

# Face Recognition System Based on Different Artificial Neural Networks Models and Training Algorithms

Omaima N. A. AL-Allaf

Assistant Professor,  
Dept. of CIS, Faculty of Sciences &  
IT, Al-Zaytoonah University of  
Jordan, P.O. Box130, Amman  
(11733), Jordan

Abdelfatah Aref Tamimi

Associate Professor,  
Dept. of CS, Faculty of Sciences &  
IT, Al-Zaytoonah University of  
Jordan, P.O. Box130, Amman  
(11733), Jordan

Mohammad A. Alia

Assistant Professor,  
Dept. of CIS, Faculty of Sciences &  
IT, Al-Zaytoonah University of  
Jordan, P.O. Box130, Amman  
(11733), Jordan

**Abstract**— Face recognition is one of the biometric methods that is used to identify any given face image using the main features of this face. In this research, a face recognition system was suggested based on four Artificial Neural Network (ANN) models separately: feed forward backpropagation neural network (FFBPNN), cascade forward backpropagation neural network (CFBPNN), function fitting neural network (FitNet) and pattern recognition neural network (PatternNet). Each model was constructed separately with 7 layers (input layer, 5 hidden layers each with 15 hidden units and output layer). Six ANN training algorithms (TRAINLM, TRAINBFG, TRAINBR, TRAINCGF, TRAINGD, and TRAINGD) were used to train each model separately. Many experiments were conducted for each one of the four models based on 6 different training algorithms. The performance results of these models were compared according to mean square error and recognition rate to identify the best ANN model. The results showed that the PatternNet model was the best model used. Finally, comparisons between the used training algorithms were performed. Comparison results showed that TrainLM was the best training algorithm for the face recognition system.

**Keywords**—Face Recognition; Backpropagation Neural Network (BPNN); Feed Forward Neural Network; Cascade Forward; Function Fitting; Pattern Recognition

## I. INTRODUCTION

Human Face represents complex, multidimensional, meaningful visual motivation. It is difficult to develop a computational model for face recognition. Building good computer system similar to human ability to recognize faces and overcome humans' limitations is regarded as a great challenge [1]. The human ability to recognize faces has several difficulties such as: similarity between different faces; dealing with large amount of unknown human faces; expressions and hair can change the face; and also face can be viewed from number of angles in many situations. A good face recognition system must be robust to overcome these difficulties and generalize over many conditions to capture the essential similarities for a given human face [2]. A general face recognition system consists of many processing stages: face detection; facial feature extraction; and face recognition. Face detection and feature extraction phases could run simultaneously [3].

In the recent years, artificial neural networks (ANN) were used largely for building intelligent computer systems related to pattern recognition and image processing [4]. The most popular ANN model is the backpropagation neural network (BPNN) which can be trained using backpropagation training algorithm (BP) [5]. Many literatures related to face recognition system which based on different approaches such as: Geometrical features; Eigenfaces; Template matching; Graph matching; and ANN approaches [6]. The obtained recognition rates from these studies are different and based on: used approach; used database; and number of classes.

Different ANN models were used widely in face recognition and many times they used in combination with the above mentioned methods. ANN simulates the way neurons work in the human brain. This is the main reason for its role in face recognition. Many researches adopted different ANN models for face recognition with different recognition rates and mean square error (MSE). Therefore, there is a need to identify the ANN model for face recognition systems with best recognition results. The objective of this research is to develop a face recognition system based on using 4 different ANN models: feed forward Backpropagation neural network (FFBPNN), cascade forward Backpropagation neural network (CFBPNN), function fitting (FitNet), and pattern recognition (PatternNet). Each one of these models was constructed separately with 7 layers (input, 5 hidden layers and output layer) architectures. Each model was trained separately with six different training algorithms.

The research includes the following sections: Section II includes related literature; Section III includes details about ANN architectures and training algorithms; Section IV explains research methodology; Section V includes implementation steps of the face recognition system; Section VI includes the experimental results; and finally Section VII concludes this work.

## II. RELATED LITERATURE

ANN has the ability to adjust its weights according to the differences it encounters during training [7]. Therefore, we focused in this research on literature studies which based on ANN models especially BPNN. Dmitry and Valery (2002) [8]

proposed ANN thresholding approach for rejection of unauthorized persons. They studied robustness of ANN classifiers with respect to false acceptance and false rejection errors.

