Off-line Arabic Hand-Writing Recognition Using Artificial Neural Network with Genetics Algorithm

Khalid Nahar
Computer Science Department, Yarmouk University, Jordan

Abstract: *Artificial Neural Networks (ANN) were used in the recognition of the printed Arabic text with a high rate of success. In contrast, Arabic hand-writing recognition has many challenges, some were tackled in some research recently. In this paper we used ANN in recognizing Arabic hand-written characters with the Genetics Algorithm (GA). The GA was used to search for the best ANN structure. We consider Arabic off-line characters represented by a series of (x, y) coordinate. The dataset was gathered from a couple of volunteers, used the E-pen to write different Arabic letters. A* Matrix Laboratory **(***Mat Lab) program was implemented to store the written characters and extracts their features. Features were determined based on the shape and number of segments that made up the characters. The recognition results were very promising when using ANN with the GA in comparison with other relevant approaches. On average more than 95% of accuracy was achieved when GA is used to adjust ANN structure in order to get the best recognition rate.*

Keywords: *ANN, GA, Feature vector, character recognition, arabic hand-written text, Hidden Markov Model (HMM).*

Received January 30, 2016; accepted March 27, 2016

1. Introduction

Arabic hand-writing recognition is an evolving research area which has been tackled in a significant number of research papers. Although the field is about developing techniques to recognize Arabic characters, these techniques can be further applied for similar languages such as: Persian, Kurd, and Urdu. Arabic language consists of 28 characters, they are written from right to left in a cursive script. Most of them come in more than one shape according to their position within Arabic word. For example, the character Ain (ع) has four different shapes based in its location inside the Arabic word; it can be an isolated Ain(ع) like in the word (فرع), it may occur at the beginning of a word (عـ) like in the word (عماره) , it may also exist in the middle of a word (ـعـ) like in the word (معاذ), and it may occur at the end of a word (ـع) like in the word (مبضع). We focused on the recognition of Arabic hand-written characters in off-line mode, where no direct recognition is to take place while writing. We considered all the characters with their different shapes in a total of 108 characters, Figure 1 shows all possible shapes of Arabic characters based on their location in the Arabic word.

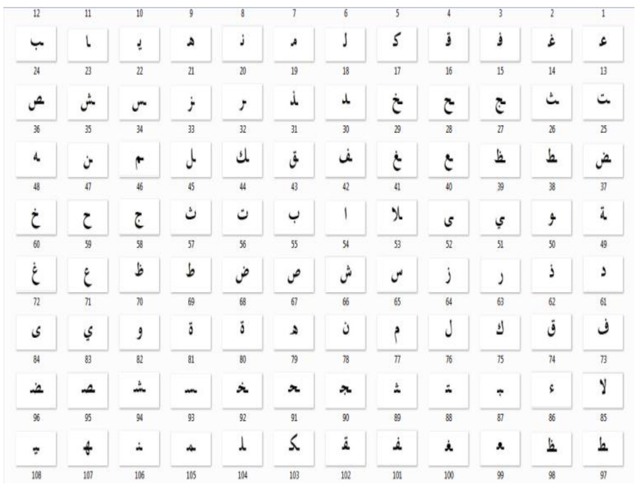


Figure 1. Arabic character shapes based on their locations inside arabic words.

For simplicity, we grouped the 108 models into 54 models based on similarity between them. In data collection phase, a media tablet with an E-pen was used to allow volunteers to input all Arabic characters. A Matrix Laboratory (Mat Lab) program was written to extract feature vectors of input characters. Genetics Algorithm (GA) was used to determine the best Artificial Neural Networks (ANN) structure that will fit for character recognition problem. The composed structure that came from the GA is used to recognize the Arabic hand-written characters.

2. Problem Statement

Off-line Arabic hand-written character recognition is the process of automatic transcription of Arabic characters written by a human hand. Hand-writing recognition was applied on different languages but, less attention was given to Arabic language due to its complexity for Non-Arab researchers. Nowadays, Arabic language becomes one of the main languages used all over the world. Moreover, some other languages like Farsi and Urdu are using the same Arabic character set of Arabic language. In fact, those who use the Arabic character set, they use it in different pronunciations. Many works have been done on hand-written recognition using Hidden Markov Model (HMM) with different rates of accuracy.

