training images
N	Number of pixels contained in an image
Γ_M	M sample images contained in the database (the training set of images)
Ψ	Mean image
Φ_i	Mean centered face training image in the image space
A	Data Matrix.
C	Covariance matrix AA^T
L	Matrix $A^T A$
μ_i	Eigen values
v_i	Eigen vectors
U	Eigenvectors of the covariance matrix AA^T
e_k	The selected eigenfaces (the matrix of the first eigenfaces)

v_k	The largest selected eigenvectors
e_k	The selected eigenfaces (the matrix of the first eigenfaces)
y_i	The projection of the centered training image Φ_i to the PCA <i>eigenspace</i> E defined by the reduced matrix of eigenvectors of the covariance matrix C
T	New face image of an individual
t	The mean centered image of the new test image of an individual
F_k	The projection of the vector of the mean testing face image T into the eigenspace
Ω^T	Vector of the calculated values of F together which describes the contribution of each eigenface in representing the input face image, treating the eigenfaces as a basis set for face images
X_k	Vector of current weights and biases in the back propagation algorithm
g_k	Current gradient in the back propagation algorithm
α_k	Learning rate in the back propagation algorithm
J	Jacobian matrix for the system that contains first derivatives of the network errors with respect to weights and biases
λ	Levenberg's damping factor
δ	Weight update vector that we want to find
E	Error vector containing the output errors for each input vector used on training the network
H	Approximated Hessian matrix $J^T J$
$F(x_i, w)$	Network function evaluated for the i -th input vector of the training set using the weight vector w in Jacobian matrix

w_j	The j -th element of the weight vector w of the network in the Jacobian matrix.
g	Error gradient in the back propagation algorithm
S_{nj}	Target state of n patterns, and j images included in each n pattern