Soon and Seiichi (2003) [9] presented face recognition system with incremental learning ability that has one-pass incremental learning and automatic generation of training data. They adopted Resource Allocating Network with Long-Term Memory (RANLTM) as a classifier of face images. Adjoudj and Boukelif (2004) [10] designed a face recognition system using ANN which can trained several times on various faces images. While Volkan (2003) [11] developed a face authentication system based on: preprocessing, principal component analysis (PCA), and ANN for recognition. Normalization illumination, and head orientation were done in preprocessing stage. PCA is applied to find the aspects of face which are important for identification.

Weihua and WeiFu (2008) [12] suggested a face recognition algorithm based on gray-scale. They applied ANN to the pattern recognition phase rather than to the feature extraction phase to reduce complexity of ANN training. Also Mohamed, et. al. (2006) [13] developed BPNN model to extract the basic face of the human face images. The eigenfaces is then projecting onto human faces to identify unique features vectors. This BPNN uses the significant features vector to identify unknown face. They used ORL database. While Latha et al (2009) [14] used BPNN for face recognition to detect frontal views of faces. The PCA is used to reduce the dimensionality of face image. They used Yale database and calculated acceptance ratio and execution time as a performance metrics.

Raman and Durgesh (2009) [15] used single layer feed forward ANN approach with PCA to find the optimum learning rate that reduces the training time. They used variable learning rate and demonstrate its superiority over constant learning rate. They test the system's performance in terms of recognition rate and training time. They used ORL database. Abul Kashem et. al (2011) [16] proposed a face recognition system using PCA with BPNN. The system consists of three steps: detecting face image using BPNN; extraction of various facial features; and performing face recognition. And Shatha (2011) [17] performed face recognition by 3D facial recognition system using geometrics techniques with two types of ANN (multilayer perceptron and probabilistic). At the end, Taranpreet (2012) [18] proposed face recognition method using PCA with BP algorithm. The feature is extracted using PCA and the BPNN is used for classification.

### III. ARTIFICIAL NEURAL NETWORKS

FFBPNN consists of many layers as in BPNN. The first layer is connected to ANN inputs. Each subsequent layer has connections from preceding layer. The final layer produces ANN output. BPNN and FFBPNN can be trained using BP algorithm. The BP includes the following equations [19][20]:

$$U_k(t) = \sum_{j=1}^n w_{jk}(t) \cdot x_j(t) + b_{ok}(t) \quad (1)$$

$$Y_k(t) = \varphi(U_k(t)) \quad (2)$$

Where,

- $x_j(t)$ : input value of j at time-step t,
- $w_{jk}(t)$ : weight assigned by neuron k to input j at time t,
- $\varphi$ : nonlinear activation function,
- $b_k(t)$ : the bias of k-neuron at time t, and
- $y_k(t)$ : output from neuron k at time t.

The process is repeated for all entries of time series and yields an output vector  $y_k$ . The training process includes weight adjustments to minimize the error between network's desired and actual output using an iterative procedure. Output  $y_k$  is compared with target output  $T_k$  using Eq.3 as an error function:

$$\delta_k = (T_k - y_k) y_k (1 - y_k) \quad (3)$$

The error is given by Eq.4 for neurons in the hidden layer:

$$\delta_k = y_k (1 - y_k) \sum \delta_k w_k \quad (4)$$

Where  $\delta_k$  is the error term of the output layer and  $w_k$  is the weight between the hidden and output layers. The error is then propagated backward from the output layer to input layer to update the weight of each connection as follows [20]:

$$w_{jk}(t+1) = w_{jk}(t) + \eta \delta_k y_k + \alpha (w_{jk}(t) - w_{jk}(t-1)) \quad (5)$$

Where,  $\eta$  is the learning rate, and  $\alpha$  is a momentum variable, which determines the effect of past weight changes on the current direction of movement.

Another ANN is CFBPNN and it is similar to FFBPNN but it includes a connection from input and every previous layer to following layers. Additional connections can improve the speed at which ANN learns the desired relationship. FitNet also presents a type of FFBPNN, which is used to fit an input output relationship. While PatternNet is a feed forward network that can be used for pattern recognition problems and can be trained to classify inputs according to target classes. The target data for PatternNet consist of vectors with all values equal to 0 except for 1 in element i, where i is the class they represent.

The BP training algorithm used to train FFBPNN and other ANN models requires long time to converge. Therefore, many optimization training algorithms were suggested and described in details in Neural Network Toolbox™ User's Guide R2012a [21]. The equations of all algorithms are the same except they differs in changing weight values.

#### A. Learning Algorithms

The optimization training algorithms adjusted the ANN weights and biases to minimize the performance function and to reduce errors as possible. Here, mean square error (MSE) is used as a performance function of the suggested face recognition system and it is minimized during ANN training. MSE represents the difference between the desired output and actual output.