In this paper we will focus on the processing of off-line Arabic hand-written character recognition using ANN with GA, in an attempt to improve the accuracy of the recognition process. In this research, some feature changes have been carried out compared to what has been done by other researchers: first, we change the method of extracting the hand-written character feature vectors to enhance recognition. Second, in order to achieve the best recognition rate we used a combination of ANN with GA. GA was used to determine the best numbers of hidden layers, the best initial weights, the best number of neurons in each hidden layer and the most suitable learning and activation functions that will fit the characters recognition problem.

3. Related Works

A text line extraction method has been proposed where a partial contour following based method was used to detect separating lines in the direction of writing and in the opposite direction, Zahour *et al*. [8] and Ben-Amara *et al*. [5]. This was accomplished for every text line and the adjacent lines were separated from each other. Databases of 100 images for Arabic texts written by multiple writers were used for evaluation. Investigation about the effect of diacritics was investigated. Authors proposed an algorithm for text line separation in the processing of unconstrained Arabic hand-written texts.

Al-Emami and Usher [2] introduced an on-line recognition system for hand-written Arabic characters. A tablet was used to input the words and then they used a process to segment the words into strokes. The length and slop of each segment were found to help in the categorization process. In the learning phase, the features of each character stroke are extracted. In the recognition phase, the attributes of each stroke are used to develop rules to find the best matching collection of the stored character features. They considered each version of a character as a different character. The characters that were used in their experiments are the building units of only four Arabic words which are very few for a character recognition system.

Abuhaiba *et al*. [1] addressed Arabic character recognition in an off-line manner. Their system was divided into two parts: preprocessing and recognition. For preprocessing, an algorithm, which produces skeletons that consist of the structure of the components of the characters, was developed. Then the character skeletons are converted to a tree structure to be used for recognition process. In the recognition stage, they introduced flexible models for each character. They also designed fuzzy constrained character graph models with labels and used them as prototypes of the character set. Each model was matched to a character tree using rules. For training and testing, hand-writings of 4 Arabic writers were used to evaluate the system.

Chan and Yeung [6] suggested a simple and robust structural approach for recognizing on-line hand-writing alphanumeric characters. Their experiments achieve a recognition rate of 97.4 % for the combined set (i.e., digits; uppercase and lowercase letters). They applied Freeman’s chain code to represent the directional information of the curves that compose letters and digits. They introduced a specific grammar for recognizing the primitives of the evaluated characters. In the recognition process, they extracted structural features from the sequence of points that were captured by a digitizer. Structural primitives are then extracted and used on a flexible structural matching method to recognize the characters.

In Alma’adeed [4], Arabic word recognition system has been proposed which fully depends on ANN. ANN is combined with a global feature scheme. The system is used to classify Arabic hand-written words which are written by one hundred volunteers. An image smoothing mechanism was used to remove some variation in the images of the words without affecting the main features. Features are then classified using ANN with claimed significant recognition results.

Omer and Ma [7] presented a decision tree and a matching algorithm to learn the stroke direction of Arabic characters. They apply their method on isolated Arabic characters for an attempt to recognize them. Strokes were represented by Freeman direction. The recognition is based on matching and similarity of directional stroke strings. The system showed the ability to recognize Arabic words with connected characters.

Al-Hajj *et al*. [3] studied the off-line recognition of hand-written Arabic with city names as a database. About 1000 entry lexicons contain these names. They used the HMM for classification purposes with sliding window approach. They discovered that the inclination, overlap and shifted positions diacritics marks are major sources of errors. Three homogenous HMM classifiers were used with different orientation of the sliding window. A comparison was held between three combinations of these three classifiers. Referring to the Tunisian city names corpus as a bench mark the paper demonstrates the superiority of ANN classifier and the behavior of the combination of those classifiers improve recognition based on a single classifier. Recognition accuracy in this system reaches about 90% based on the bench mark.

