OIAHCR: Online Isolated Arabic Handwritten Character Recognition Using Neural Network

Basem Alijla and Kathrein Kwaik
Faculty of Information Technology, Islamic University of Gaza, Palestine

Abstract: *In this paper, an online isolated Arabic handwritten character recognition system is introduced. The system can be adapted to achieve the demands of hand-held and digital tablet applications. To achieve this goal, despite of single neural networks, four neural networks are used, one for each cluster of characters. Feed forward back propagation neural networks are used in classification process. This approach is employed as classifiers due to the low computation overhead during training and recall process. The system recognizes on-line isolated Arabic character and achieves an accuracy rate 9٥.7% from untrained writers and 99.1% for trained writers.*

Keywords: *Back propagation, classification, feature extraction, feature selection, feed forward neural networks, optical character recognition.*

Received February 15, 2010; accepted May 20, 2010

1. Introduction

Keyboards and electronic mousses may not endure as the prevalent means of human-computer interface. Devices such as tablet PC, hand-held computers, and mobile technology, provide significant opportunities for alternative interfaces that work in forms smaller than the traditional keyboard and mouse. In addition, the need for more natural human-computer interface becomes ever more important as computer use reaches for a larger number of people [3].

Speech and handwriting are natural alternative that can be used easier than keyboard. Handwritten recognition was classified into two types: on-line and off-line recognition [3]. The on-line recognitions requires a direct interaction with user, while off-line recognitions systems apply features extraction upon scanned pictures without any needs to a direct interaction with users. Character recognition is becoming more and more important in the modern world, and it become more and more complex when we deal with cursive language like Arabic language, in addition to the need of character segmentation with offline recognition [11].

In this paper, as first step toward an Arabic words and text recognition, an online handwritten recognition for isolated Arabic character is chosen. In order to solve recognition problem, four feed forward Neural Networks (NN) are used to classify character depending on its extracted features. Set of character's features are selected according to the information gain as a data mining algorithms. The paper is organized as follows: Section 2 explains the basic characteristic of Arabic language. In section 3 talks about related work in this field. Section 4 concerns with neural networks, while section 5 describes system model step by step, Experiments and results are described in section 6. Finally, draws some conclusion points and suggested direction for future works.

2. Arabic Character

Arabic is a language spoken by Arabs in over 20 countries, and roughly associated with the geographic region of the Middle East and North Africa, and it is considered as a second language for several Asian countries in which Islam is the principle religion (e.g, Indonesia). In addition, non-semitic languages such as Farsi, Urdu, Malay, and some West African languages such as Hausa have adopted the Arabic alphabet for writing [1]. Since handwritten isolated Arabic character is the domain of the proposed system, some characteristics which differs it from the other should be known, as stated by Abuhaiba in [1]. Isolated characters have the following interested features:

a. Arabic script is cursive and is written from right to left.
b. Any Arabic character has exactly on main stroke and zero or more secondary strokes as (ش، ظ، ب، س).
c. Usually, a secondary stroke does not touch the main stroke as (ب). If this happens, it will be in limited number of character as (ظ).
d. Some Arabic characters have the same shape; however, they are distinguished from each other by the addition of secondary strokes, e.g., dots, in different positions relative to the main stroke as (ب، ت، ط، ظ). Sometimes, the ambiguity of the position of these secondary strokes in handwriting brings out many different readings for one word.

e. Some Arabic characters contain loops as (ف), but no more than two loops may be adjacent share a common link.
f. Arabic characters vary in size, particularly in width, even within the same font of type printed text.
g. Some Arabic characters use special marks to modify the character accent, such as Hamza (ء) and Madda (~), which are positioned at a certain distance from the character.

