On Arabic Character Recognition Employing Hybrid Neural Network

Md. Al-Amin Bhuiyan
Dept. of Computer Engineering
King Faisal University
Al Ahsa, Saudi Arabia

Fawaz Waselallah Alsaade
Dept. of Computer Engineering
King Faisal University
Al Ahsa, Saudi Arabia

***Abstract*—Arabic characters illustrate intricate, multidimensional and cursive visual information. Developing a machine learning system for Arabic character recognition is an exciting research. This paper addresses a neural computing concept for Arabic Optical Character Recognition (OCR). The method is based on local image sampling of each character to a selected feature matrix and feeding these matrices into a Bidirectional Associative Memory followed by Multilayer Perceptron (BAMMLP) with back propagation learning algorithm. The efficacy of the system has been justified over different test patterns of Arabic characters. Experimental results validate that the system is well efficient to recognize Arabic characters with overall more than 82% accuracy.**

Keywords—Arabic characters; Arabic OCR; image histogram; BAMMLP; hybrid neural network

I. Introduction

Arabic language occupies a significant role in mass communication. Over 200 million people speak in Arabic language as mother tongue [1], and more than one billion people exercise it for multifarious religion-oriented matters. Arabic character recognition, therefore, has become one of the exciting areas of research. In spite of its emergent interests in this area, no appropriate solution is presented due to the distinct and intricate characteristics of Arabic scripts.

Numerous research articles have been cited in scientific journals in the field of recognizing English, Chinese, Japanese, Latin, Indian and Bangla characters [2]-[8]. A minute development, however, has been attained in the recognition of Arabic characters, principally owing to their cursive behavior [9]. A simple method for Arabic character recognition system was proposed by Abdelwadood *et al*. [10] where segmentation of Arabic characters were performed by dynamic windowing and correlation were employed to recognize Arabic alphabets. AbdelRaouf *et al*. offered a comprehensive study on multi-modal Arabic corpus for OCR development [11]. Dreuw *et al*. proposed a hidden Markov model based OCR system [12]. Oujaoura *et al*. proposed a Zernike moments based Walsh Transformation for feature extraction and employed neural networks for classification of Arabic characters [13]. Abulnaja and Batawi have proposed a fault-tolerant method to increase the success rate of Arabic character recognition [14]. With cursive styles, Alkhateeb *et al*. [15] employed hidden Markov model for Arabic alphabet identification. Vaseghi *et al*. [16] presented a holistic approach to recognize handwritten Farsic/Arabic word employing discrete Markov chain and Kohonen feature map for Arabic character recognition. Al-Taani *et al*. [17] analyzed the structural features of Arabic characters and made a decision tree learning approach for character identification.

AbdelRaouf *et al*. [18] have proposed the Haar cascade classifier approach which employs discrepancies between rectangular sub-windows to collect features of the Arabic characters. Although the characters with diagonal shapes were prominent while considering the rotated features, but character with other orientations were poorly recognized by their method. Elnagar and Bentrcia [19] have used a neural network to validate the over-segmentation problem in Arabic character recognition and proposed a heuristic-based rule to accumulate strokes for accurate segmentation of characters. Supriana and Nasution [20] have implemented binarization and median filter for Arabic character recognition. They employed Hilditch operator for thinning combined by two templates, one to prevent redundant tail and the other one to eliminate redundant interest points. During segmentation, they employed line segmentation by horizontal projection by connected pixel components, and letter segmentation by Zidouri algorithm. For feature extraction, they used 24 features. Parvez and Mahmoud [21] have segmented the Arabic texts into words and sub-words to extracted the dots and have developed an Arabic handwriting script recognition by means of morphological procedures and fuzzy polygon matching algorithm. Mohammad *et al*. [22] employed three hidden Marcov model skewed windows: aligned to the left, right, and vertical, and combined the effects employing a set of arrangements: addition law, majority vote and multilayer perceptron. Al-Helali and Mahmoud [23] have processed the delayed strokes of Arabic characters and proposed a framework for Arabic character recognition. Although they evaluated the statistical features of Arabic characters but they did not consider the connectivity problems, variability, and style change of text.

