Volume 2 No. 10, October 2012 ISSN 2223-4985

International Journal of Information and Communication Technology Research

http://www.esjournals.org

Artificial Neural Networks for Iris Recognition System: Comparisons between Different Models, Architectures and Algorithms

[1]Omaima N. Ahmad AL-Allaf, [2]Abdelfatah Aref Tamimi, [3]Shahlla A. AbdAlKader

[1]Dept. of CIS, Faculty of Sciences and IT, Al-Zaytoonah University of Jordan, P.O. Box130, Amman (11733), Jordan

[2]Dept. of MMS, Faculty of Sciences and IT, Al-Zaytoonah University of Jordan, P.O. Box130, Amman (11733), Jordan

[3]Dept. of Computer Systems, Foundation of Technical Education, Technical Institute, Mosul, Iraq

ABSTRACT

In this research, an iris recognition system was suggested based on five Artificial Neural Network (ANN) models separately: feed forward (FFBPNN), cascade forward (CFBPNN), function fitting (FitNet), pattern recognition (PatternNet) and learning vector quantization (LVQNet). For each ANN model, two architectures were constructed separately; 4 layers and 7 layers, each with different numbers of hidden layer units (5, 10 and 15). Ten different ANN optimization training algorithms (LM, BFG, BR, CGF, GD, GDM, GDA, GDX, OSS and RP) were used to train each model separately.

Many experiments were conducted for each one of the five models. Each model used two different architectures, a different number of hidden layer neurons and ten different training algorithms. The performance results of the models were compared according to mean square error to identify the best ANN model. The results showed that the PatternNet model was the best model used. Finally, comparisons between the ten training algorithms were performed through training the PatternNet model. Comparison results showed that TrainLM was the best training algorithm for the iris recognition system.

Keywords: *Feed Forward, Cascade Forward, Function Fitting, Pattern Recognition, Learning Vector Quantization*

1. INTRODUCTION

Artificial neural networks (ANN) are simple models based on the central nervous system. ANN models offer promising approaches to building successful intelligent computer systems [1]. In the recent years, ANN models have been conducted for pattern recognition, image processing, speech recognition, data compression and forecasting. The most popular ANN models are the multi-layer feed forward networks BPNN which use a backpropagation training algorithm (BP). BP is useful only when ANN architecture is chosen correctly. Too small ANN cannot learn the problem well, whereas too large ANN will lead to over fitting and poor generalization performance [2].

A biometric system provides automatic identification of any person based on unique feature like facial features, iris, voice, hand geometry, handwriting and fingerprints. Iris recognition is the most reliable biometric system available because of iris uniqueness. Iris recognition analyzes the iris pattern. Iris pattern is a combination of specific characteristics such as cornea, filaments, crypts, freckles, pits, radial furrows and striations. Iris patterns are more unique, stable and reliable with age in comparison with other biometric features such as fingerprints and faces [3].

An iris recognition system includes: localization of iris region, data set generation of iris images, and finally iris pattern recognition. Biometric systems play an important role in security domains [4]. At the same time, a biometric authentication technique based on iris patterns is suitable for high level security systems. Successful biometric systems require accurate and fast algorithms with less memory implementation especially for large programs and database applications [5]. Therefore, many literature researches were concerned with constructing iris recognition algorithms according to these specifications.

Abiyev and Altunkaya (2008) [4] proposed an algorithm for localization of inner and outer boundaries of the iris region. The located iris is extracted from an eye image and represented by a data set after normalization and enhancement. ANN is used for the classification of iris patterns based on this data set. This ANN is trained using an adaptive learning strategy. Their results illustrate the effectiveness of ANN in personal identification. Leila, et al, (2010) [6] proposed an iris recognition method based on discrete wavelet covariance using competitive ANN. They used a set of edges of iris profiles to build a covariance matrix by the discrete wavelet transform using ANN. Gopikrishnan and Santhanam (2010) [7] illustrated ANN approach for iris patterns identification. For optimization, they extended their work for iris recognition using two ANN models [8]. They used a cascade forward back propagation ANN model (CFBPNN) and an FFBPNN model. Their results showed that the performance of the CFBPNN model is better than that of the FFBPNN model. The authors of paper [7] tried in another work [9] to save the computation effort with no loss in accuracy by reducing the size of templates from 20×480 to 10×480. They extended their work for iris recognition optimization using various ANN training model algorithms. Whereas, Aishwarya et. al, (2011) [10] built an iris recognition system to overcome the limitations of personal identification methods. They used a fast algorithm for the localization of iris region. They used a fuzzy neural network algorithm to extract the deterministic patterns in iris in the form of feature vectors. Then, feature vectors were compared in terms of weighted hamming distance to verify the identity. They used a binary coding scheme for better efficiency. Sherline (2011) [11] presented a face and iris recognition system using ANN and principal component

