Handwritten Arabic Alphanumeric Character Recognition using BP and SOFM Neural Networks

Rafid Ahmed Khalil, Ph.D.,
rafidahmed@hotmail.com
Computer Science Department, Al-Hadbaa University College
Mosul, Iraq

Abstract

This paper presents results obtained by applying two neural networks models Backpropagation (BP), and Self-Organized Feature Map (SOFM) to a new application of handwritten Arabic alphanumeric character (HAAC) recognition. A novel method for features extraction, based on a shadow projection is used. Both networks are trained using Arabic character samples written by different people (learning set). They are required, after the learning is over, to recognize characters out of the learning set. Evaluation of the recognition (classification) capability of the two models for 28 alphanumeric characters is achieved. Depending on the experimental results, a comparison of both algorithms is done.

[illegible]
[illegible] **SOFM** [illegible] **BP**

[illegible]
[illegible]
[illegible]

[illegible]

[illegible] SOFM [illegible] BP [illegible] 28 [illegible]

Received 31st May 2005 **Accepted 10th Nov. 2005**

1. Introduction

Automatic recognition of handwriting has become an important discipline at the beginning of the 21st century. On-line systems are now available on handheld computers with acceptable performance. Off-line systems are less accurate than on-line systems. However, they are now good enough for specialized systems such as interpreting handwritten postal addresses on envelopes and reading currency amounts on bank checks [1].

Needless to say, explicit and proper recognition should always be made as longe as the quality of written characters is very high. Furthermore, from a practical point of view, accurate recognition should be made even if the handwritten style is deformed, or some characters closely reseble each other.

Several literatures were discussed the subject of Arabic characters recognition in detail as in [2-5]. Also Gallinari [6], has investigated the recognition of handwritten digit using BP algorithm. Kamel et. al. [7], have proposed a multiple classification architecture for handwritten Arabic characters recognition. Khalil [8], and Sharaidah [9], have reported on experiments in handwritten Arabic character recognition using neural networks. Amin [10], has presented a survey of Arabic characters recognition.

This paper is organized as follows: In section 2, we describe the use of Backpropagation (BP), and self-organized feature map (SOFM) networks. The handwritten Arabic alphanumeric character HAAC recognition procedure is presented in section 3. Evaluation of the learning algorithm is discussed in section 4. In section 5, an efficiency comparison between these algorithms is performed by the help of some performance evaluation statistics, gathered during the experiments. Section 6 is dedicated to conclusions.

2. Neural Networks for HAAC Recognition

The Backpropagation (BP) and self-organized feature map (SOFM) learning algorithms are applied in handwritten Arabic alphanumeric character (HAAC) recognition.

The network topology of BP algorithm is illustrated in figure 1, It consists of a 3-layer fully connected feed forward network. The algorithm is used to train this network having feature maps as inputs and alphanumeric character classes as outputs[11]. The network topology used in self-organized feature map (SOFM) algorithm is shown in figure 2 [12]. It should be noted that, in self-organization, the values of output neurons have no importance in handwriting recognition, since all the information is stored as weight values, not as output values.

3. HAAC Recognition Procedure

3.1 Arabic Alphanumeric Characters

The Arabic characters consists of 28 basic characters and 10 numerals. The Arabic characters differ from other systems of characters in their structure and in the way they connect to form words. The same character may take different shapes according to its position in the word. This feature increases the number of Arabic characters to about 60 different shapes. Table (1) gives a classification of the Arabic alphanumeric characters according to their features. These features are divided into stable and variable categories, and a different representation (and consequently a different classification strategy) is needed. In general, the number of secondary strokes and the number of dots and their relative positions are stable features of Arabic characters, while the structure of the main strokes is the most variant portion of the characters [8].