In this research, six optimization ANN training algorithms were used to train the Four models separately to identify the model with the best results for the face recognition system [20][21]:

- Levenberg-Marquardt algorithm (TRAINLM)
- TRAINBFG algorithm
- Bayesian regularization algorithm (TRAINBR)
- TRAINCGF algorithm
- Gradient descent algorithm (TRAINGD)
- Gradient descent with momentum (TRAINGDM)

#### IV. RESEARCH METHODOLOGY

In this research, 4 ANN models (FFBPNN, CFBPNN, FitNet and PatternNet) were used separately for the face recognition system. Each one of these models was constructed separately with 7 layers (input, 5 hidden layers and output layer). Fig.1 shows 7-layer FFBPNN with 15 neurons in each hidden layer. Fig.1 can be used also to describe FitNet and PatternNet separately. The only difference in these ANN models is in training functions. Fig. 2 shows the 7 layers CFBPNN with 15 neurons in each hidden layer.

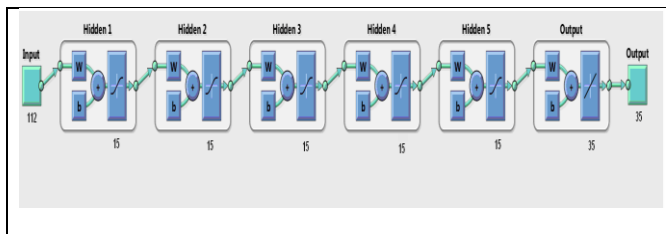


Fig. 1. Feed Forward Backpropagation Neural Network with 7 layers.

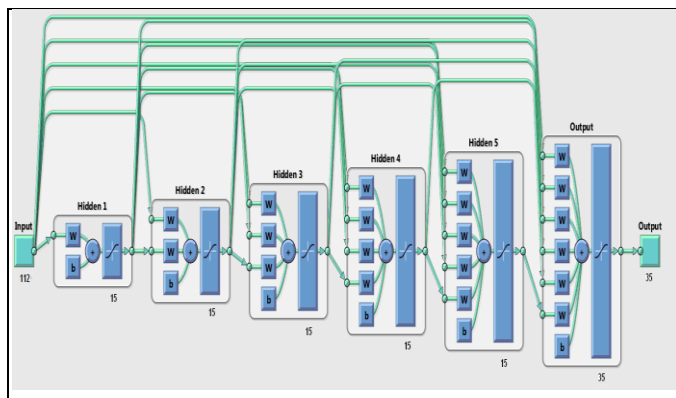


Fig. 2. Cascaded Feed Backpropagation Neural Network with 7 layers.

#### A. Training/Testing Samples

In this research, the training and testing samples were taken from the Oral face database (Olivetti Research

Laboratory) [22]. This database contains a set of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, UK. There are 10 different images of 40 distinct persons. For each person, the images were taken at different times, varying lighting slightly, facial expressions (open/closed eyes, smiling/non-smiling) and facial details (glasses/no-glasses). All the images are taken against a dark homogeneous background and the subjects are in up-right, frontal position (with tolerance for some side movement). All images are stored in ORL in PGM format with resolution  $92 \times 112$ , 8-bit grey levels. Fig.3 shows samples from Oral face database for 6 persons.

As training samples, 350 face images (each with  $92 \times 112$  dimension) were taken for 35 persons each with 10 samples. Each one of these images ( $92 \times 112 = 10304$ ) is normalized and segmented into many blocks each with dimensions  $8 \times 14 = 112$ . This segmentation ( $(92 \times 112) / (8 \times 14)$ ) will result in 92 sub images (blocks) for each face image. Each one of these samples (block) is with size 112. Therefore the number of input layer units is 112.

Whereas the number of output layer units is 35 to recognize these 35 persons. At the same time, the total number of training samples = number of images used in training process (350) multiplied by number of sub images (blocks) for each image (92) and this is equal to 32200. These samples are used in the face recognition system training process.

As testing samples, firstly, we select 50 random images from training samples for 5 persons each with 10 samples. Secondly, 50 ( $92 \times 112$ ) images were selected from Oral face database (which are not used in training) for 5 persons each with 10 samples. Each one of the 50 selected images ( $92 \times 112$ ) is divided into blocks of dimension  $8 \times 14$  to obtain 92 blocks for each image. Therefore, the total number of testing samples for the 50 randomly selected face images is equal to 4600.