3. Related Work

Character recognition has been seen as one of important Pattern recognition pillars. Arabic character recognition has been one of the major languages to receive attention. Since high variability is expected even in printed characters, due to the large number of font styles and other reasons, Nouh in [10], suggested a standard Arabic character set, in order to facilitate computer processing of Arabic characters. Isolated characters are simulated and described by suitably chosen components (radicals). The simulated Arabic alphabet is classified utilizing a sequential tree search technique and certain correlation measurements. The disadvantage of the proposed system is the assumption that the incoming characters are generated according to specified standard rules putting strict constrains on font style design.

Al-Jawfi in [2] presents a handwriting Arabic character recognition method using LeNet NN after applying character segmentation. LeNet neural network was design to recognize a set of handwritten Arabic characters. This NN Design depend on two main stages the first to recognize character shape using pixel matrix of 16×16 an features inputs, while the second stage is to recognize the number of dots, position, and where it is a dot or zigzag using back propagation algorithms. Performance of this algorithm depends firstly of the accuracy of segmentation algorithm in addition to the noise removal. On the other hand, neural networks rely on Image Based features to recognize body shape, which may not hold all of character feature.

Al-Sheik and Al-Taweel in [5] assumed a reliable segmentation stage, which divided letters into the 4 groups of letters (initial, medial, final and isolated). The recognition system depended on a hierarchical division by the number of strokes. One stroke letters were classified separately from two stroke letters etc., ratios between lines and position of dots in comparison to the primary stroke were defined heuristically on the data set to produce a rule-based classification. This approach had an excellent recognition rate and a good divide-and-conquer strategy by reducing the classes through hierarchical rules. However, it would be extremely sensitive to noisy data in terms of the number of strokes since the hierarchy was built on counting the exact number of strokes.

El-Wakil and Shoukry in [6] used stable features to hierarchically reduce the number of letter class considered based on template matching. The stable features were:

1. The number of dots.
2. Relative position of the dots compared with the primary stroke.
3. Number of secondary strokes.
4. Slope of secondary stroke.

A k-nearest neighbour classifier then used primary strokes encoded as a primitive of angular directions in the stroke to determine the closest class. Recognition accuracy varied with the length of primitive strings but the optimal string length gave an accuracy of 84% by testing 7 writers on sets of 60 characters. Weighting the features manually by their relative importance gave a maximum accuracy of 93%. Like many other systems the authors showed good recognition results. Also, like many other systems, this approach's stable features were sensitive to noise and might not generalize well since the results were based on a test set of 60 characters alone.

Zafar *et al*. in [13] describe a simple approach involved in online handwriting recognition by avoiding lengthy pre-processing and extract only useful character information. The system evaluates the use of the Back Propagation Neural network (BPN). The recognition rates were 51% to 83% using the BPN for different sets of character samples. They tested the techniques for upper-case English alphabets for a number of different styles from different subject.

We cannot generalize well since the results depend on the number of samples/characters to determine the rate of performance.

4. Neural Network

An Artificial Neural Network (ANN), often just called a NN, is an information processing system. ANN is a collection of very simple and massively interconnected cells. The cells are arranged in a way that each cell derives its input from one or more other cells. It is linked through weighted connections to one or more other cells. This way, input to the ANN is distributed throughout the network so that an output is in the form of one or more activated cells [4]. Figure 1 shows the architecture of the NN. It consists of 3 layers: the input layer, one hidden layer and the output layer.

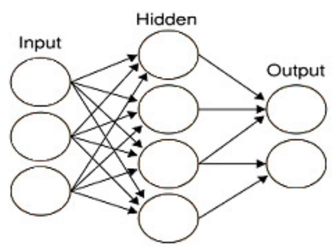


Figure 1. Neural network architecture.