analysis (PCA) to enable the detection of changes in face and iris images of individuals to an appreciable extent. The recognition system was able to tolerate local variations in the face or iris image of an individual. The performance of both systems was evaluated by comparing its recognition rate. Murugan and Savithiri (2011) [12] presented an iris recognition system based on a partial portion of iris patterns using Back Propagation Neural Network (BPNN). They compared their experimental results with the results of the previous methods. Rashad et al (2011) [13] proposed a hybrid model for iris recognition based on local binary pattern and histogram properties as statistical approaches for feature extraction. Their model was also based on a combined learning vector quantization neural network classifier for classification. They used iris datasets CASIA. Their proposed system gave a high recognition rate of 99.87% compared with other methods. Chaudhary and Mubarak (2012) [14] described the use of BPNN in iris patterns classification. They described, in detail, the image acquisition and segmentation, feature extraction and pattern forming. The recognition rate of their BPNN system was found to be 99.25%. Finally, Saminathan et al (2012) [15] presented a simple method for preprocessing pairs of iris images by taking both left and right eyes of a human rather than either the right or left eye. They presented the design and training of feed forward ANN for the iris recognition system. The best accuracy (93.34%) was obtained with 10 iris block partitions with 10 input layer neurons and 50 hidden layer neurons and only one hidden layer.

Many iris recognition research papers in the literature focused on the strengths and limitations with regard to performance and security. For this purpose, we require an accurate and fast iris recognition algorithm for the person identification system. At the same time, many research papers adopted different ANN models in the iris recognition system with different recognition rates. Therefore, there is a need to identify the best ANN model for iris recognition systems. The objective of this research is to develop an ANN model that can be used for accurate iris recognition systems. At the same time, it is required to identify the best optimization ANN training algorithm to train the best ANN model. This is done by constructing 5 different ANN models (feed forward neural network (FFBPNN), cascade forward neural network (CFBPNN), function fitting neural network (FitNet), pattern recognition neural network (PatternNet) and learning vector quantization neural network (LVQNet)). Each one of these models was constructed separately with two different ANN architectures (4 layers and 7 layers). Each model was trained separately with 10 different ANN training algorithms. A different number of hidden units was used in each experiment.

The rest of this research is as follows: Section 2 includes details about different ANN architectures and training algorithms. Section 3 explains materials and methodology. Section 4 includes implementation of the iris recognition systems including the used database, training and testing samples and system parts. Section 5 includes the experimental results and finally, Section 6 concludes this research.

2. ARTIFICIAL NEURAL NETWORKS MODELS

ANN includes many models. Feed forward neural network (FFBPNN) is the simplest model, which consists of layers as in BPNN. The first layer is connected to inputs. Each subsequent layer has a connection from the preceding layer. The final layer produces the network output. FFBPNN is trained using the BP algorithm according to the following equations [16][17]:

$$U_K(t) = \sum_{j=1}^{n} w_{jk}(t).x_j(t) + b_{0k}(t) \quad \text{..... (1)}$$

$$Y_k(t) = \varphi(U_k(t)) \quad \text{..... (2)}$$

where, $x_j(t)$ is input value of j at time-step t, $w_{jk}(t)$ is the weight assigned by neuron k to input value of j at time t, φ is a nonlinear activation function, $b_k(t)$ is the bias of k-neuron at time t, and $y_k(t)$ is output from neuron k at time t. The process can be repeated for all entries of the time series and yields an output vector y_k. The learning process refers to weight adjustments to minimize the error between the network's desired and actual output using an iterative procedure that adjusts weights. Output y_k is compared with target output T_k using an error function (Eq.3):

$$\delta_k = (T_k - y_k)\, y_k\, (1 - y_k) \quad \text{..... (3)}$$

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

$$\delta_k = y_k\, (1 - y_k)\, \Sigma \delta_k w_k \quad \text{..... (4)}$$

where δ_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 the input layer to update the weight of each connection as follows [17]:

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

Here, η is the learning rate, and the term α is called the momentum factor, which determines the effect of past weight changes on the current direction of movement. Whereas cascade forward neural network (CFBPNN) is similar to FFBPNN but include a connection from input and every previous layer to following layers. Additional connections might improve the speed at which ANN learns the desired relationship. Function fitting neural networks (FitNet) also present a type of FFBPNN, which is used to fit an input output relationship. While a pattern recognition neural network (PatternNet) can be created for pattern recognition problems, it is a feed forward network that can be trained to classify inputs according to target classes. The target data for pattern recognition networks should consist of vectors with all values equal to zero except for a 1 in element i, where i is the class they represent. Finally learning vector quantization ANN (LVQNet) consists of two layers. The first layer maps input vectors into clusters that are found by the network during training. The second layer maps merge groups of first layer clusters into the classes defined by the target data. The total

number of first layer clusters is determined by the number of hidden neurons. The larger the hidden layer, the more clusters the first layer can learn and the more complex mappings of input to target classes can be made.

The traditional BP training algorithm used to train FFBPNN and other ANN models require long time to converge. Therefore, many training algorithms were suggested and described in details in Neural Network Toolbox™ User's Guide R2012a [18]. The equations of all algorithms are the same except for changing weight values.