This paper proposes a BAMMLP approach for Arabic character recognition that is commenced on local image sampling by converting each Arabic character into a selected $M \times N$ feature matrix. The system is organized with a Bidirectional Associative Memory (BAM) and a Multi-Layer Perception (MLP). The remainder of the article is organized as: Section II describes salient features of Arabic scripts, Section III describes the proposed Arabic OCR algorithm, Section IV highlights the architecture of BAMMLP network, Section V outlines the experimental results, and finally the conclusion section outlines the overall conclusions of the article.

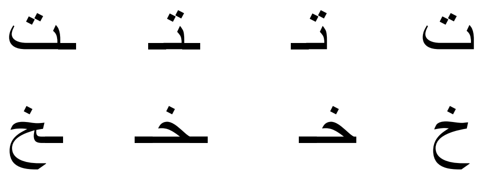
II. Salient Features of Arabic Characters

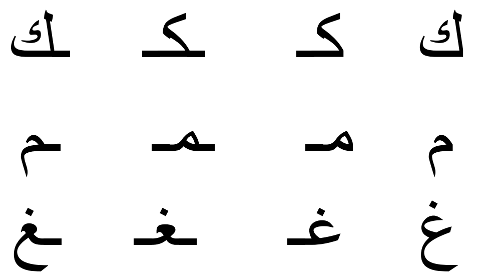
Arabic scripts are written from right to left and are always cursive [24], [25]. There are 28 basic characters and each character has multiple forms depending on its place in the word. Table 1 shows the 28 Arabic characters with their numerous forms: Isolated, Beginning, Middle, and End forms.

TABLE. I. Arabic Characters

SL	Characters	Isolated	Begining	Middle	End
1	Alif	ا	ا	ـا	ـا
2	Baa	ب	بـ	ـبـ	ـب
3	Taa	ت	تـ	ـتـ	ـت
4	Thaa	ث	ثـ	ـثـ	ـث
5	Jeem	ج	جـ	ـجـ	ـج
6	Hha	ح	حـ	ـحـ	ـح
7	Kha	خ	خـ	ـخـ	ـخ
8	Dal	د	د	ـد	ـد
9	Thal	ذ	ذ	ـذ	ـذ
10	Raa	ر	ر	ـر	ـر
11	Zay	ز	ز	ـز	ـز
12	Seen	س	سـ	ـسـ	ـس
13	Sheen	ش	شـ	ـشـ	ـش
14	Sad	ص	صـ	ـصـ	ـص
15	Dhad	ض	ضـ	ـضـ	ـض
16	Tta	ط	طـ	ـطـ	ـط
17	Ttha	ظ	ظـ	ـظـ	ـظ
18	Ain	ع	عـ	ـعـ	ـع
19	Ghain	غ	غـ	ـغـ	ـغ
20	Faa	ف	فـ	ـفـ	ـف
21	Gaf	ق	قـ	ـقـ	ـق
22	Kaf	ك	كـ	ـكـ	ـك
23	Lam	ل	لـ	ـلـ	ـل
24	Meem	م	مـ	ـمـ	ـم
25	Noon	ن	نـ	ـنـ	ـن
26	Ha	ه	هـ	ـهـ	ـه
27	Waw	و	و	ـو	ـو
28	Yaa	ي	يـ	ـيـ	ـي

While writing separately, each Arabic character is patterned in an isolated style and is implied in three different styles when it is joined with other characters. Fig. 1 shows some characters whose isolated forms are distinguished from the Beginning, Middle, and End forms. Characters possessing the same shape but vary in number of dots provide the similar characteristics.





(a) End (b) Middle (c) Beginning (d) Isolated

Fig. 1. Characters whose isolated forms are distinguished from their Beginning, Middle, and End forms.

Arabic scripts belong to the following features [1]:

1) The texts are being written from right to left.

2) Different characters have different sizes.

3) Different characters have different number of dots. Some characters have dots located in the upper side, some have in the lower side, some contain one dot, some contain two dots, some contain three dots, and some characters even do not have any dot.

4) The same character appears in diverse profiles depending on its location in the word.

5) Within a word, every character is usually joined to the preceding character. However, there are six characters that do not attach to the preceding character. These characters have only the Isolated and End forms.

6) Some Arabic words consist of sub-words. Example, the word رسول contains three sub-words: a character ر, the second sub-word سو, and finally the character ل.