3.2 The Database

In our experiments of HAAC recognition, 28 alphanumeric characters are used. These characters are those with main stroke only (see Table 1). Ten different samples of HAAC written by 10 different people, where each sample consists of 28 alphanumeric characters were collected. The resulting database has 280 character patterns. These patterns are used as learning set. It is necessary to perform some additional preprocessing to normalize the shape of the characters. We scaled the bitmaps to fit into a square window. Some scaled samples taken from the database are shown in figure 3.

The database is used to generate epochs for the learning phase of both the BP and SOFM algorithms, where an epoch is defined as "one presentation of the entire set of 280 learning patterns". In epochs, the patterns from each person may be presented in a random ordering, otherwise, the system may tend to memorize the output for latest cases, forgetting some of the previously given patterns.

In this experiment, the collected patterns are photographed by digital scanner in order to expose it on the screen. The object here was to change the representation from bit maps to feature maps. This is achieved after the processes of simple scaling of the bit maps, to fit into fixed size window. The scaled alphanumeric characters are then processed by software to obtain a shadow projection value on segments bar mask (i.e., feature representation).

3.3. Shadow Codes for HAAC

Handwritten pattern recognition systems are known to perform more accurately and efficiently when input features are well selected and encoded. A particular encoder which satisfies these requirements is based on a shadow projection on 16-segment mask array as shown in Figure 4(a).

An input alphanumeric character is first normalized so that it extends the full height and width of the bar mask. A shadow projection operation is defined, which simultaneously projects a point into its three closest vertical, horizontal, and diagonal bars. A projected shadow turns on a set of bits distributed uniformly along the bar. After all the points on a character are projected, the number of on bits in each bar is counted. The character is represented by these 16 numbers, which we refer to as the shadow codes.

An example showing the encoding of the character "hah ح" is illustrated in figure 4(b). Code values ranging from 0 to 100 are normalized to the rang (0, 1) for the neural network.

4. Evaluation of the Learning Algorithms

The HAAC recognition problem which is outlined in the previous sections, is implemented on a microcomputer based system. The behaviors of BP and SOFM algorithms are simulated on P4, 2.4 GHz full cache processor PC using C^{++} language programs. The program runs under MS-DOS and was compiled with Borland C^{++}. After the training is completed, the network is required to recognize any input pattern (HAAC) out of the learning set patterns. Figure 5, shows four different samples of characters written by four different people used as test set. Then the recognition rate of the system is given as :

$$\text{Recognition rate } \Re = \frac{\Im}{\aleph} * 100 \qquad \text{.............................}4.1$$

Where $\Im$ is the number of correct recognition, and $\aleph$ is the total number of recognition.

4.1 Backpropagation (BP) Algorithm

The BP algorithm is used to train classification network with single hidden layer for HAAC. The network has 16 input neurons corresponding to 16 elements input (feature) vectors, and 28 output neurons, one for each character class.

The desired (target) vector contains exactly 27 zero's and a single one, whose location is a key to the recognized character. Output values in the range (0-1) is obtain experimentally. Therefore, the output with the largest value is chosen as a key to the recognized character.

This approach is based on a gradient search that may find a local minimum in the error surface instead of the desired global minimum in the error surface instead of the desired global minimum [11]. This problem corresponds to identifying two or more alphanumeric characters as one, and can be minimized by using multiple starts with different random weights and a low updating function to adapt weights.

In the learning phase, approximately 100 epochs are applied (see figure 3), for which the error is calculated and weights are adjusted repeatedly, until the error is reduced to 0.0001. The random weights and external input are typically initialized to small random values. This start the search in a relatively "safe" position. The random value generator is set in the interval (0.01 – 0.1) to prevent overflows. Furthermore the sigmoid function uses a steeper with a multiplicity of $\beta = 3$, which results in a better distinguishment of the highest output than the other outputs, producing output values in the range : 0.01 – 0.97.

Another important point is the number of neurons in the hidden layer. When 48 neurons are used instead of 64 neurons in the hidden layer, oscillation may occur at 60-th epochs during learning phase. Note that the number of hidden layer neurons should always must be less than the number of learning patterns, otherwise the network simply memorizes the learning patterns, resulting in poor generalization. Finally, the system is very sensitive to input pattern characteristic. If the examples of the learning set have many patterns with similar feature, the satisfactory recall rate degrades.