3. MATERIALS and METHODOLOGY

In this research, five ANN models (FFBPNN, CFBPNN, FitNet, PatternNet and LVQNet) were adopted separately for the iris recognition system. Each one of these models was constructed separately with two different architectures. The first architecture has 4 layers (input, 2 hidden layers and output layer). The second architecture has 7 layers (input, 5 hidden layers and output layer). Fig.1 shows 4-layer FFBPNN with 5 neurons in each hidden layer, whereas Fig.2 shows 7-layer FFBPNN with 5 neurons in each hidden layer. Fig. 1 can be used to describe FFBPNN, FitNet, PatternNet and LVQNet separately, each with 4 layers. The only difference in these ANN is in training functions.

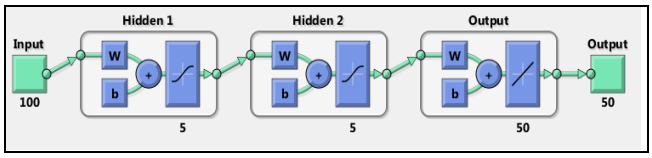

Fig.1: FFBPNN with 4 layers

Fig2 can be used also to show FFBPNN, FitNet, PatternNet and LVQNet separately has 7 layers. However, they differ in training functions. CFBPNN with 7 layers and 5 hidden units in each hidden layer is shown in Fig.4.

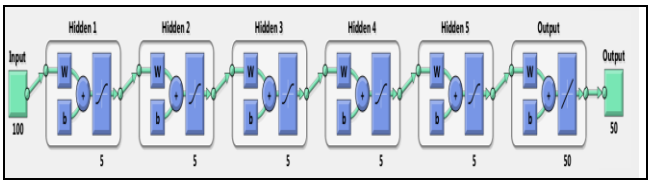

Fig.2: FFBPNN with 7 layers

The CFBPNN with 4 layers is shown in Fig.3, where Fig.2 will be the same when using either 10 or 15 hidden units in each hidden layer.

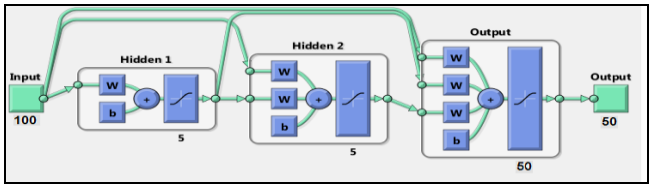

Fig.3: CFBPNN with 4 layers

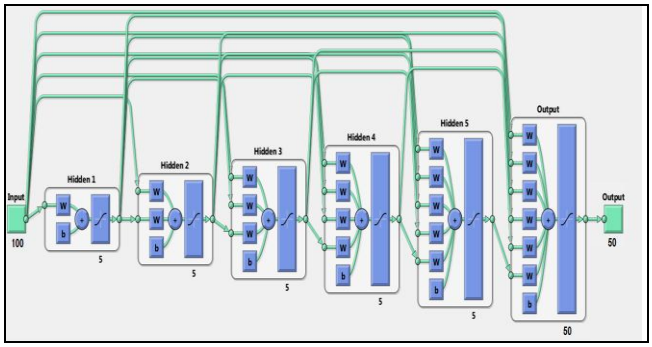

Fig.4: CFBPNN with 7 layers

3.1. The Used ANN Training Algorithms

The optimization training algorithms adjusted the ANN weights and biases to minimize the performance function and to reduce errors as much as possible. Here, mean square error (MSE) is used as a performance function of the suggested iris recognition system and it is minimized during FFBPNN training. MSE represents the difference between the desired ANN model and actual output as shown in Eq.7.

$$MSE = \sum_{i=1}^{n} \frac{\left(error_i \right)^2}{n} = \sum_{i=1}^{n} \frac{\left(Ti - Oi \right)^2}{n} \quad \ldots \quad (7)$$

Here, ten optimization ANN training algorithms were used to train the 5 ANN models separately to identify the model with the best results for the iris recognition system [18][19]:

1. Levenberg-Marquardt algorithm (TRAINLM)
2. TRAINBFG algorithm
3. Bayesian regularization algorithm (TRAINBR)
4. TRAINCGF algorithm
5. Gradient descent algorithm (TRAINGD)
6. Gradient descent with momentum (TRAINGDM)
7. TRAINGDA algorithm
8. Gradient descent momentum and an adaptive learning rate (TRAINGDX)
9. TRAINOSS algorithm
10. TRAINRP algorithm

We used 10 terms (LM, BFG, BR, CGF, GD, GDM, GDA, GDX, OSS and RP) in tables that will be described in the results section to represent the 10 algorithms respectively.

3.2. Preprocessing of Iris Image

The human iris identification process consists of five steps: eye image acquisition, segmentation, normalization, feature extraction and matching [21]. The normalization process is required because irises of different peoples may be captured in different sizes. Also, the size may vary for the same person because of the variation in illumination and other factors. Feature extraction is needed because the iris provides sufficient texture information. A feature vector is formed to

contain the ordered sequence of features extracted from the various representations of iris images. Finally, in the matching process, the feature vectors are classified using different ANN models. More details about iris feature extraction can be found in reference [21].