7) During formation of words, some characters appear with different compound strings. For example, Noon followed by Alif is written (نا) rather than (ا ن), Lam in the middle of a word is often written as (ـلـ), Ta (ت) and ha (هـ) has other different shapes which are (ة) and (ه), respectively, like in the word “word” (كـلـمـة), (ك ل م ة) and in the word “Lost” (تاه) and not (ت ا هـ).

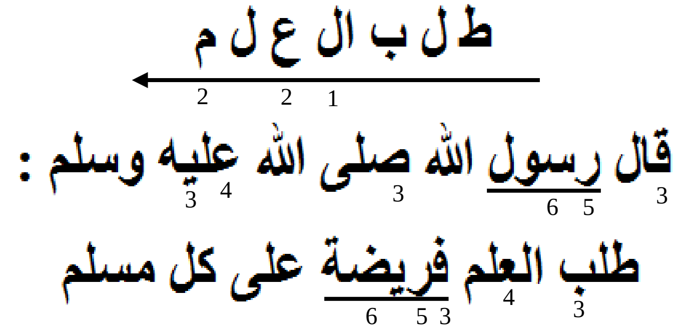

ط ل ب ال ع ل م
2
2
1
قال رسول الله صلى الله عليه وسلم :
3
6 5
3
3 4
طلب العلم فريضة على كل مسلم
3
4
6 5 3

Fig. 2. Features of Arabic script.

Fig. 2 illustrates a precise summary of the striking features of Arabic scripts: 1) written from right to left; 2) different characters have different sizes; 3) different characters have different number of dots, some characters even do not contain any dot; 4) the same character appears with different profiles;

5) some characters are not connected to the succeeding characters; 6) some words consist of sub-words.

III. Arabic Character Recognition

Since the Arabic alphabets possess diverse profiles at different positions of a word and most letters contain one, two, or three dots, the proposed Arabic OCR algorithm, therefore, employs a two stage method: the first stage serves for dots identification; and the second stage is dedicated for recognizing the main shape of the characters. The reason behind dots identification is to reduce the complexity of the problem domain. Since some characters have different number of dots above or below the basic skeleton but have the similar shapes, as shown in Fig. 3, so counting the dots and identification of the basic shape reduces the search space.

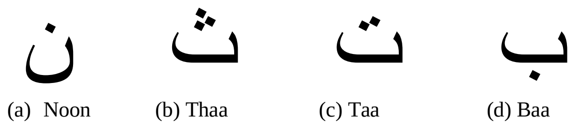

(a) Noon (b) Thaa (c) Taa (d) Baa

Fig. 3. Similar characters with different number of dots.

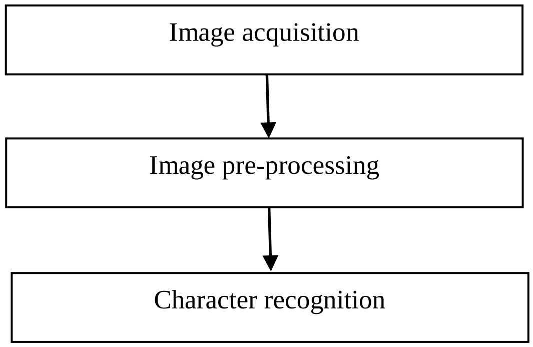


Fig. 4. Steps employed for the proposed Arabic character recognition system.

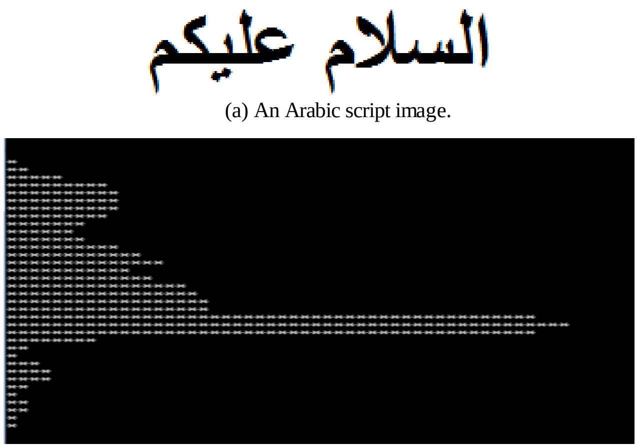
To recognize the main shape of characters, the system employs a three steps procedure, as shown in Fig. 4.