ANN has two main phases in its cycle. The learning phase (training) in which the network adapts its structure based on the input information while the weight of the connection between each unit in the ANN is updated until the best weight is produced. The second phase is called Test phase, in which ANN with stable weight will be run to give the classification result. Back-propagation algorithm, a common method of learning ANN, consists of two phases. First phase is the forward phase; activations are propagated from the input to the output layers. The second phase is the backward phase, the error between the actual observed value and requested output value at the output layer are propagated backwards to modify the weights and bias values [4]. The following algorithm is used in ANN:

Initialize the weights in the network (often randomly)
Do
1. *For each example e in the training set*
2. *O=neural-net-output(network, e); forward pass*
3. *T=teacher output for e*
4. *Calculate error (T - O) at the output units*
5. *Compute delta_wh for all weights from hidden layer to output layer; backward pass*
6. *Compute delta_wi for all weights from input layer to hidden layer; backward pass continued*
7. *Update the weights in the network*

Until all examples classified correctly or stopping criterion satisfied
Return the network.

5. System Model

Figure 2 shows the flow chart of our methodology, which clarifies the steps used to build the recognition system.

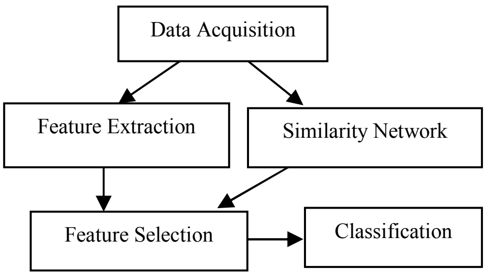


Figure 2. Methodology flow chart.

5.1. Data Acquisition

An external mouse has been used to take the samples from different subjects. Each subject has been asked to write by the mouse. No restriction was imposed on the content or writing style; except the stipulation on the isolation of characters. Every written character will be seen as a black digital ink with white background. Thus one can make use of black and white colours for some useful processing of written characters. Table 1 shows the number of samples taken for each character as input for training phase in section 3.

Table 1. Number of samples/character.

Character	# of Samples
ل، م، هـ، و، ب، أ، د، ر، ص، ح، ع، ج، خ، ذ، ز، غ، ض، ف، ك، ن، ظ، ق	14
س	28
ت	8
ث	20
ش	23
ي	11

5.2. Feature Extraction

Feature extraction abstracts high level information about individual patterns to facilitate recognition. Extracted features should contain the useful information carried by the character image. The purpose of feature extraction is two-fold: to realize that not all data points are equally relevant or useful for pattern recognition and, in the case of NN, further reduction of the data input space to keep the network sizes computationally tractable [4, 7, 8]. Two type of features extraction are considered: the first is on-line extraction during writing the letter by mouse, second is off-line after finish the writing of the whole letter.

5.2.1. Online Feature Extraction

1. *Number of Segments (stroke):* By segment (object) we mean the separate letter component that must be written without lifting the pen (mouse). Character stroke is the segment from click mouse to release it. So, as example, character (ب) has 2 strokes. Figure 3 shows a character that has three segments, surrounded with different colours.

Figure 3. Number of segment.

Arabic characters can be divided into four categories by applying these features. Table 2 shows four categories.

Table 2. Categories of characters.

One- Segment Class (11 Character)
ا ح د ر س ص ع ل م ه و
Two- Segment Class (15 Character)
ب ت ث ج خ ذ ز س ض ط غ ف ق ك ن
Three- Segment Class (4 Character)
ت ظ ق ي
Four- Segment Class (2 Character)
ث ش

This is not considered optimum division for all styles as some one can write Thaa "ث" with two segments as in Figure 4, so another feature must be applied.

Figure 4. Different style.

2. *Letter Direction:* The direction (stroke sequence) method is used in online handwritten recognition. The stroke is defined as the direction of the pen (mouse) movement from one pixel to the next. The direction of the main object (letter body) will be extracted while the writer writing the letter by comparing each pixel with previous one if it is in the left, right, top or bottom. After calculating the number of each direction, the ratio is the division between this number and the whole letter pixel. Each pixel is compared with the previous one to show the direction it will go to. So to measure the right direction of one character, the following equation is used:

   Right Direction =
   *(number of pixel in right direction from previous one / number of all pixel) * 100* (1)

   For example: if pixel 1 has coordinates (3, 5) and pixel 2 has (2, 7) so for horizontal direction pixel 2 is in the left of pixel 1 and in the vertical direction pixel 1 is in the top of pixel 2. Figure 5 shows the vertical and horizontal direction.

a) Shows the direction of letter from x-axis (left- right). b) Shows the direction of letter from y-axis (top-down).