3.3. Training and Testing Samples

As training samples, 150 iris images were taken for 50 people with 3 samples per person, where the dimensions of each image were 320×280. The 150 samples were taken from the CASIA iris image database Ver. 1.0 (CASIA-IrisV1) [20]. All images are stored in CASIA-IrisV1 in BMP format with resolution 320×280. Fig.5 shows some samples from CASIA-IrisV1. Each one of the iris images is processed using many steps (eye image acquisition, segmentation, normalization, feature extraction and matching) to finally produce the same image but with dimensions 200×170. Each one of the processed iris images with dimensions 200×170 is then divided into sub-images as a set of blocks with dimensions 10×10 each. This operation will result in (200×170)/(10×10) = 340 samples (image sub-block) for each iris image. Therefore, the total number of training samples for the 150 iris images is 51000 [340 samples for each iris] × 150 persons = 51000. These samples are used in the iris recognition system training process.

As testing samples, 200 images were selected from CASIA-IrisV1 each with 320×280 dimensions for 50 people (with 4 samples per person). After applying iris preprocessing steps on randomly selected 50 images form the 200 images, images with dimension 200×170 are obtained. After that each (200×170) image is divided into blocks of dimension 10×10 to obtain 340 sub-blocks for each iris image. Therefore, the total number of testing samples for the 50 randomly selected iris images is equal to 17000.

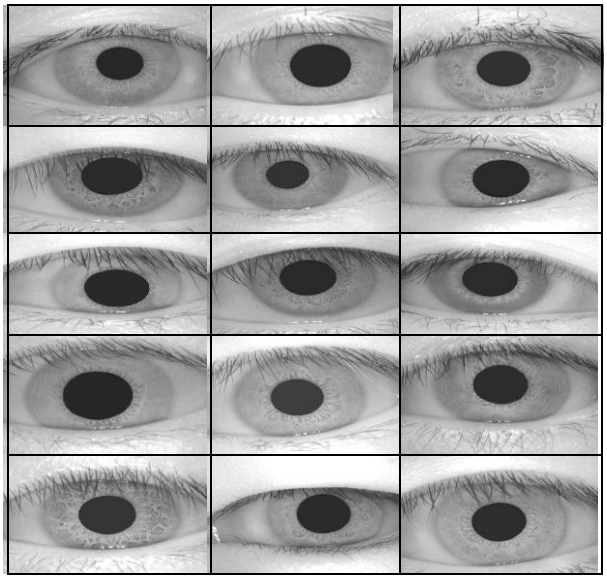

Fig. 5: Samples from CASIA-IrisV1

3.4. The suggested ANN Architecture

Here, the input layer represents the iris image as system input. The number of input layer neurons depends on the image sample dimensions and it is equal to 100.

The number of hidden layers may be more than one according to the problem requirements. In this ANN model for iris recognition systems, different numbers of hidden layer neurons (5, 10 and 15) were adopted separately for each ANN model training process. This is done in order to obtain the ANN architecture with the best results.

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 iris recognition training process. Since 150 images of 50 different persons were adopted, the number of classes is equal to 50 and hence, this is the number of output layer neurons.

4. IMPLEMENTATION OF IRIS RECOGNITION

This section includes the description of the suggested iris recognition system. This system is divided into two parts: an ANN model training part and an ANN model testing part. Fig.6 includes a summary description of the training part of the iris recognition system, and Fig.7 includes a description of the ANN model testing part.

4.1. ANN Training Part

The steps required to train the suggested ANN model for the iris recognition system are as follows:

1. Determine the architecture of the suggested ANN model as described earlier in Subsection 3.4. This includes: number of layers, number of hidden layers, number of input layer neurons, number of hidden layer neurons, and number of output layer neurons.

2. Initialize the ANN model weights and bias unit.

3. Initialize learning rate (0.1-0.9), momentum variable (0.1-0.9) and the threshold error with a small value (0.0000001 produces best results).

4. Initialize the target (desired) output vector for each input vector (10×10 sub-block) of the 150 iris images. For example, let the target output for all sub-blocks related to the three iris images of the first person be:

1	2	3	4	5	6	……..	49	50
1	0	0	0	0	0	……..	0	0

and the target output for all sub-blocks related to the three iris images of the second person be:

1	2	3	4	5	6	……..	49	50
0	1	0	0	0	0	……..	0	0

and so on for each of the 3 iris images of the remaining 50

persons. This process is called classification of iris images.

5. Determine which ANN training algorithm (LM, BFG, BR, CGF, GD, GDM, GDA, GDX, OSS and RP) will be used to train the ANN model on iris recognition.

6. Apply the input vector. Compute outputs of each layer to find the actual output vector. Calculate the ANN error depending on the desired output. According to this error, the training is stopped or repeated again by adjusting the ANN weights according to equations of the selected training algorithm. These operations are repeated until we get an ANN total error equal to or less than the threshold error to stop training process.