Zimmermann *et al*. [9], made a combination between bottom up chart parser for syntactic analysis of the English sentence with the HMM for recognition. This mixture between bottom up chart parser and HMM improves the recognition rate. This paper puts no constraints on recognition based on grammar of English language since the recognition rate reaches 100% accuracy.

4. Experiment Setup and Verification

Our methodology work flow consists of five ordered phases shown in Figure 2. Data acquisition, preprocessing, features extraction, training (i.e., building the model) and final evaluating or testing the model. The final phase tests and validates the model on untrained data or outlet data to measure accuracy or recognition rate.

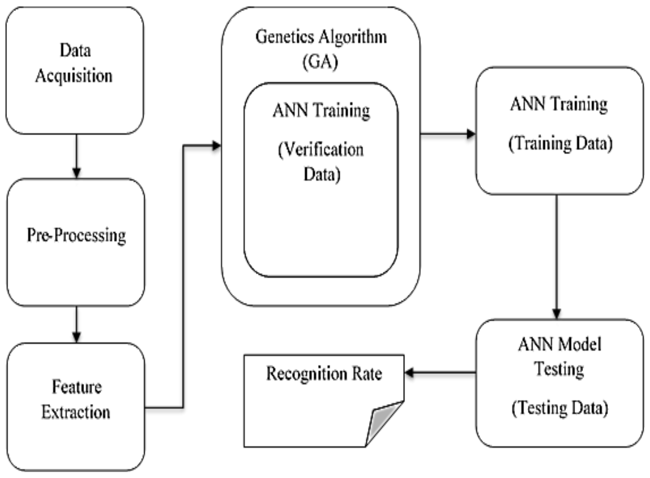

Data Acquisition
Pre-Processing
Feature Extraction
Genetics Algorithm (GA)
ANN Training (Verification Data)
ANN Training (Training Data)
ANN Model Testing (Testing Data)
Recognition Rate

Figure 2. The overall character recognition methodology process.

4.1. Data Acquisition

About 15 volunteers' wrote different shapes of Arabic letters using a media tablet Figure 3, which is a special device for inputting data by writing on it (i.e., a touch screen). A special electronic pen (i.e., Stylus) was used for this purpose. The device was helpful and useful in collecting the data from different users regardless of the way and speed of writing. The most important thing is that, the media tablet looks like a paper for the user and the E-pen as a normal pen. This provides users with comfortable and flexible environment when writing. We build an application with a Graphical User Interface (GUI) that facilitates the process of collecting the data. Figure 4 shows a snapshot of these applications GUI. The application captures a series of (x, y) coordinates from the tablet while the user is writing the letters. A total of 108 letters indifferent forms (based on their locations in the different words) are collected from different users. At the end of this process we came up with the coordination of each pixel in the letter and we recorded the time when the pixel was written.

Figure 3. Media tablet device used in collecting data.

Each handwritten letter was represented based on the Equation (1).

$$\forall\ letter\ l_i:\ l_i = (x_m, y_m), \qquad m = 0,1,2, \ldots n, \tag{1}$$

Where (m) is the number of points in the curve of l_i and (i) is the label of the letters (1...108)

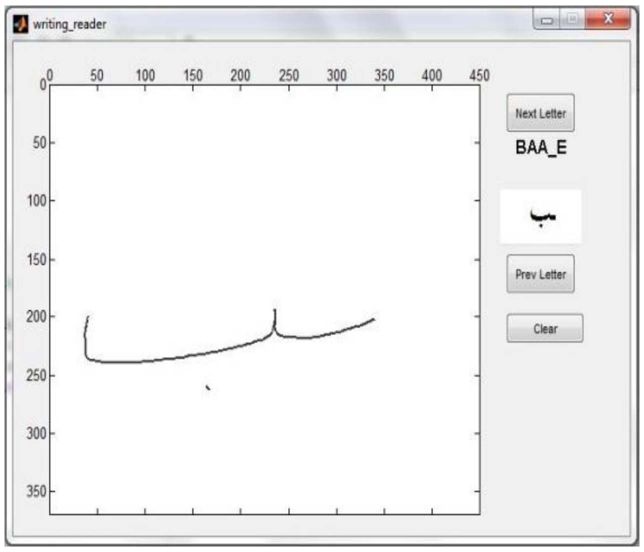


Figure 4. User input screen interface.