Figure 5. Letter direction.

3. *Secondary Object:* The Arabic writing system has five secondary objects: one dot, two dots, three dots, Hamzah (ء), Maddah (~). If the vertical line of the characters Tah (ط) and Thah (ظ) is written in a disconnected form from the main object of the character, then it is considered the sixth secondary. Figure 6 shows some types of these secondary objects. The Algorithm we use is to determine whether the second objects are dot or not.

a) Dot. b) Hamza. c) Vertical line.

Figure 6. Secondary objects styles.

- *Secondary Object Location:* The location of the secondary object can be above, below, or within the main object, as in Thal (ذ), Baa (ب), and Jeem (ج), respectively. The written character is divided into main object and secondary objects. Then the type of each secondary and its location relative to the main object is found. In the Arabic language writing system, the location of the secondary object is very important. Some characters have the same main object shape but differ in the location of the secondary object, e.g., Jeem (ج) and Khah (خ).
- *Secondary Object Direction:* This feature is similar to the First object Direction, but we use to differ between some letter. Figure 7 shows how this feature will make a difference in classification between Taa "ت" and Thaa "ث".

a) Show the direction for the secondary object for Thaa. b) Show the direction for the secondary object for Taa.

Figure 7. Second object direction.

5.2.2. Off Line Features Extraction

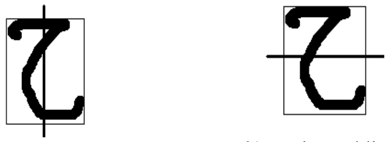
The features extracted from a character after completing its writing are called off line features. The number of these features is less than on line feature; these features are:

1. *Density:* The character area is the total number of white and black pixels of the writing character Panel. [8] The density is the number of drawing pixels to character area ratio. Before calculating density, we must determine the boundary of the letter as in Figure 8 in order for any size to have the correct ratio.

a) Big letter in large boundary area. b) Small letter in small boundary.

Figure 8. Density rates.

2. *Aspect Ratio:* Since different writers write the same character in different sizes, the absolute width and height is not a reliable feature to recognize handwritten characters. However, some Arabic characters are wider than others. Therefore, the aspect ratio (height/width ratio) is a useful feature [9, 12].
3. *Character Aignment Ratio:* Many characters have a noticeable property. Distribution of a written character on fixed boundary and calculating density for the whole character do not explain the alignment of the character. In this feature character is divided into two parts bottom-up (horizontal alignment) or left-right (vertical alignment). As shown in Figure 9.



a) Show vertical line. b) Horizontal line.

Figure 9. Character alignment lines.

5.3. Q network (Similarity-Network)

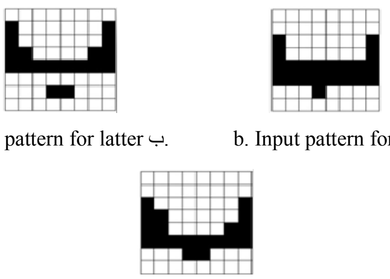
We called *Q* network, considered as a type of NN, similarity network because it is used to calculate the similarity rate between two letter images, so it can be used as a means for classification. Q-value provided by this technique is used as a feature, which improves the system accuracy and increases the NN performance in the next steps. As NN, this technique has training and testing phases but in a different way. The purpose of training phase is to produce a weight matrix to represent each Arabic letter, so there are 28 weight matrices one for each letter, each matrix consists of 100*100 elements equal to the letter size after resizing process. During the training process, the input to the *Q* network is the input matrix *M* which is depending on the Matrix *I* produced after binarization (see binarization algorithm). The input matrix M defined as follows:

If I (i,j) = 1 Then
M (i,j) = 1
Else If I (i,j) = 0 Then
M (i,j) = -1

It is typical for any Q-network to learn in a supervised or unsupervised manner by adjusting its weights. In the current method of learning, each candidate character was taught to the network processes a corresponding weight matrix. For the k^{th} character to be taught to the network, the weight matrix is denoted by *Wk*. The weight matrix *Wk* is updated in the following manner:

for all i=1 to x
for all j=1 to y
W k i j =W k i j +M i j

Figure 10 shows the digitization of three input patterns representing ب that are presented to the system for it to learn.



a. Input pattern for latter ب. b. Input pattern for latter ب.

c. Input pattern for latter ب.

Figure 10. Pattern style.

Figure 11 gives the weight matrix, say, *W* corresponding to the alphabet (ب). The matrix has been updated three times to learn the alphabet (ب). It should be noted that this matrix is specific to the alphabet (ب) alone. Other characters shall have a corresponding weight matrix.

-3	-3	-3	-3	-3	-3	-3	-3
3	-3	-3	-3	-3	-3	-3	3
3	-3	-3	-3	-3	-3	-3	3
3	1	-3	-3	-3	-3	-1	3
3	3	1	1	1	1	3	3
1	1	1	1	1	1	1	1
-3	-3	-3	3	1	-3	-3	-3
-3	-3	-3	-3	-3	-3	-3	-3

Figure 11. W matrix of (ب).

A close observation of the matrix would bring the following points to notice:

1. The matrix-elements with higher (positive) values are the ones which stand for the most commonly occurring image-pixels.
2. The elements with lesser or negative values stand for pixels which appear less frequently in the images.

The overall architecture of the Q-Network is shown in Figure 12. The candidate pattern I-which is the binarization matrix – is the input. The block 'M' provides the input matrix M to the weight blocks W_k for each k. There are totally 28 weight blocks for the 28 Arabic characters to be taught (or already taught) to the system[1].

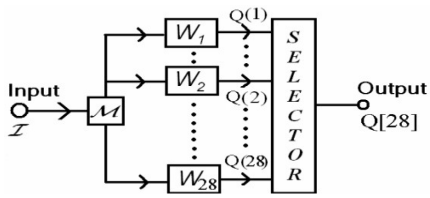


Figure 12. Q-network architecture [10].

The recognition of patterns is now done based on a certain statistics that shall be defined next.

- *Candidate Score (ψ):* This statistic is a product of corresponding elements of the weight matrix W_k of the k^{th} learnt pattern and an input pattern *I* as its candidate. It is formulated using the equation as follows:

$$\omega(k) = \sum_{i=1}^{x} \sum_{j=0}^{y} W_k(i,j) * I(i,j) \quad (2)$$

It should be noted that unlike in the training process where *M* was the processed input matrix, in the recognition process, the binary image matrix *I* is directly fed to the system for recognition.

[1] See Shashank A., "Visual Character Recognition using Artificial Neural Networks," India.

- *Ideal Weight-Model Score (μ):* This statistic simply gives the sum total of all the positive elements of the weight matrix of a learnt pattern. It may be formulated as follows (with μ (k) initialized to 0 each time).

for i=1 to x
for j=1 to y
if $W_k (i, j) > 0$ *then*
$\mu(k) = \mu(k) + W_k (i, j)$

- *Recognition Quotient (Q):* This statistic gives a measure of how well the recognition system identifies an input pattern as a matching candidate for one of its many learnt patterns. It is simply given by:

$$Q(k) = \psi(k)/\mu(k \qquad (3)$$

The greater value of Q, the more confidence system bestows on the input pattern as being similar to a pattern already known to it. The output is an array of 28 elements that describe the similarity degree between one character and the other Arabic characters. So the Final recognition does not depend on the maximum value from Q array but the final value is produced from the final methodology phase which is BPN network described next. To calculate the Q-value for each character there are different steps of Image processing must be applied to the image prior the recognition process, Noise removal, edge detection, resizing and others [13]. In our research we use three pre-processing steps: Cloning, Resizing, and Image Digitalization (Binarization) as shown in Figure 13.