Note that, the above steps can be adopted for each of the 5 constructed ANN models separately.

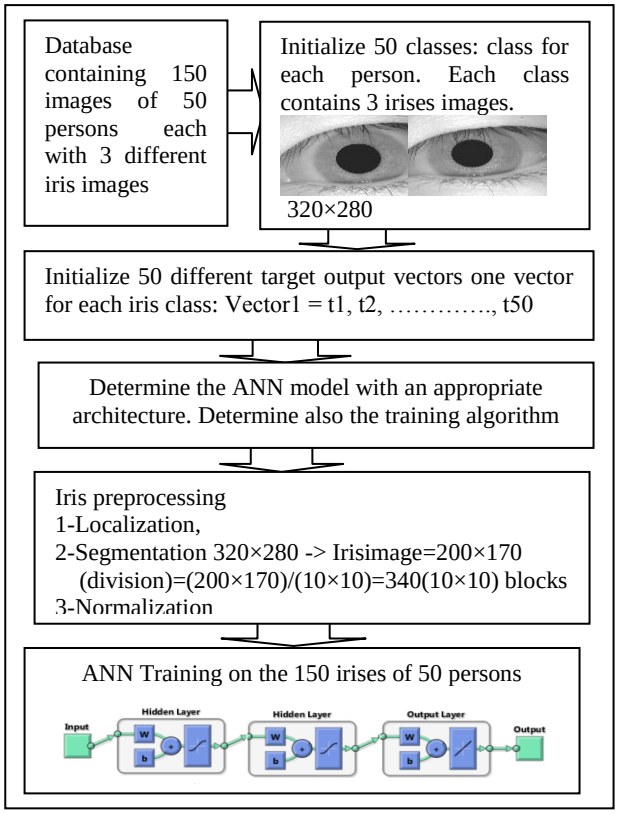

Fig.6: ANN Training for the Iris Recognition System

4.2. ANN Testing Part

After the training process, the result will be a trained ANN model on iris recognition according to the selected ANN model and training algorithm. The testing process of the ANN model includes the following steps:

1. Apply iris sub-block (10×10) to input layer neurons.

2. Compute outputs for all layers of the selected model according to equations of the selected training algorithm until the outputs of the output layer neurons are found.

3. The iris sub-block is recognized by the ANN model if the output vector resultant from applying it is the same as one of the 50 vectors (classes); i.e., if its MSE value is too small. Otherwise, the ANN does not recognize this iris sub-block; i.e., its MSE value is large.

5. EXPERIMENTAL RESULTS

Five different ANN models were constructed separately, each with two ANN architectures (4 layers and 7 layers) for the iris recognition system. Each of the constructed models was trained separately with 10 training algorithms. For each ANN model, we separately adopted a different number of hidden layer units (5, 10, and 15). MATLAB is used to: write programs related to training the different ANN models, and also to write programs for testing ANN models to determine the optimal ANN architecture. Experiments were applied separately for the 5 ANN models; each with 2 ANN architectures using different ANN training algorithms and different numbers of hidden layer units. To evaluate the performance of the constructed models, the number of iterations needed for the training process and MSE were calculated. Various ANN training algorithms for the iris recognition system are discussed.

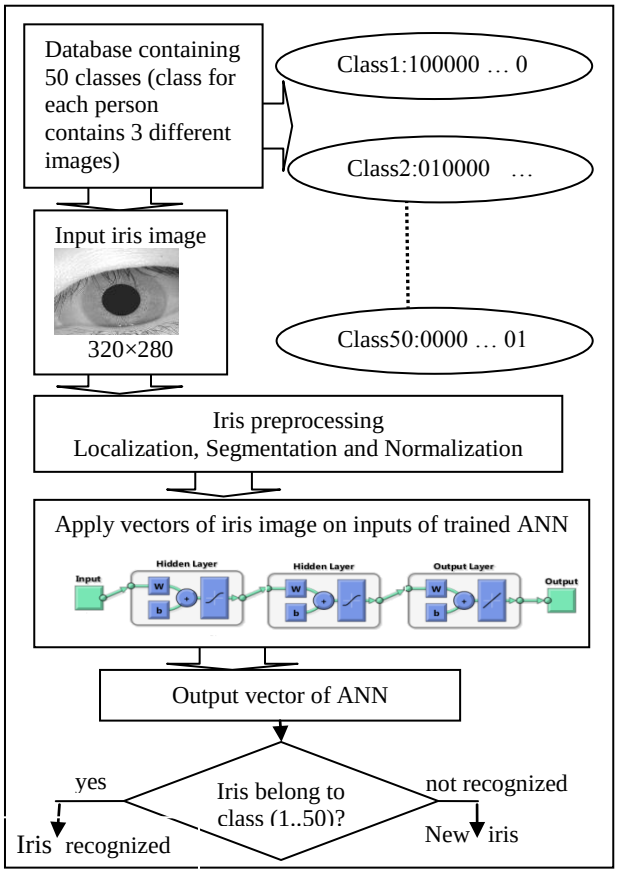

Fig.7: ANN Testing for the Iris Recognition System

5.1. Effect of Number of Hidden layers

The predicted results for each model and training algorithm were compared statistically using MSE. The first experiment was performed by constructing 5 ANN models each with four layers (input layer, two hidden layers each with 5 units and output layer). Table (1) shows the effect of using 5 different ANN models with 10 algorithms on MSE.