4.2. Preprocessing Data

The users entered the letters as mentioned in the previous section which in turn form the data set. The entered curves of the letters are pre-processed, by making a simple normalization for each letter. This operation is accomplished, as shown in Figure 5.Normalization was implemented by translating the curves of a letter so that it's x-coordinate starts from the beginning of the X-axis, and its y-coordinate starts from the beginning of the Y-axis. Equation (2), represents the normalization process.

$$\forall x \in point\ P(X,Y)\ in\ curve\ C, \text{ then}$$
$$x = x - \min(x\ \in\ C), y = y - \min(y\ \in\ C) \tag{2}$$

Another important stage of preprocessing, which is the smoothing step of the letter curve so that the additional noisy pixels results from inaccurate scan of the tablet device, or the user hand movement while inputting data is removed. Figure 6shows the smoothing of the Arabic character "س" as an example. This step is accomplished by a local regression smoothing using weighted linear least squares and a second degree polynomial model. This step may help in letter recognition and it may affect the overall recognition rate.

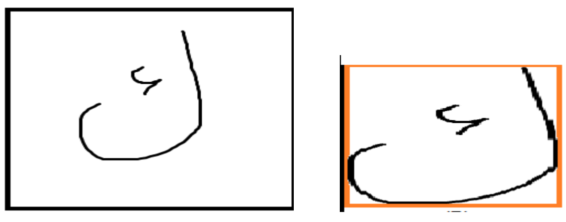

a) The Arabic letter before normalization. b) The Arabic letter after normalization.

Figure 5. The Normalization process of the Arabic letter "ك".

We have identified different segments in the letter by storing a specific tag each time the user releases the Stylus from the tablet. An example of these coordinates for the letter Ba (ب) is shown in Table 1. Those were originally 214 points where we show here only a subset of them.

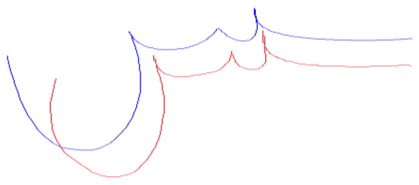


Figure 6. Red character "س" is unsmoothed (bottom), Blue character "س" is smoothed (Top).

Table1. The Ba (ب) Letter coordinates.

X	Y
217	201
214	201
210	201
207	201
203	201
201	201
194	201
192	201
189	201
…	…

4.3. Features Extraction

After the completion of the preprocessing phase, each character is still represented by a series of XY-Coordinates that describes its shape and segments. A set of features was extracted from this representation.

Each letter is converted into a grid of 64 by 64 pixels. This process is based on Equation (3):

$$\forall\, pixel\ p(x,y) \in Curve\ C,$$

$$newx = 64 * \frac{x}{\max(X)},$$

$$newy = 64 * \frac{y}{\max(Y)},$$

$$where\ \ X \in \{x_1, x_2, \dots \dots \dots x_n\},$$

$$Y \in \{y_1, y_2, \dots \dots \dots y_n\}$$

This step was necessary to extract the granularity space that represents the features. Information of each grid cell is extracted based on the following definition set by Equation (4):

$$\forall\ letter\ L\ represented\ by\ 64\ by\ 64\ grid\ then,$$

$$fX_i = \begin{cases} 0 & ,if\ no\ black\ pixel\ in\ col\ i \\ of\ black\ pixels\ in\ col\ i, & otherwise \end{cases} \quad (4)$$

$$fY_j = \begin{cases} 0 & ,if\ no\ black\ pixel\ in\ col\ j \\ of\ black\ pixels\ in\ col\ j, & otherwise \end{cases}$$

Where $i, j = 1 \dots 64$

The combination of fX-Vector and fY-Vector produces 128 features, which represent the main feature vector of each letter.