Figure 13. Image processing flow chart.

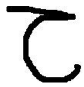
First step is cloning image, after the extraction of all features, each written character should be converted to image then each image must be cloned to a new size matching the boundary of the character. Figure 14 shows cloning image to standard size for all characters.

a) The original image with 148*148 pixels.

b) Cloned image to a size equal boundary of the area.

Figure 14. Cloning image.

Cloning image will lead to variety size of images, so to have a one size images each image must be resized. We resize each image to 100*100 pixels. Figure 15 illustrates this concept.



a) Cloned image with 40*140 pixels.

b) Resizing image to 100*100 pixels.

c) Cloned image with 65*56 pixels.

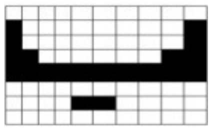
d) Resizing c to 100*100 pixels.

Figure 15. Resizing image.

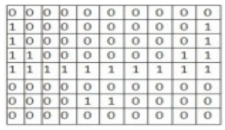
After resizing each image to 100*100 pixels, all images will be digitized and represented by 1's or 0's. Binarization is an important step to Q network. Image was converted to a block of 100*100 black and white pixels; each block was converted to a two dimensional 100*100 array which consists of one's, and zeros for black and white respectively. As shown in Figure 16.

a) Cloning the image.

b) Resizing to 100*100.



c) Binarization blocks.



d) Binarization array (I matrix).

Figure 16. Binarization step.

5.4. Feature Selection

The usage of information gain (Info gain) as a selection criterion presents a very good result in the classification algorithm due to its powerful which become from information theory[2]. Assigning gained information to the weights of attributes (extracted feature) is the main process to select the best features for each cluster.

5.5. Classification (Neural-Network)

Multi-layer feed-forward neural networks, which has high performance approximations of input and output function with back-propagation algorithm, which is a computational efficient used to classify characters [4].

6. Experiments and Results

After applying the system the five steps shown in Figure 2 many times in different experiments, Arabic characters are classified into four clusters (groups) according to the number of segment in each character. Therefore four NN have been submitted in the system one for each cluster of character. According to the selected features, which are selected by info gain algorithms, and number of characters in each cluster, the four NN have different structure. Each NN has different input specified by number of features, and

[2]See Fazil A., Using information gain as feature weight.

different output specified by number of characters in each cluster as shown in Table 3. Table 4 shows the selected features with its rank for every character in the cluster.

Table 3. Characteristics of NN for each segment.

Neural Nets	Number of Segment	Input Feature	Hidden Layer Neuron	Number of Output	Build Time /Second
1	1	20	21	11	5.25
2	2	26	21	15	8.83
3	3	14	9	4	0.44
4	4	8	6	3	0.09

Supervised learning, back-propagation algorithm, as learning algorithms for neural was used to achieve characters classification. NN was trained on different data set , before the testing process , which applied to system for trained and untrained data set as shown in Tables 5 and 6 systems gives an average accuracy of 99.1% with very small error rate for trained data set, and 95.7% for untrained data set.

As it is notes from the Table the fourth group for both trained and untrained data set gives the lowest accuracy rate, which returned to : the small number of feature that are extracted for the forth cluster of characters as shown in Table 3 part D, and large variation of writing style especially. These made classification more difficult.

Table 4. Feature for each segment character.