#### B. ANN Architecture

Fig.4 shows the architecture of the suggested ANN model for face recognition system and it consists of 7 layers (input layer, 5 hidden layers each with 15 hidden units and finally output layer). The input layer represents the face sub image (block) as system input. The number of input layer neurons depends on sub image dimensions ( $8 \times 14$ ) and here it is equal to 112.

Finally, the output layer returns the output vector. The number of output layer neurons depends on the problem nature and here it depends on the number of classes used in the face recognition training process. Since 350 images of 35 different persons were adopted, the number of classes is equal to 35 and hence, this is the number of output layer neurons.

class no.	Images belong to each class which related to one person
1	
2	
6	
7	
18	
22	

Fig. 3. Samples from Oral face database for 8 persons.

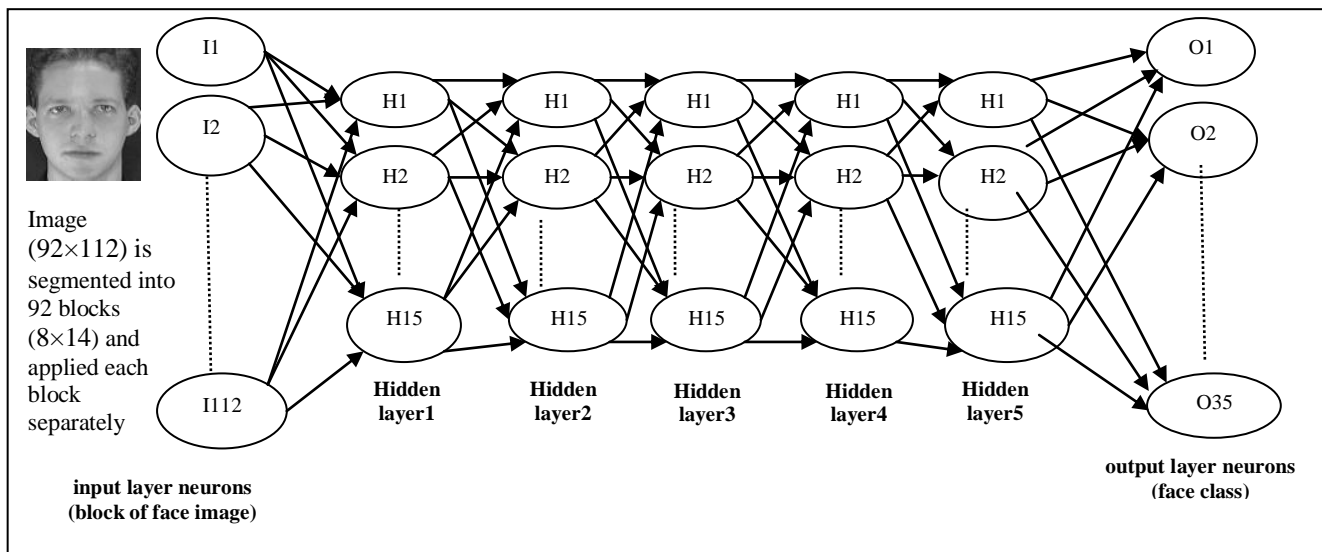


Fig. 4. Feed Forward BackPropagation Neural Network with 7Layers.

## V. IMPLEMENTATION OF FACE RECOGNITION

A MathLab used to write an ANN training and testing face recognition system. This section includes the main steps of training and testing process.

### A. Steps of ANN Training

Steps required to train the ANN model for face recognition system are as follows:

- 1) Initialize the ANN model weights and bias unit.
- 2) Initialize learning rate, momentum variable and threshold error with very small value like 0.0000001.

3) Initialize 35 classes: class for each person. Each class containing 10 faces images of one person.

4) Classification process: Initialize 35 target vectors one vector for each face class: vector = t1, t2... t35. All bits of vector1 are 0 except the first bit is 1. All bits of vector2 are 0 except the second bit is 1, and so on for other vectors.

5) Initialize the target output vector for each input vector (block 8×14) of the 350 face images. As example, the target output for face image blocks related to the 10 face images of the first person is equal to vector1. And, the target output for face image blocks related to the 10 face images of second

person is equal to vector2. And so on for each one of the 10 face images related to remaining 33 persons.

6) Apply steps of the selected training algorithm (LM, BFG, BR, CGF, GD, GDM) to train the ANN model. Apply input vector; compute outputs of each layer to find the actual output vector. Calculate the ANN error and according to this error the training is stopped or repeated again by adjusting the ANN weights. These operations repeated until we get ANN total error equal to threshold error to stop training process. Fig.5 shows these steps.