Step 1: *Image acquisition*: The proposed Arabic OCR system is commenced on image acquisition process that scans the texts in 600 dots per inch and the generated images are being saved in .pgm files. This research employs popular Arabic words for image database. After scanning, images of the characters are being Affine (scaling, translation and rotation) transformed [26].

Step 2: *Image pre-processing*: The input images sometimes may be corrupted by various sources of noise. If the noise is not suppressed, it may cause incorrect results. Therefore, these images are filtered by median filter to remove noise and then converted into binary image for processing.

Step 3: *Image recognition*: This step involves word segmentation, character segmentation and recognition steps.

Arabic characters are being segmented by histogram analysis and baseline detection method. The baseline is described by one or more rows with the higher number of black pixels on them compared to other lines. Baselines are being detected by employing histogram construction in counting the number of black pixels followed by white pixels in a single line, as shown in Fig. 5. Subsequently, each line is considered separately for segmenting the words.


(a) An Arabic script image.

(b) Histogram of the image (a)

Fig. 5. Baseline detection.

IV. BAMMLP Network

The BAMMLP is the hybridization of two neural networks: 1) Bidirectional Associative Memory (BAM) network and 2) Multilayer Perceptron (MLP). The design of the BAMMLP network [27] is illustrated in Fig. 6.

Once the image pre-processing is done, the Arabic characters are patterned in a 20×20 matrix and subjected to the input of the BAM network. Thus the matrix pattern is characterized as vectors of 400 neurons. The BAM accepts an input pattern as a vector and generates an associated vector to reduce the size. To develop the BAM, a correlation matrix is created for each pattern pair. The BAM disseminates the input Vector ***A*** to the ***B*** layer where the net input is computed as:

$$y_k = \sum_{j=1}^{N} x_j w_{jk} \tag{1}$$

and control the output values by the thresholding function:

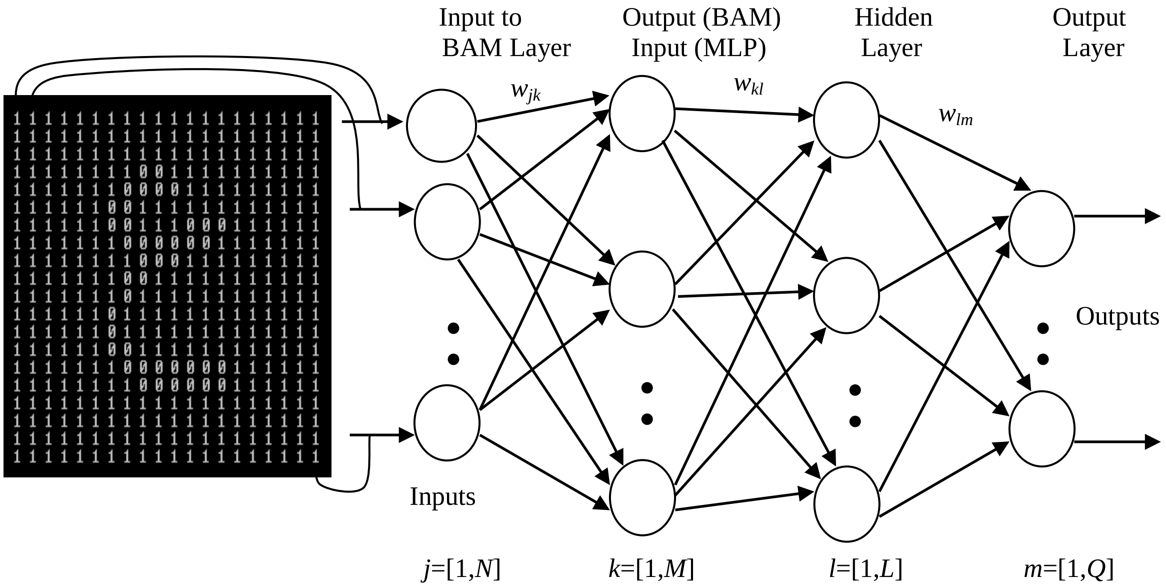
$$y_k(p+1) = \begin{cases} +1 & if\ \ y_k > 0 \\ y_k(p) & if\ \ y_k = 0 \\ -1 & if\ \ y_k < 0 \end{cases} \tag{2}$$

for k=1, 2, …, N.