Another feature is added to the feature vector; is the number of segments in the letter. For example the letter "ك" in Figure 5 consists of two segments; character "ء" and the character "ل". After counting the number of segments in the letter, the largest segment is considered as the main feature that represents the letter while the smaller segments are counted as extra parts like the dots in the letter "ث". The position of the extras like the dot is another feature that is also added to the feature vector. The number of dots in the Arabic letters varies from one letter to another for those letters that contain dots. For example the letter "ث" has three dots while the letter "ن" contains only one dot. The numbers of dots are implicitly added to the feature vector. Figure 7 represents the feature vector parts where the final vector length is 130 features.

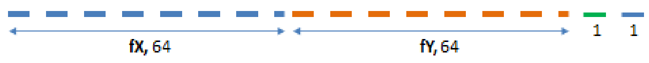


Figure 6. Final feature vector graph.

4.4. The Genetic Algorithm

The genetic algorithm is an optimization technique that is usually used in finding optimal/sub-optimal solutions of searching problems. For example a genetic algorithm is used to find a solution in the 8-Quene problem. In ANN, one of the most open problems is to find the best structure that fits the problem under study i.e. to find the best initial weights, best number of hidden layers, best number of neurons in each layer and best activation function. We formulated this problem into searching/optimization n problem and we used the GA to solve it.

The GA chromosome consists of eleven bits. The first four bits represent the number of neurons in the hidden layer since we have only one hidden layer. The four bit addresses the neurons i.e. the number of neurons not more than 16. The next two bits address the activation functions that will be used in the hidden layer i.e., it will be one of the following activation functions; pure line, hardlim, tansigmoidal and logsigmoidal.

As we mentioned earlier, the next two bits address the activation function of the output layer which is one of the previous functions. The last three bits address the back propagation learning function that is one of eight known functions in ANN.

We used independent validation dataset to apply the GA. Our original data set was divided into 70% for training the ANN and 30% for testing it. The 30% was picked from the 70% in order to be used for finding the best ANN structure with the assist of GA.

Features of the training samples were stored in a 3-D matrix as shown in Figure 7. Each column in the matrix represents a set of samples for the whole letter written by a user. Each row represents different samples for one model or letter written by a user. Each row in the Z-axis represents the set of structural features for one sample.

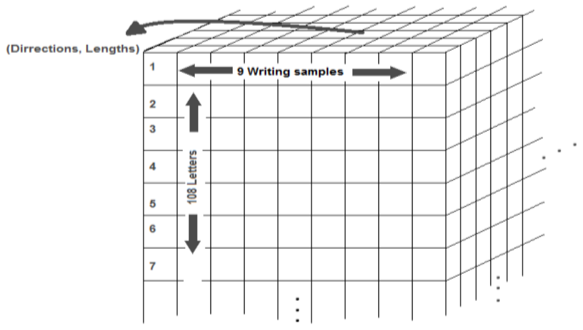


Figure 7. A 3D matrix for training. Cube[i,j,k] = (length, direction) of the i-th sample of the j-th letter of the k-th line segment.

The GA was executed to solve this problem. The output of the GA was the ANN structure that optimally fits our problem. About 25 structures appear to be optimal through the execution of the GA. Figure 9shows the best and mean fitness values of the GA.

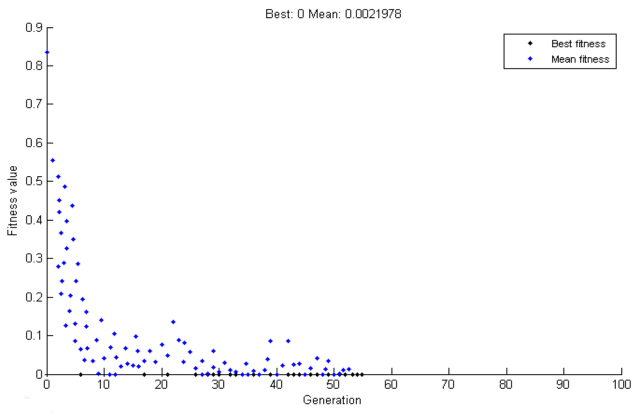


Figure 8. Best and mean fitness values of GA.

4.5. ANN Structure

In the ANN training phase we used the resulted ANN structure as a benchmark structure. The structure consists of 130 input represent the feature vector, one hidden layer with 13 neurons and one output layer.