Part A: 1 Segment		
Number	**Feature**	**Rank**
1	Right Direction	2.0204
2	Bottom Alignment	1.824
3	Bottom Direction	1.8037
4	Qseen [3]	1.7822
5	Top Alignment	1.728
6	Aspect Ratio	1.7052
7	Qhaa	1.6944
8	Qayn	1.6855
9	Top Direction	1.6788
10	Qsaad	1.6744
11	Left Direction	1.6425
12	Density	1.617
13	Qha	1.5986
14	Qmeem	1.5528
15	Qraa	1.4067
16	Qalef	1.3949
17	Qwaw	1.3849
18	Qlam	1.2451
19	Qdal	1.1912
20	Right Alignment	0.9373
Part B: 2 Segment		
Number	**Feature**	**Rank**
1	Right Direction	1.9114
2	Top Direction	1.7956
3	Qzai	1.6649
4	Qfaa	1.6418
5	Left Direction	1.6164
6	Qtah	1.6122
7	Left Alignment	1.6116
8	Density	1.6034
9	Qdaad	1.5762
10	Bottom Direction	1.4909
11	Qkhaa	1.4869
12	Qsheen	1.4485
13	Qthaa	1.446
14	Aspect Ratio	1.4336
15	Qghain	1.3965
16	Right Alignment	1.3866
17	Qkaf	1.298
18	Qtaa	1.288
19	Qjeem	1.251
20	Qthal	1.2435
21	Qnoon	1.1916
22	Second Bottom Direction	1.1532
23	Qbaa	1.0496
24	Object Place	1.0406
25	Top Alignment	1.0241
26	Second Right Direction	0.9699
Part C: 3 Segment		
Number	**Feature**	**Rank**
1	Qthah	1.386
2	Qyaa	1.295
3	Qqaf	1.226
4	Qtaa	1.199
5	Right Direction	1.136
6	Left Direction	0.918
7	Second Bottom Direction	0.918
8	Left Alignment	0.891
9	Right Alignment	0.891
10	Object Place	0.75
11	Bottom Direction	0.701
12	Second Right Direction	0.671
13	Top Alignment	0.527
14	Bottom Alignment	0.527
Part D: 4 Segment		
Number	**Feature**	**Rank**
1	Top Alignment	0.86
2	Qsheen	0.86
3	Bottom Alignment	0.86
4	Density	0.857
5	Qthaa	0.692
6	Right Alignment	0.653
7	Left Alignment	0.653
8	Aspect Ratio	0.561

Table 5. Trained users testing result.

	Net 1	Net 2	Net 3	Net 4
Number of Segment	1	2	3	4
Training Set	240	256	64	11
Testing Set	240	256	64	11
Correctly Classified	100%	100%	100%	٩6.4%
Incorrectly Classified	0%	0%	0%	3.6%
Error/ Epoch	0.0000503	0.0000318	0.0000677	0.0043558
Mean Absolute Error	0.0039	0.0036	0.0091	0.0458
Root Mean Square Error	0.0089	0.0086	0.0113	0.0655

Table 6. Untrained users testing result.

	Net 1	Net 2	Net 3	Net 4
Number of Segment	1	2	3	4
Training Set	168	178	42	7
Testing Set	72	87	27	4
Correctly Classified	97.2222%	97.7011%	100%	٨٧%
Incorrectly Classified	2.7778%	2.2989%	0%	١٣%
Error/ Epoch	0.0000622	0.000049	0.0001173	0.010766
Mean Absolute Error	0.0039	0.0095	0.0137	0.1382
Root Mean Square Error	0.0089	0.00542	0.0193	0.2472

[3] Qx mean the value produced from the Q-network which measure the similarity between the input character and the x character, e.g., Qseen measure the similarity between the inputted character and the seen (س) character.

7. Conclusions

Arabic language has some distinctions from Asian and Latin-scripts languages in which characters are written in various styles, which increase complexity of recognition process, and classification system. NN was proved as a viable concept, and considered as the most successful method used for handwritten recognition, especially for Arabic characters. Integration between neural networks increases the system accuracy. Clustering character into four groups that lead to design the system as four NN with small number of features in each decrease system complexity and increase the accuracy. Finally, Arabic handwriting recognition is a difficult problem but OIAHC system is a step towards a robustly neural network approach to solve Arabic handwritten recognition problem.