The pattern ***B*** formed in the Y layer is then disseminated back to the X layer computing the net input as:

$$x_j = \sum_{k=1}^{M} y_k w_{jk} \tag{3}$$

and decide the output values as:



```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 1 1 1 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Fig. 6. Architecture of hybrid neural network.

$$x_j(t+1) = \begin{cases} +1 & if\ \ x_j > 0 \\ x_j(t) & if\ \ x_j = 0 \\ -1 & if\ \ x_j < 0 \end{cases} \qquad (4)$$

The output of the BAM layer is subjected to the input of the MLP. The Multi-layer Perceptron (MLP) is being trained by back-propagation algorithm [28], [29].

Step 1: *Initialization*: Initialize the network with all the weights and threshold parameters of the MLP to small random numbers.

Step 2: *Activation*: Activate the MLP by subjecting the training set $y_k, k=[1,M]$ and the expected outputs $y_m, m=[1,Q]$. Compute the activation of neurons in the *l* and *m* layers:

$$y_l(z) = sigmoid\left[\sum_{k=1}^{M} y_k(z) \times w_{kl}(z) - \theta_l\right], \qquad (5)$$

$$y_m(z) = sigmoid\left[\sum_{l=1}^{L} y_l(z) \times w_{lm}(z) - \theta_m\right], \qquad 6)$$

where *sigmoid* is the sigmoidal activation function, w_{kl} and w_{lm} are the weights between neuron *k* is the input layer of MLP and neuron *l* in the hidden layer, and neuron *l* is the hidden layer and neuron *m* in the output layer, respectively. θ_l and θ_m are the threshold values of the respective neurons.

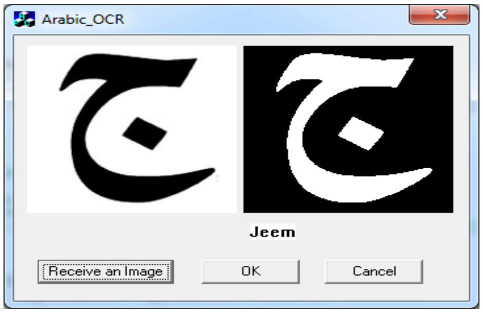
Step 3: *Weight modification*: Modify the weights of the MLP disseminating the errors in the backward direction.

Step 4: *Iteration*: Increase iteration *i* by one, loop back to Step 2 and repeat the process until the error value reduces to the desired level.

For reorganizing Arabic characters, all the characters of Arabic dictionary need not train. Only the basic or mainstream characters (without dots) need to be trained. All other characters can be assessed by means of the information about the position and number of dots containing the characters.

V. Experimental Results

The efficacy of the approach has been validated with numerous Arabic texts of different resolutions. Our system is capable of segmenting and identifying characters in images of various orientations and background conditions. Experiments are carried out on an Intel Core ™ i5-2390T CPU @ 2.70 GHz PC with 4 GB MB RAM. The Arabic character recognition system has been implemented employing Visual C++ programming language. Fig. 7 illustrates the program snapshot for a typical Arabic character individually.