Table (1): MSE for 4 layers ANN with 5 hidden nodes

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	0.06	0.09	0.08	0.05	0.10
BFG	0.11	0.12	0.11	0.09	0.18
BR	0.15	0.13	0.13	0.10	0.19
CGF	0.13	0.12	0.09	0.10	0.15
GD	0.37	0.37	0.33	0.27	0.43
GDM	0.36	0.36	0.32	0.22	0.41
GDA	0.27	0.28	0.25	0.22	0.26
GDX	0.31	0.32	0.30	0.21	0.36
OSS	0.12	0.11	0.09	0.08	0.18
RP	0.13	0.13	0.11	0.1	0.15

We can note from table (1) that the lowest values of MSE were obtained for the 5 ANN models when the LM training algorithm was used. Also, MSE values obtained when using the PatternNet model were less than the MSE values obtained when using other ANN models.

The second experiment was performed by constructing 5 ANN models with 7 layers (input layer, 5 hidden layers each with 5 hidden units and output layer). Table (2) shows MSE values when the 5 models were trained separately using 10 different training algorithms.

Table (2): MSE for 7 Layers ANN with 5 Hidden Nodes

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	0.07	0.07	0.06	0.05	0.10
BFG	0.15	0.11	0.08	0.08	0.11
BR	0.12	0.10	0.11	0.09	0.16
CGF	0.11	0.12	0.08	0.06	0.13
GD	0.31	0.33	0.24	0.22	0.34
GDM	0.33	0.30	0.26	0.20	0.35
GDA	0.24	0.25	0.24	0.19	0.32
GDX	0.23	0.29	0.25	0.20	0.33
OSS	0.09	0.12	0.11	0.06	0.12
RP	0.12	0.14	0.13	0.07	0.17

As seen in Table (2), the lowest values of MSE were obtained for all models when the LM algorithm was used. Also, note that MSE values obtained from the PatternNet model were less than MSE values obtained from other models. As shown in Tables (1) and (2), the lowest MSE values for the 5 models with 10 algorithms were obtained when the number of hidden layers was increased to 5 layers. Therefore, we adopted the ANN with 7 layers for the remaining experiments. Table (3) shows the number of iterations required for the training process of the 5 models for the same ANN architecture (7 layers with 5 hidden units in each hidden layer). The best results were obtained from the PatternNet model and LM was faster than those of other algorithms to require the lowest number of iterations.

Table (3): Impact of Algorithms on number of Iterations

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	42	40	20	16	29
BFG	40	39	21	17	26
BR	41	43	22	23	28
CGF	39	49	32	22	37
GD	43	53	30	19	28
GDM	46	51	25	26	33
GDA	50	63	26	25	37
GDX	56	51	33	27	42
OSS	46	43	41	33	52
RP	44	43	46	34	55

5.2. Effect of Number of Hidden layer Units

In order to check the impact of the number of hidden layer units on MSE and number of iterations, the 5 ANN models were constructed separately, each with seven layers (input layer, five hidden layers each of 10 hidden units and output layer). Table (4) shows MSE values obtained when the 5 ANN models were trained separately using 10 algorithms.

Table (4): MSE for 7 Layers ANN with 10 Hidden Nodes

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	0.06	0.07	0.04	0.04	0.08
BFG	0.11	0.1	0.07	0.05	0.11
BR	0.11	0.10	0.08	0.06	0.13
CGF	0.10	0.11	0.06	0.06	0.12
GD	0.30	0.29	0.24	0.21	0.29
GDM	0.31	0.28	0.25	0.16	0.35
GDA	0.22	0.24	0.21	0.18	0.31
GDX	0.27	0.27	0.23	0.15	0.33
OSS	0.08	0.11	0.09	0.05	0.13
RP	0.10	0.11	0.09	0.06	0.12

According to Table (4), the lowest values of MSE were obtained for all ANN models when the LM algorithm was used. Also, note that the lowest values of MSE were obtained when using the PatternNet model.

Table (5) shows that, PatternNet is better than other models and LM was better than other algorithms according to the number of iterations.

Table (5): Impact of Algorithms on Number of Iterations

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	42	44	22	19	32
BFG	45	45	25	21	25
BR	40	42	29	25	33
CGF	45	51	33	31	36
GD	51	55	33	22	29
GDM	52	59	28	24	33
GDA	63	62	33	32	40
GDX	64	63	34	32	43
OSS	46	50	44	40	47
RP	49	53	55	37	55

According to the values of the number of iterations in Table (3) and Table (5), it can be noted that increasing the number of hidden units resulted in increasing the number of iterations required for the training process. Therefore, the smaller numbers of iterations were obtained when 5 hidden units were used in each hidden layer.