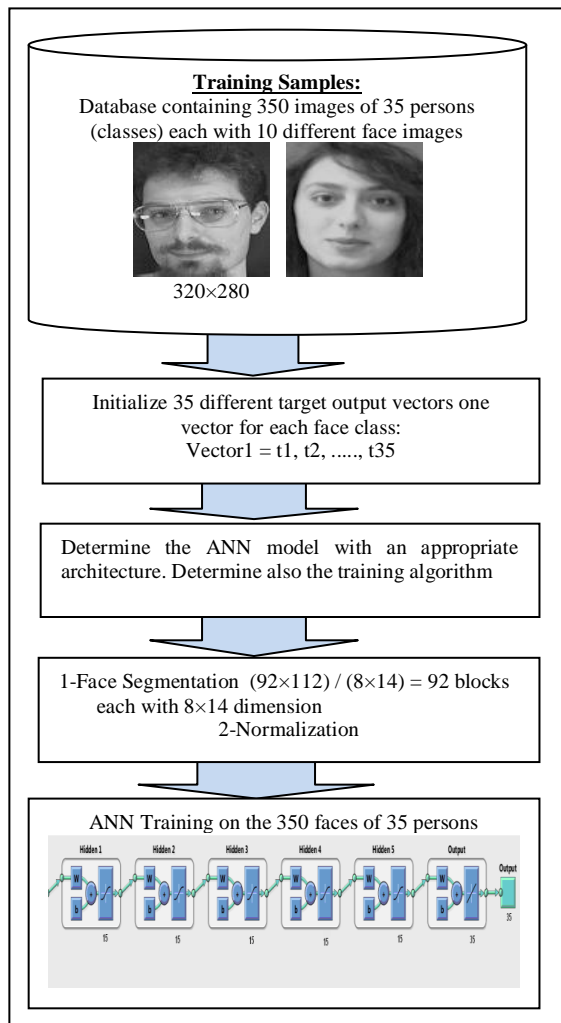


Fig. 5. ANN Training for the Face Recognition System

### B. Steps of ANN Testing

The steps required to test the ANN model for face recognition system are as follows:

- 1) Apply one face block  $8 \times 14$  to input layer neurons.
- 2) Compute the output of all layers in the ANN according to the steps required by training algorithm which was used in ANN training process until finding the outputs of output layer neurons.
- 3) Check if output of output layer neurons (output vector) is the same as one of the 35 classes (it's computed MSE is too

small), then ANN is recognized the block. And if the computed MSE of the ANN output is large, then the ANN is not recognized this block. Fig.6 shows these steps.