Jeem

Fig. 7. Snapshot of the software interface for Arabic character recognition.

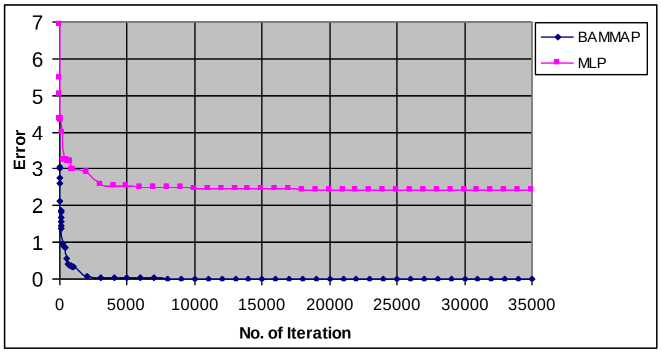


Fig. 8. Error versus iteration.

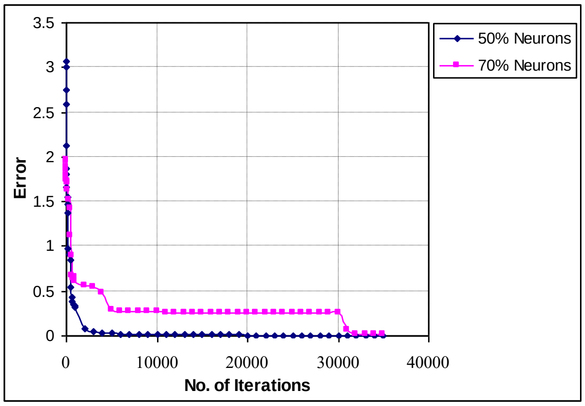
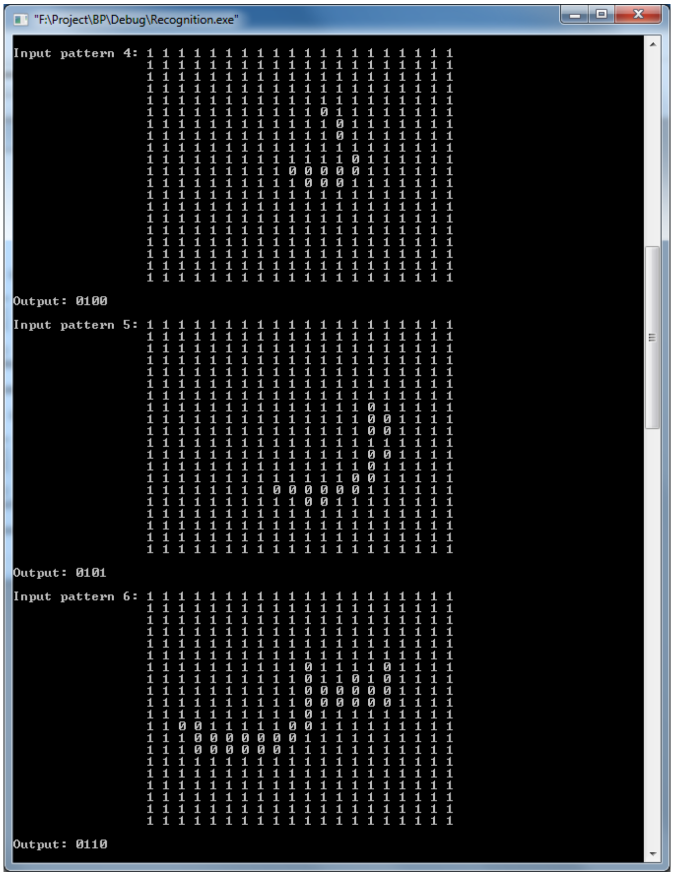


Fig. 9. Error versus iteration graph for BAMMLP.

The learning outcome of the hybrid network has been tested with different experiments. For each character images, investigations were accomplished with 5 training and 3 test images. There was no overlap between the training and test image sets. First the system was implemented employing the multilayer perceptron with back propagation algorithm. Then BAMMLP was employed to train and recognize the Arabic characters. The MLP was trained with back-propagation learning algorithm employing the parameters: learning rate 0.1 and momentum 0.25, for 35000 iterations. During training session, the algorithm runs until the error value reduces to the desired threshold level of 0.001. The error versus iteration graphs for both BAMMLP and MLP are jointly shown in Fig. 8.

The graphs imply that the errors reduce exponentially. Although for BAMMLP, the error value reduces to 0.01 at 1996 iterations, it still remains 0.264 even after 35000 iterations. The graphs reveal that the BAMMLP network outperforms the MLP in terms of minimum number of iterations to train the Arabic characters. Fig. 9 shows the error versus iteration graph for BAMMLP for 50% and 70% neurons with respect to the input layer, respectively. Obviously, as the number of neurons in the hidden layer is less, there is less computational cost and recognition process becomes faster. But for accuracy, we need more neurons in the hidden layer. So there is always a trade off in choosing the number of neurons in the hidden layer. For this experiment, the learning process achieved the expected threshold level within less than 5,000 iterations while choosing the number of neurons in the hidden layer to be 50% of the number of neurons in the input layer. On the contrary, considering 70% neurons in the hidden layer, the same threshold level is being achieved after 30,000 iterations. Therefore, the number of neurons in the hidden layer was chosen as 50% of the number of neurons in the input layer for recognizing Arabic characters. Later on, the BAMMLP hybrid neural network was used to recognize characters randomly, as shown Fig. 10.