Finally, 5 ANN models were constructed with seven layers (input layer, five hidden layers each of 15 hidden units and output layer). Table (6) shows the MSE values for the 5 ANN models which were trained with 10 algorithms. The lowest values of MSE were obtained from the PatternNet model and the LM algorithms. Table (7) shows the number of iterations required to train the same 5 models with 10 algorithms. The 5 ANN models required a larger number of iterations when the number of units in the hidden layer was increased.

Table (6): MSE for 7 Layers ANN with 15 Hidden Nodes

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	0.06	0.06	0.01	0.001	0.05
BFG	0.10	0.11	0.08	0.02	0.11
BR	0.08	0.12	0.07	0.04	0.11
CGF	0.12	0.13	0.08	0.03	0.16
GD	0.29	0.30	0.23	0.15	0.30
GDM	0.32	0.27	0.23	0.16	0.33
GDA	0.19	0.20	0.21	0.14	0.28
GDX	0.23	0.22	0.23	0.17	0.32
OSS	0.08	0.11	0.06	0.02	0.16
RP	0.11	0.15	0.09	0.06	0.15

Table (7): Impact of Algorithms on No. of Iterations

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	50	43	26	20	31
BFG	52	50	32	26	33
BR	40	54	40	29	45
CGF	58	66	46	34	46
GD	67	65	34	25	33
GDM	60	62	30	27	33
GDA	64	65	33	30	41
GDX	72	75	45	39	46
OSS	47	55	55	43	55
RP	54	65	66	41	67

Overall, the lowest values of MSE for the iris recognition system were obtained when the number of nodes in hidden layers was increased for the same architecture with 7 layers. However, increasing the number of units in each hidden layer increased the number of iterations required for the training process.

5.3. ANN Testing Process

As mentioned earlier, the number of testing samples used is 50 iris images. 150 experiments were adopted for the testing process. This is according to: the number of constructed 5 ANN models, the number of used ANN training algorithms and using different numbers of hidden layer units (5, 10 and 15). First, we tested the 5 constructed models each with 7 layers and different numbers of hidden layer neurons (5, 10 and 15) which were trained using 10 algorithms. This was done by selecting 50 different iris images from the training samples.

Table (8) shows the MSE values of the 5 tested models with 50 samples selected from the training samples. The lowest values of MSE were obtained for the 5 ANN models when 15 units were used in each hidden layer. Also, the lowest MSE values were obtained from the PatternNet model and the LM algorithm. Second, the 5 different ANN models were tested each with 7 layers with different numbers of hidden units in each hidden layer (5, 10 and 15) which were trained using 10 algorithms. This was done using 50 testing samples (i.e., iris images which were not used in the training process).

Table (8): Testing ANN Models with Training Samples

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	0.04	0.03	0.02	0.002	0.02
BFG	0.09	0.09	0.07	0.01	0.09
BR	0.10	0.11	0.08	0.03	0.10
CGF	0.13	0.15	0.07	0.02	0.12
GD	0.25	0.26	0.21	0.12	0.25
GDM	0.23	0.24	0.25	0.14	0.28
GDA	0.17	0.19	0.22	0.13	0.25
GDX	0.21	0.20	0.26	0.15	0.27
OSS	0.08	0.10	0.05	0.01	0.14
RP	0.09	0.14	0.08	0.05	0.12

Table (9) shows the MSE values of the 5 tested models with 50 testing samples.

Table (9): Testing ANN Models with Testing Samples

Algorithm	FFBP	CFBP	Fit Net	Pattern Net	LVQ Net
LM	0.81	0.79	0.75	0.66	0.76
BFG	0.86	0.85	0.84	0.82	0.88
BR	0.82	0.89	0.85	0.86	0.84
CGF	0.84	0.88	0.81	0.89	0.88
GD	0.91	0.92	0.90	0.91	0.93
GDM	0.93	0.94	0.93	0.91	0.92
GDA	0.91	0.92	0.93	0.94	0.91
GDX	0.89	0.91	0.92	0.91	0.95
OSS	0.91	0.88	0.86	0.83	0.82
RP	0.84	0.81	0.83	0.81	0.86

The MSE values which were obtained from this test were very high because the 5 ANN models did not recognize the testing samples (not trained images). Note that the lowest MSE values were also obtained from the architecture with 15 units in the hidden layer. The lowest MSE values were obtained from the PatternNet model and the LM algorithm.

6. CONCLUSION

In this research, we presented an iris recognition system using 5 different ANN models and 10 training methods. Five feed forward ANN models (FFBPNN, CFBPNN, FitNet, PatternNet and LVQNet) were constructed for the iris recognition system. Each one of the 5 models was constructed with a 4-layer architecture and then with a 7-layer architecture. This iris recognition system consists of two parts: training and testing. Ten ANN optimization training algorithms (LM, BFG, BR, CGF, GD, GDM, GDA, GDX, OSS and RP) were used to train each of the constructed ANN models separately. The database of the suggested iris recognition system consisted of 150 iris images (with resolution 320×280) which belong to 50 people with three samples for each person. These iris images were taken from the CASIA iris image database Ver. 1.0 (CASIA-IrisV1) [20].