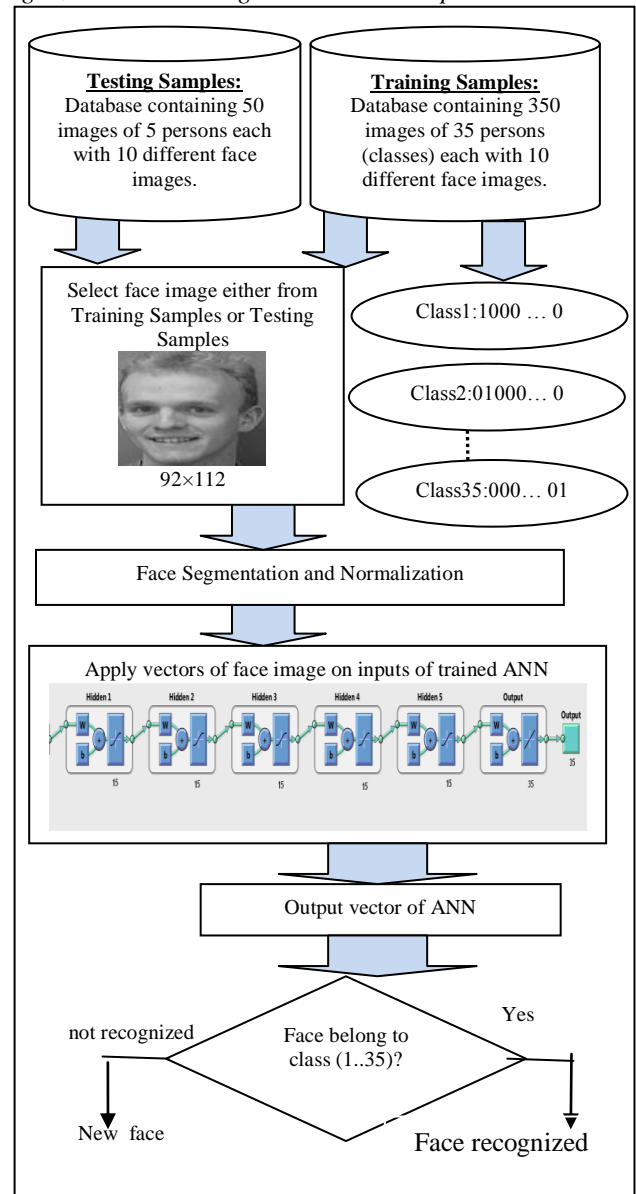


Fig. 6. ANN Testing for the Face Recognition System

## VI. EXPERIMENTAL RESULTS

A MathLab was used to write the simulation program of training/testing of each one of the Four models (FFBPNN, CFBPNN, FitNet and PatternNet). The architecture of each model consists of 7 layers: input; 5 hidden layers each with 15 units; and output layer. The training data includes 350 ( $92 \times 112$ ) face images for 35 persons each with 10 samples were selected from Oral face database (Olivetti Research Laboratory) [22]. Here, we used the Mean Square Error (MSE), peak signal to noise ratio (PSNR) and recognition rate (RR) to evaluate the performance of ANN model for face recognition system.

Many experiments were conducted to examine the ANN model with best results of training and testing processes for the face recognition system. Many experiments were based on adopting different number of hidden layers (2, 3, 5, 7 and 9). Other experiments were based on adopting different numbers of neurons in each hidden layer (5, 10, 15, 20, 25 and 30). The best results were obtained from using 5 hidden layers each with 15 hidden units because we noticed from experiments that increasing number of hidden layers and number of hidden units will result in increasing the training time.

### A. Results of Training Process

To determine the performance of each one of the 4 models, experiments were conducted by training these models separately each with 6 training algorithms. TABLE I shows MSE values of the 4 models.

TABLE I. IMPACT OF ANN MODELS ON MSE

Algorithm	FFBP	CFBP	FitNet	PatternNet
TRAINLM	0.08	0.09	0.03	0.003
TRAINBFG	0.20	0.15	0.09	0.03
TRAINBR	0.10	0.16	0.1	0.05
TRAINCGF	0.14	0.15	0.09	0.04
TRAINGD	0.25	0.27	0.18	0.09
TRAINGDM	0.34	0.31	0.22	0.11

From TABLE I, we noted that the lower values of MSE are obtained for these models when LM training algorithm was used and these values were ranged between 0.003 and 0.09. Also the lowest MSE values were obtained from the PatternNet model. Also, we calculated the number of iterations needed for training process for each experiment. The ANN model required more number of iterations when we increased the number of hidden layer neurons. Therefore we used only 15 hidden units in each hidden layer for each model. TABLE II shows the number of iterations required to train the 4 models with 6 algorithms.

TABLE II. IMPACT OF ALGORITHMS ON NUMBER OF ITERATIONS

Algorithm	FFBP	CFBP	FitNet	PatternNet
TRAINLM	65	54	31	21
TRAINBFG	67	59	35	28
TRAINBR	58	71	41	35
TRAINCGF	72	74	58	41
TRAINGD	66	61	36	31
TRAINGDM	73	62	38	35

From TABLE II, PatternNet required lowest number of iterations (21) for the training process especially when it was trained using TRAINLM algorithm. But PatternNet required 41 iterations when it was trained using TRAINCGF algorithm. Also, TABLE II shows that the other ANN models require more iterations when they were trained using TRAINLM training algorithm.

### B. Results of ANN Testing Process

In testing process, we mentioned earlier in sub section A in section IV that the number of samples used in testing process is 50 images for 5 persons each with 10 different samples. Also sub section B in section V includes the main steps of testing process. The testing process includes two parts. Firstly, the 4 ANN models were tested using 50 face images which randomly selected from the training samples (samples which were used earlier in training process). The lowest values of MSE were obtained from PatternNet model as shown in TABLE III. Also TABLE III shows that the lowest values of all models were obtained from using TRAINLM training algorithm.

TABLE III. MSE FOR TESTING 50 TRAINED IMAGES

Algorithm	FFBP	CFBP	FitNet	PatternNet
TRAINLM	0.07	0.07	0.04	0.004
TRAINBFG	0.19	0.17	0.11	0.05
TRAINBR	0.13	0.18	0.12	0.07
TRAINCGF	0.18	0.17	0.13	0.06
TRAINGD	0.21	0.22	0.16	0.11
TRAINGDM	0.36	0.3	0.2	0.1

TABLE IV shows the recognition rates related to testing the 4 models using 50 randomly selected trained images. Best values of recognition rate were obtained from PatternNet model trained using TRAINLM algorithm.