8. Future Works

In future we intend to recognize cursive Arabic character (words) not only isolated one, so writer can write the whole paragraph while system recognizes words on line. In addition to the clustering (grouping) characters according to number of segment different clustering techniques can be used like SOM (self organization maps) or other.

Acknowledgements

We would like to express our considerable gratitude to many people who, in one way or another, have helped with the process of doing this research. We would like to take this opportunity to thank Dr. Prof. Nabil Hewahi and Motaz Saad for guiding and helping us throughout this research.

References

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

[2] Al-Jawfi R., "Off Handwriting Arabic Character Recognition LeNet Using Neural Network," *The International Arab Journal of Information Technology*, vol. 6, no. 3, pp. 304-309, 2009.

[3] Alma'adeed S., higgin C., and Elliman D., "Off-Line Recognition of Handwritten Arabic Words Using Multiple Hidden Markov Models," *in Proceedings of the 23rd SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence*, UK, pp. 75-79, 2004.

[4] Bishop C., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

[5] El-Sheikh T. and El-Taweel S., "Real-time Arabic Handwritten Character Recognition," *Pattern Recognition*, vol. 23, no. 12, pp. 1323-1332, 1990.

[6] El-Wakil M. and Shoukry A., "On-Line Recognition of Handwritten Isolated Arabic Characters," *Pattern Recognition*, vol. 22, no. 2, pp. 97-105, 1989.

[7] Khatatneh K., El-Emary I., and Rifai B., "Probablistic Artificial Neural Network for Recognizing the Arabic Hand Written Characters," *Journal of Computer Science*, vol. 2, no. 12, pp. 879-884, 2006.

[8] Khedher M., Abandah G., and Alkhawaldeh A., "Optimizing Feature Selectin For Recognizing Handwritten Arabic Characters," *in proceedings of World Academy of Science, Engineering and Technology*, Spain, pp. 81-84, 2005.

[9] Liu C., Nakashima K., Fujisawa H., and Fujisawa H., "Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques," *Pattern Recognition*, vol. 37, no. 2, pp. 265-279, 2004.

[10] Nouh A., Sultan A., and Tolba R., "On Feature Extraction and Selection for Arabic Character Recognition," *Arab Gulf Journal for Scientific Research*, vol. 2, no. 1, pp. 329-347, 1984.

[11] Shubair A., Amer A., and Rosalina A., "Off-Line Arabic Handwritten Word Segmentation using Rotational Invariant Segments Features," *The International Arab Journal of Information Technology*, vol. 5, no. 2, pp. 200-208, 2008.

[12] Trier Ø., Jain A., and Taxt T., "Feature Extraction Methods for Character Recognition-A Survey," *Pattern Recognition*, vol. 29, no. 4, pp. 641-662, 1996.

[13] Zafar M., Dzulkifli M., and Razid M., "Write Independent Online Handwritten Character Recognition using A Simple Approach," *The International Arab Journal of Information Technology*, vol. 5, no. 3, pp. 476-484, 2006.

Basem Alijla a Lecturer in the Department of Information Technology Systems, Faculty of Information Technology, Islamic university of Gaza since March, 2005 till now. Earned my BSc of computer science from Islamic University of Gaza in 2002, and hold a MS of computer science from Yarmouk University Jordan in 2005, Irbid. My Interested area is Pattern Recognitions and Natural Language Understanding.

Kathrein Kwaik is a Database Programmer in Information Technology and Telecommunication Department at General Personal Council in Gaza, Palestine. She Holds a BSc degree in Information Technology Systems from Islamic University of Gaza, 2008. Kwaik has more than 10 works in analyzing, designing and implementing system for various companies and with different languages. She worked as teacher assistance in Information technology faculty in Islamic University of Gaza from 2008 to 2010. Nowadays she is preparing for master thesis in the same University in Palestine.