A set of experiments were conducted to evaluate the performance of the suggested iris recognition system by calculating the MSE. This was done using five different ANN models, two different ANN architectures, ten different optimization algorithms, and different numbers of hidden units (5, 10 and 15). After obtaining the training and testing results and computing MSE values, the lowest MSE values resulted from the PatternNet model. The best results for the PatternNet model where obtained from using the Levenberg Marquardt (LM) training algorithm.
Future work may include making a survey of other solutions related to iris recognition and comparing their results to those presented in this paper.

ACKNOWLEDGEMENT

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

REFERENCES

[1] Dan W. Patterson. Artificial Neural Networks, Theory and Applications, Singapore: Prentice Hall, 1996.

[2] Hamid Beigy and Mohamad Reza Meybodi, 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, Jan2009.

[3] Bhalchandra A. S., et al., Iris Recognition, Proceedings of World Academy of Science, Engineering and Technology, vol.36, 1073-1078, ISSN:2070-3740, Dec 2008.

[4] Rahib H.Abiyev and Koray Altunkaya, Personal Iris Recognition Using Neural Network, International Journal of Security and its Applications, vol.2, no.2, 41-50, Apr2008.

[5] Sheela S. V. and Vijaya P. A., Iris Recognition Methods – Survey, International Journal of Computer Applications (0975 – 8887), vol.3, no.5, 19-25, June 2010.

[6] Leila Fallah Araghi, et al., IRIS Recognition Using Neural Network, proceedings of the International Multi conference of engineers and Computer Scientists (IMECS2010), vol.I, 17-19 Mar, 2010, HongKong.

[7] Gopikrishnan M. and Santhanam T., A tradeoff between template size reduction and computational accuracy in Iris Patterns Recognition using Neural Networks, SEEC proceedings, 2010.

[8] Gopikrishnan M. and Santhanam T., Effect Of Different Neural Networks on the Accuracy in Iris Patterns Recognition, International Journal of Reviews in Computing, 30Sep2011, vol.7, ISSN:2076-3328, E-ISSN:2076-3336, www.ijric.org

[9] Gopikrishnan1 M. and Santhanam T., Effect of Training Algorithms on the Accuracy in Iris Patterns Recognition using Neural Networks, International Journal of Computer Science and Telecommunications, vol.2, Issue.7, Oct2011, www.ijcst.org.

[10] Aishwarya Raghavi K. et al., Human Iris Recognition Using Fuzzy Neural Concepts, 2011 International Conference on Bioscience, Biochemistry and Bioinformatics IPCBEE, vol.5 (2011), IACSIT Press, Singapore.

[11] Sherline Jesie R., Multimodel Authentication System using Artificial Neural Network, International Conf. on Emerging Technology Trends (ICETT), Proceedings published by International Journal of Computer Applications (IJCA), 2011.

[12] Murugan A. and Savithiri G., Fragmented Iris Recognition System using BPNN, International Journal of Computer Applications (0975 – 8887), vol.36, no.4, 28-33, Dec2011.

[13] Rashad M. Z. et. al, Iris Recognition Based on LBP and Combined LVQ Classifier, International Journal of Computer Science & Information Technology (IJCSIT), vol.3, no.5, Oct2011, DOI: 10.5121/IJCSIT.2011.3506 67.

[14] Chaudhary U. and Mubarak C. M., Iris Recognition Using BPNN Algorithm, International Journal of Engineering Research and Applications (IJERA), 203-208, ISSN: 2248-9622 National Conference on Emerging Trends in Engineering & Technology, Mar2012.

[15] Saminathan K., et. al., Pair of Iris Recognition for Personal Identification Using Artificial Neural Networks, IJCSI International Journal of Computer Science Issues, vol.9, Issue.1, no.3, 324-327, Jan2012, ISSN:1694-0814, www.IJCSI.org

[16] Haykin S., Neural Networks: A Comprehensive Foundation 2nd edition, Prentice-Hall Inc., 1999. http://www.statsoft.com/textbook/stneunet.html

[17] Elminir H. K., et. al, "Prediction of Hourly and Daily Diffuse Fraction Using Neural Network as Compared to Linear Regression Models," Energy, vol.32, pp.1513-1523. 2007.

[18] 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

[19] Howard Demuth and Mark Beale, Neural Network Toolbox: for Use with MATLAB, User's Guide, Version 4.

[20] Chinese Academy of Sciences, CASIA Iris Image Database Ver. 1.0 (CASIA-IrisV1), collected by the Institute of Automation (CASIA), http://biometrics.idealtest.org

[21] Ezhilarasan M. et al., Iris Recognition Based On Its Texture Patterns, International Journal on Computer Science and Engineering (IJCSE), Vol.2, No.9, 2010, 3071-3074, ISSN: 0975-339.mfvlmv