TABLE IV. RECOGNITION RATE FOR TESTING 50 TRAINED IMAGES

Algorithm	FFBP	CFBP	FitNet	PatternNet
TRAINLM	97.2	97.7	98.1	98.5
TRAINBFG	96.1	96.7	96.9	97
TRAINBR	96	96.1	96.5	96.8
TRAINCGF	95	95.5	95.8	96
TRAINGD	95.2	95.6	95.9	96.5
TRAINGDM	95.6	95.9	96.1	96.7

Also TABLE V shows the PSNR of the testing process related to the trained 5 models using 50 randomly selected trained images. Best values of PSNR were obtained from PatternNet model. At the same time, TRAINLM algorithm results in best values of recognition rates for all models.

TABLE V. PSNR FOR TESTING 50 TRAINED IMAGES

Algorithm	FFBP	CFBP	FitNet	PatternNet
TRAINLM	33.2	33.5	33.7	34.1
TRAINBFG	32.5	32.7	33	33.3
TRAINBR	32.1	32.4	32.7	33
TRAINCGF	31.5	31.8	32.2	32.6
TRAINGD	31.7	31.9	32	32.8
TRAINGDM	31.8	32.3	32.5	32.9

Secondly, the 4 ANN models were tested with 50 testing untrained samples (i.e. images which were not used in training

process). The MSE values obtained from this testing were very high because the 4 ANN models were not recognized these testing images.

The lowest MSE values were obtained from using PatternNet model which was trained using TRAINLM algorithm as shown in TABLE VI.

TABLE VI. MSE FOR TESTING 50 UN-TRAINED IMAGES

Algorithm	FFBP	CFBP	FitNet	Pattern Net
TRAINLM	0.88	0.85	0.76	0.61
TRAINBFG	0.87	0.81	0.75	0.72
TRAINBR	0.8	0.81	0.72	0.69
TRAINCGF	0.87	0.85	0.89	0.9
TRAIINGD	0.84	0.87	0.91	0.94
TRAIINGDM	0.89	0.91	0.95	0.96

Finally, TABLE VII shows the recognition rate for the testing process of the 4 models on untrained images. Therefore the values of recognition rates in TABLE VII are not high.

TABLE VII. RECOGNITION RATE FOR TESTING 50 UNTRAINED IMAGES

Algorithm	FFBP	CFBP	FitNet	PatternNet
TRAINLM	22.1	22.4	22.44	22.7
TRAINBFG	23.4	23.2	23.8	24
TRAINBR	21.5	21.6	21.8	22.2
TRAINCGF	19.2	19.6	19.3	19.7
TRAIINGD	19.5	19.7	19.8	20.2
TRAIINGDM	20.4	20.6	20.7	21.2

## VII. CONCLUSION

In this research, we presented a face recognition system using Four feed forward ANN models (FFBPNN, CFBPNN, FitNet and PatternNet) and 6 training methods. Each one of the 4 models was constructed with 7-layer architecture. This face recognition system consists of two parts: training and testing. Six ANN optimization training algorithms (TRAINLM, TRAINBFG, TRAINBR, TRAINCGF, TRAIINGD, and TRAIINGD) were used to train each of the constructed ANN models separately.

The training and testing samples of the suggested face recognition system were taken from The ORL Database of Faces [22]. As training samples, we selected 350 face images (92×112) from ORL database which belong to 35 persons each with 10 different samples. As testing samples (untrained images), we selected 50 images (92×112) from ORL database which belong to 5 persons each with 10 different samples.

A set of experiments were conducted to evaluate the performance of the suggested face recognition system by calculating the MSE, number of iterations, recognition rate and PSNR. This was done using 4 different ANN models and 6 different optimization algorithms. The results showed that the lowest values of MSE and number of iterations were resulted from the PatternNet model. The best results of the PatternNet model were obtained when this model was trained

using the Levenberg Marquardt training algorithm (TRAINLM).

Future work may include a survey of other techniques related to face recognition systems and comparing their results with those presented in this paper. These comparisons will be based on many factors like: recognition rate, PSNR, algorithm complexity, ANN learning time and number of iterations required for training and so on.

## ACKNOWLEDGEMENT

The authors would like to thank Al-Zaytoonah University of Jordan, Amman, Jordan, for sponsoring the scientific research.