The activation function for the hidden layer was Tan sigmoidal function. For the output layer we used the pure line activation function. The ANN was a Multi-Perceptron back propagation structure, with Trainlm learning function which uses Levenberg-Marquardt to update weights and bias values accordingly. Figure 10 represents the ANN structure.

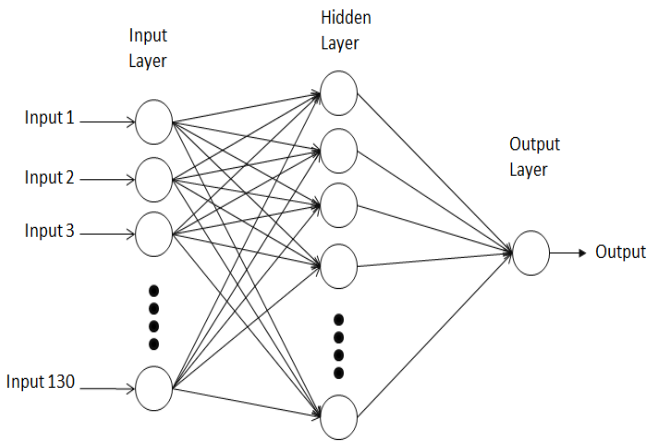
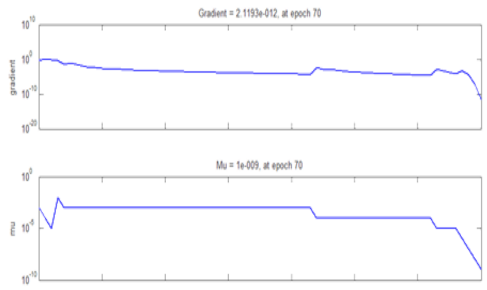


Figure 9. The ANN structure.

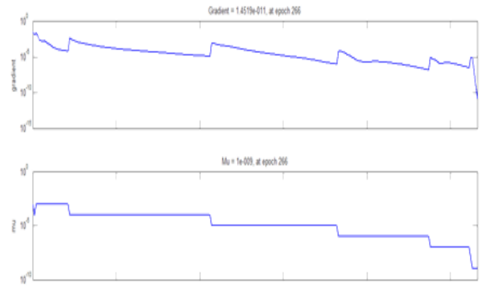
4.6. Training, Testing and Results

We have a collection of 1728 samples entered by 15 users; each sample represents one Arabic letter. For simplicity, we reduced the 108 characters into 54 models. Each model groups the letters with similar shapes i.e., the letter “ص” and the letter “ض” will be in one group/model and the letters “ب”, “ت” and “ث” represent another group.

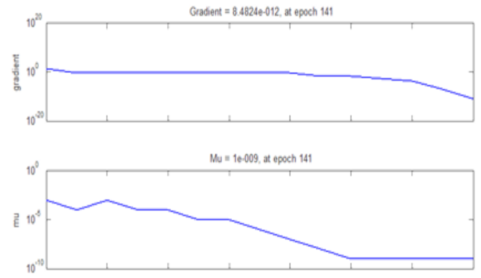
In order to train the ANN we used 70% of the dataset and the remaining 30% were kept for testing purposes. The training/testing processes repeated five times, each time the training and testing sets were chosen randomly (Shuffled). Figure 10shows the training states in the five runs where gradients appear on the top graphs, while μ appears on the bottom graphs.



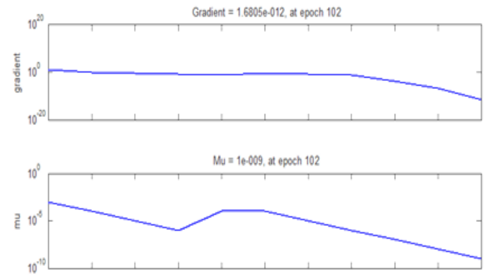
a) Training run no.1.