```
Input pattern 4: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Output: 0100
Input pattern 5: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1
                 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Output: 0101
Input pattern 6: 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1
                 1 1 0 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1
                 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1
                 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
                 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

Output: 0110
```

Fig. 10. Recognition of three test images (د ,ر ,صِ)

Experiments were conducted separately for Arabic character recognition for four different forms: Isolated, Beginning, Middle, and End form and their outcomes are furnished in the Table 2.

TABLE. II. ACCURACY FOR DIFFERENT FORMS OF ARABIC CHARACTERS

Isolated form	**Beginning form**	**Middle form**	**End form**
91.5%	90.5%	82.71%	84.29%

The recognition rate for different Arabic characters in isolated form is shown in Fig. 11.

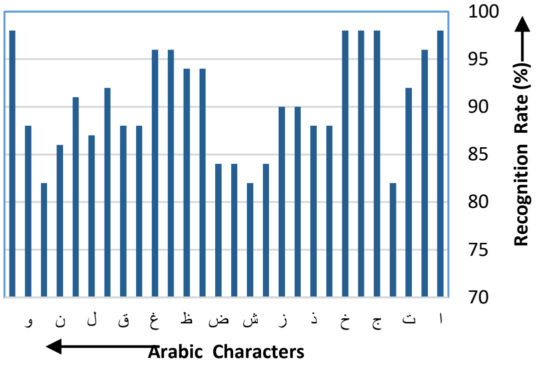


Fig. 11. Recognition rate for different Arabic characters in isolated form.

VI. Conclusions

An efficient Arabic character recognition system has been presented through a hybrid neural network which consists of a BAM and a multilayer perceptron. The system is very fast and is able to carry out the recognition in less than 1ms for all forms of Arabic characters, which demonstrates that the method is an appropriate one for real-time applications. Our next approach will be to recognize Arabic number plate identification for any desired application, including black-lists, white-lists, and alarm functions.

Acknowledgment

The authors would like to express their gratitude to the Deanship of Scientific Research, King Faisal University, Saudi Arabia, for the financial support of the project No 150169.

References

[1] A. Hassin, X.L. Tang, J.F. Liu and W. Zhao, “Printed Arabic character recognition using HMM”, Journal of Computer Science and Technology, Vol. 19, No. 4, pp. 538-543, 2004.

[2] M.R. Gupta, P.N. Jacobson, E.K. Garcia, “OCR binarization and image pre-processing for searching historical documents”, Pattern Recognition, Vol. 40, pp. 389 – 397, 2007.

[3] Y. Yang, X. Lija, and C. Chen, “English character recognition based on feature combination”, Procedia Engineering, Vol. 24, pp. 159-164, 2011.

[4] X.-D. Zhou, D.-H. Wang, F. Tian, C.-L. Liu, and M. Nakagawa, “Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 35, no. 10, pp. 2484–2497, 2013.

[5] Q.-F. Wang, F. Yin, and C.-L. Liu, “Handwritten Chinese text recognition by integrating multiple contexts”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 34, no. 8, pp. 1469–1481, 2012.

[6] C.-L. Liu, F. Yin, D.-H. Wang, and Q.-F. Wang, “Online and offline handwritten Chinese character recognition: Benchmarking on new databases,” Pattern Recognition, vol. 46, no. 1, pp. 155–162, 2013.

[7] M. Kumar, M.K. Jindal and R.K. Sharma, “Review on OCR for handwritten Indian scripts character recognition”, Advances in Digital Image Processing and Information Technology Communications in Computer and Information Science, Vol. 205, pp. 268-276, 2011.

[8] T. Hashem, M. Asif and M.A. Bhuiyan, “Handwritten Bangla digit recognition employing hybrid neural network approach”, Proc. of 16th International Conference on Computer and Information Technology (ICCIT), pp. 360-365, 2014.

[9] B. Al-Badr and S. Mohmoud, “Survey and bibliography of Arabic text recognition”, Signal processing, Vol. 4, pp. 49-77, 1995.