## REFERENCES

- [1] Turk M. and Pentland A., 1991. "Eigenfaces for recognition", *Journal of Cognitive Neuroscience*, vol.3, pp.71–86.
- [2] Jonathan Howell A. and Hilary B., 1996. "Face Recognition using Radial Basis Function Neural Networks", *Proceedings of British Machine Vision Conference*.
- [3] Zhao W., et al, 2003. "Face recognition: A literature survey", *ACM Computing Surveys*, Vol.35, No.4, December, pp. 399–458.
- [4] Dan W. Patterson, 1996. "Artificial Neural Networks, Theory and Applications", *Singapore: Prentice Hall*.
- [5] Hamid B. and Mohamad R. M., 2009. "A learning automata-based algorithm for determination of the number of hidden units for three-layer neural networks", *International Journal of Systems Science*, vol.40, no.1, 101–118, Jan.
- [6] Steve Lawrence, et. al., 1997. "Face Recognition: A Convolutional Neural Network Approach", *IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition*, vol.8, no.1, pp.98–113.
- [7] Hossein S., Mahdi R., and Hamid D., 2008. "Face Recognition Using Morphological Shared-weight Neural Networks", *World Academy of Science, Engineering and Technology*, vol.45, pp.555- 558.
- [8] Dmitry B. and Valery S., 2002. "Access Control By Face Recognition Using Neural Networks And Negative Examples", *The 2nd International Conference on Artificial Intelligence*, pp.428-436, 16-20Sep, Crimea, Ukraine.
- [9] Soon Lee Toh and Seiichi O., 2003. "A Face Recognition System Using Neural Networks with Incremental Learning Ability", *Proceeding of the 8th Australian and New Zealand Conf. on Intelligent Information Systems*, pp.389-394.
- [10] Adjoudj R. and Boukelif A., 2004. "Artificial Neural Network-Based Face Recognition", *First International Symposium on Control, Communications and Signal Processing*, pp.439-442.
- [11] Volkan A., 2003. "Face Recognition Using Eigenfaces and Neural Networks", *Master of Science Thesis*, The Graduate School of Natural And Applied Sciences, The Middle East Technical University, Dec.
- [12] Weihua W. and WeiFu W., 2008. "A Gray-Scale Face Recognition Approach", *Second International Symposium on Intelligent Information Technology Application*, 978-0-7695-3497-8/08, IEEE computer society, DOI 10.1109/IITA.2008.101.
- [13] Mohamed R., et. al. 2006. "Face Recognition using Eigenfaces and Neural Networks", *American Journal of Applied Sciences*, vol.2, no.6, pp.1872-1875, ISSN 1546-9239, Science Publications.
- [14] Latha P., Ganesan L.& Annadurai S., 2009. "Face Recognition using Neural Networks", *Signal Processing: An International Journal*, vol.3, issue.5, pp.153–160.
- [15] Raman B. and Durgesh Kumar M., 2009. "Face Recognition Using Principal Component Analysis and Neural Network", *Journal of Technology and Engineering Sciences*, Vol.1, No.2.
- [16] Mohammad A. K., et, al. 2011. "Face Recognition System Based on Principal Component Analysis (PCA) with Back Propagation Neural Networks (BPNN)", *Canadian Journal on Image Processing and Computer Vision*, vol.2, no.4, April.
- [17] Shatha K. Jawad, 2011. "Design a Facial Recognition System Using Multilayer Perceptron and Probabilistic Neural Networks Based Geometrics 3D Facial", *European Journal of Scientific Research*,

- ISSN:1450-216X, vol.60, no.1, pp.95-104, Euro Journals Publishing, Inc. 2011, <http://www.eurojournals.com/ejsr.htm>
- [18] [18] Taranpreet S. R., 2012. "Face Recognition Based on PCA Algorithm", *Special Issue of International Journal of Computer Science & Informatics (IJCSI)*, ISSN:2231-5292, vol.2, No1.2, pp.221- 225.
- [19] Haykin S., *Neural Networks: A Comprehensive Foundation* 2nd edition, Prentice-Hall Inc., 1999. <http://www.statsoft.com/textbook/stneunet.html>
- [20] H. K. Elminir, Y.A. Azzam, and F. I. Younes, "Prediction of Hourly and Daily Diffuse Fraction Using Neural Network as Compared to Linear Regression Models," *Energy*, vol.32, pp.1513-1523. 2007.
- [21] Mark Hudson Beal, Martin T. Hagan and Howard B. Demuth, *Neural Network Toolbox™ User's Guide R2012a*, The MathWorks, Inc., 3 Apple Hill Drive Natick, MA 01760-2098, 2012, [www.mathworks.com](http://www.mathworks.com)
- [22] AT&T (ORL): Oral face database (Olivetti Research Laboratory), <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.htm>