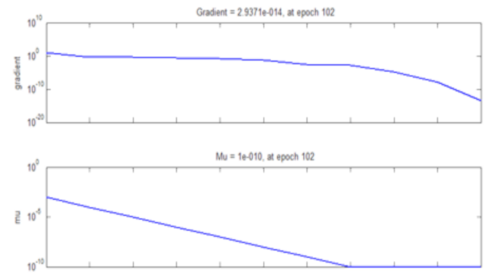
b) Training run no. 2.



c) Training run no. 3.



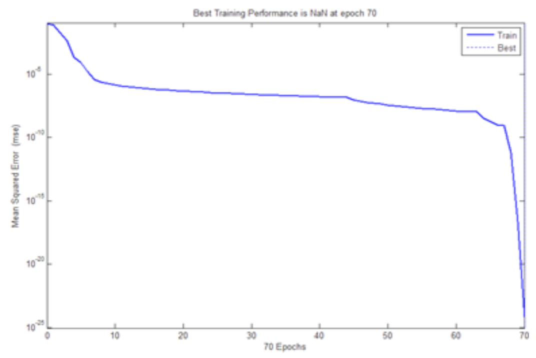
d) Training run no. 4.



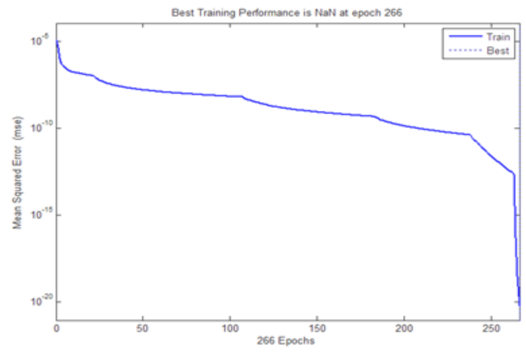
e) Training run no. 5.

Figure 10. Training states for five runs, the upper graph is the (Gradiants) and the bottom graph is (μ).

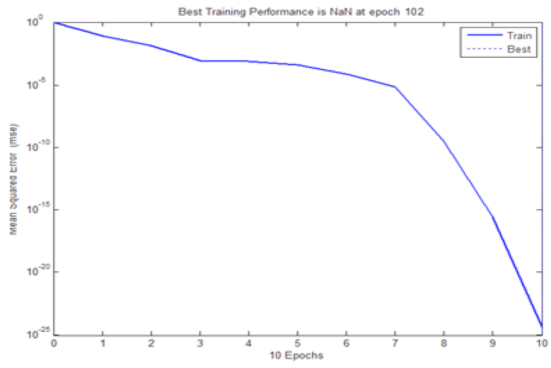
Testing of the performance is also repeated for five times and average is taken. Figure 11 shows these performances testing in the five runs.



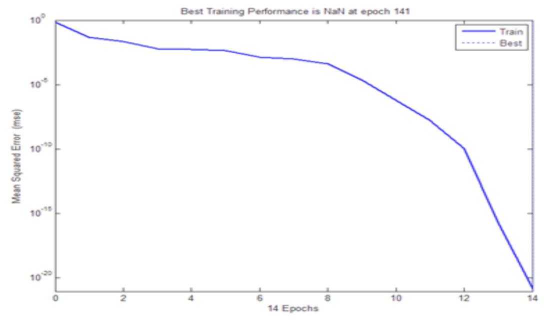
a) Preformance test no. 1.



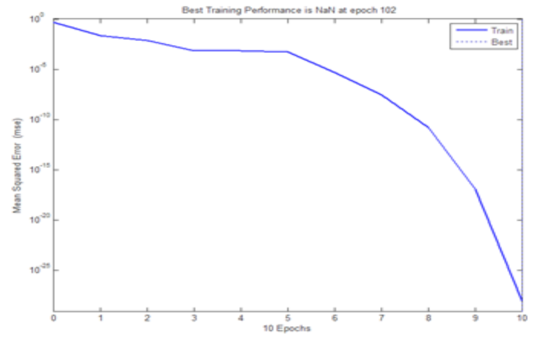
b) Preformance test no. 2.



c) Preformance test no. 3.



d) Preformance test no. 4.



e) Preformance test no. 5.

Figure 11. Performance testing for five runs, the X-axis is the mean square error, while the Y-axis is the best training performance.