[10] M. Abdelwadood, S. Ahmed, J. Al-Azzeh, M. Abu-Zaher, N. Al-Zabin, T. Jaber, O. Aroob, and H. Myssa'a, “An optical character recognition”, Contemporary Engineering Sciences, Vol. 5, No. 11, pp. 521 – 529, 2012.

[11] C. AbdelRaouf, T. Higgins, T. Pridmore and M. Khalil, “Building a multi-modal Arabic corpus (MMAC)”, International Journal on Document Analysis and Recognition, Vol. 13, pp. 285-302, 2010.

[12] P. Dreuw, D. Rybach, G. Heigold and H. Ney, “RWTH OCR: A large vocabulary optical character recognition system for Arabic scripts”, Guide to OCR for Arabic Scripts Chapter, Part II: Recognition, Springer, London, UK, pp. 215-254, 2012.

[13] M. Oujaoura, R.E. Ayachi, M. Fakir, B. Bouikhalene and B. Minaoui, “Zernike moments and neural networks for recognition of isolated Arabic characters”, International Journal of Computer Engineering Science, Vol. 2, pp. 17-25, 2012.

[14] O. Abulnaja and Y. Batawi, “Improving Arabic optical character recognition: accuracy using n-version programming technique”, Canadian Journal on Image Processing and Computer Vision, Vol. 3, pp. 44-46, 2012.

[15] J.H. Alkhateeb, J. Ren, J. Jiang, H. Al-Muhtaseb, “Offline handwritten Arabic cursive text recognition using hidden Markov models and re-ranking”, Pattern Recognition Letters, Vol. 32, No. 1, pp. 1081-1088, 2011.

[16] B. Vaseghi and S. Hashemi, “Farsi/Arabic handwritten word recognition using discrete HMM and self-organiaing feature map”, International Congress on Informatics, Environment, Energy and Applications, IPCSIT, Vol. 38, No. 1, pp. 55-62, 2012.

[17] A. T. Al-Taani, S. and Al-Haj, “Recognition of on-line Arabic handwritten characters using structural features”, Journal of Pattern Recognition Research, Vol. 1, pp. 23-37, 2010.

[18] A. AbdelRaouf, C.A Higgins, T. Pridmore and M.I. Khalil, “Arabic character recognition using a Haar cascade classifier approach”, Pattern Analysis and Application, Vol. 19, pp. 411–426, 2016.

[19] A. Elnagar and R. Bentrcia, “A recognition-based approach to segmenting Arabic handwritten text”, Journal of Intelligent Learning Systems and Applications, Vol. 7, No. 1, pp. 93-103, 2015.

[20] I. Supriana and A. Nasution, “Arabic character recognition system development”, Procedia Technology, Vol. 11, pp. 334 – 341, 2013.

[21] M.T. Parvez and S.A. Mahmoud, “Arabic handwriting recognition using structural and syntactic pattern attributes”, Pattern Recognition, Vol. 46, pp. 141–154, 2013.

[22] R.A. Mohamad, L. Likforman-Sulem, C. Mokbel, 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.

[23] B.M. Al-Helali and S.A. Mahmoud, “A statistical framework for online Arabic character recognition”, Cybernetics and Systems: An International Journal, Vol. 47, No. 6, pp. 478–498, 2016.

[24] G.A. Abandah and K.S. Younis, “Handwritten Arabic character recognition using multiple classifiers based on letter form”, Proc. 5th IASTED Int'l Conf. on Signal Processing, Pattern Recognition and Applications (SPPRA), pp.128-133, 2008.

[25] K. Addakiri and M. Bahaj, “On-line handwritten Arabic character recognition using artificial neural network”, International Journal of Computer Applications, Vol. 55, No. 13, pp. 42-46, 2012.

[26] M.A.Bhuiyan and H. Hama, “Identification of Actors Drawn in Ukiyoe Pictures, Pattern Recognition, vol. 35, no. 1, pp. 93-102, 2002.

[27] A. Khatun and M.A. Bhuiyan, “Neural network based face recognition with Gabor filters”, International Journal of Computer Science and Network Security, Vol. 11, No. 1, pp. 71-76, 2011.

[28] S.K. Saha, M. Shamsuzzaman and M.A. Bhuiyan, “On Bangla character recognition”, Proc of 13th International Conference on Computer and Information Technology (ICCIT), pp. 436-439, 2010.

[29] M. Negnevitsky, Artifical Intelligence - A Guide to Intelligent Systems, Addision-Wesley Inc., Pearson Edition, London, England, 2002.