For each run we got the prediction accuracy in the testing phase. Table 2 shows the accuracy of the recognition, for each one of these five runs.

Table 2. Accuracy in each test case.

Test Number	Number of Epochs	Mean Square Error	Accuracy
Test 1	70	10^{-20}	0.9597
Test 2	266	10^{-30}	0.9961
Test 3	102	10^{-28}	0.9888
Test 4	141	10^{-27}	0.9885
Test 5	102	10^{-22}	0.9792

The average accuracy for the five tests was 0.98246. This recognition rate could be considered a good improvement in comparison to other similar accomplished results such as those described in [4]. The selected features and the use of GA played a significant rule in increasing the prediction accuracy.

5. Conclusions and Future Work

In this paper, we used the GA to seek for the best structure of a neural network. Such approach enhances ANN accuracy and performance. Moreover, the usage of this combination with suitable features as those we suggested in character/text recognition problem gives a high recognition rate. In our work we achieved an average of 98.2% accuracy, and 99.6% maximum accuracy.

This result is considered to be one of the highest recognition rates in the field of Arabic character hand-writing recognition. The reasons behind this high rate of recognition are: the smoothing algorithm that was applied on the written character and the use of the media tablet with E-pen. The E-pen provides a pure E-shape of the character which is better than pure hand written shape. As a future work we try to remove media tablet and try to recognize purely hand-written characters.

References

[1] Abuhaiba I., Mahmoud S., and Green R., "Recognition of Handwritten Cursive Arabic Characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 664-672, 1994.

[2] Al-Emami S. and Usher M., "On-Line Recognition of Handwritten Arabic Characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 704-710. 1990.

[3] Al-Hajj M., Likforman-Sulem L., and Mokbel C., "Combining Slanted-Frame Classifiers for Improved HMM-Based Arabic Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1165-1177, 2009.

[4] Alma'adeed S., "Recognition of Off-Line Handwritten Arabic Words Using Neural Network," *in Preccedings of Geometric Modeling and Imaging--New Trends*, England, pp. 141-44, 2006.

[5] Ben-Amara N., Mazhoud O., Bouzrara N., and Ellouze N., "ARABASE: A Relational Database for Arabic OCR Systems," *The International Arab Journal Of Information Technology*, vol. 2, no. 4, pp. 259-266, 2005.

[6] Chan K. and Yeung D., "Recognizing on-Line Handwritten Alphanumeric Characters through Flexible Structural Matching," *Pattern Recognition*, vol. 32, no. 7, pp. 1099-1114, 1999.

[7] Omer M. and Ma S., 2010. "Online Arabic Handwriting Character Recognition Using Matching Algorithm," *in Preccedings of 2nd International Conference on Computer and Automation Engineering*, Singapore, pp. 259-62, 2010.

[8] Zahour A., Taconet B., Mercy P., and Ramdane S., "Arabic Hand-Written Text-Line Extraction," *in Proceedings of 6th International Conference on Document Analysis and Recognition*, Seattle, pp. 281-285, 2001.

[9] Zimmermann M., Chappelier J., and Horst B., "Offline Grammar-Based Recognition of Handwritten Sentences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 818-821, 2006.

Khalid Nahar is an assistant professor in the Department of Computer Science, Yarmouk University, Irbid-Jordan. He received his BS and MS degrees in computer science from Yarmouk University in Jordan, in 1992 and 2005 respectively. He was awarded a full scholarship to continue his PhD in Computer Science and Engineering from King Fahd University of Petroleum and Minerals (KFUPM), KSA. In 2013 he completed his PhD and started his job as an assistant proof. at Tabuk University, KSA. He served at Tabuk University from 2013 to 2015. For now he is working at Yarmouk University-Jordan, as an assistant professor in the Department of Computer Science. His research interests include: continuous speech recognition, Arabic computing, natural language processing, multimedia computing, content-based retrieval, artificial intelligence, and software engineering. He had published several papers in his field and gain a fund for his project titled "Knowledge and Intelligent Speech Recognition System To Assist Handicapped Persons" under the number S-1436-0012 From Tabuk University, KSA.