In order to demonstrate the ability of the system, the four different samples of HAAC shown in figure 5 are used as the test set. Table (2) shows the experimental results for BP, obtained by applying the above four samples, and the recognition rate calculation described in Eq (4.1). The parameter setting, that we have used in these experiments are as follows:

Neurons in the input layer :	= 16
Neurons in the hidden layer:	= 64
Neurons in the output layer:	= 28
Updating value:	= 0.02
Sigmoid gain:	= 3
Random generator:	0.01- 0.1
Momentum term:	= 0.8
Number of weights:	= 2944
Patterns processed in the learning phase:	= 28,000

The 28,000 patterns of the HAAC are needed approximately during the learning phase (as a rule of thumb [11]) to achieve acceptable level of recognition rate as shown in Table (2).

4.2 Self-Organized Feature Map (SOFM) Algorithm

In this experiment, the SOFM algorithm is used as a classifier for HAAC. It classifies a set of input characters in a number of disjoint classes in such a way that the input patterns within each class are all similar to one anther. The inputs to the system, and the learning set are as in the previous BP system.

The input HAAC patterns from the epoch, locate the neuron with best-matching weights to this input, and adapt weights in the topological neighborhood of this neuron, to increase the matching. In this procedure, learning stops when updating value decreases to zero, so that no more weights are changed, and the system is said to converge. So the updating value has to be chosen carefully, so that it must not decay to zero, before all learning set is exhausted. The neurons in the output layer of the system are arranged in a two dimensional grid

The neighborhood radius function is used as a means of clustering. At the beginning, the neighborhood covers all of the neurons in the output layer. However, when learning is repeated over the learning epochs, the neighborhood is forced to shrink, so that the clusters will start to separate from each other. At the end of learning, there shouldn't be any clusters which are overlapping topologically. And this shows that the neighborhood radius, number of neurons in the output layer and number of classes should be in agreement. If the neighborhood radius function is chosen to shrink very slowly, then different patterns may be clustered into overlapping clusters. If on the other hand, the neighborhood radius function is chosen to shrink very rapidly, even similar patterns belonging to one class may be clustered to small separate clusters. So the decreasing rate of the radius function, which is to be set experimentally, is a delicate point of this algorithm.

In the learning phase, approximately 100 epochs are applied as in the previous BP system. Table (2), shows the experimental results for SOFM, obtained by applying the 4 samples of test set shown in figure 5 and the recognition rate (see Eq. (4.1)). The parameter setting, that we have used in these experiments are as follows:

Neurons in the input layer :	*= 16*
Neurons in the output layer:	*= 1024*
Number of weights:	*= 17,408*
Patterns processed in the learning phase:	*= 28,000*

The 28,000 patterns of HAAC are used during the learning phase as in BP network to achieve a level of recognition rate shown in table (2).

4.3 Performance Comparison of the Learning Algorithms

Depending on the experimental results and statistics gathered by different handwritten styles, both algorithms produced satisfying results. According to the recognition rates and the parameter settings for both BP and SOFM, we can say that BP turned out to be more efficient than SOFM. If we compare both systems according to resource consumption, BP consumes approximately 92 neurons, whereas SOFM consumes approximately 1024 neurons. If a number of operations per pattern comparison is to be made then BP takes 3000 interconnects per pattern for the running phase and approximately 8816 interconnects per pattern for the learning phase where as SOFM takes 18,432 interconnects per pattern for the running phase and approximately 35,840 interconnects per pattern for the learning phase.

If a time comparison is to be made, then BP spends 600 seconds, in learning all 28 HAAC, while SOFM spends 37 minutes for the same task. The overall throughput including image acquisition, are approximately 10 patterns per second for BP, and 1.4 patterns per second for SOFM. These values are limited mainly by the scaling and shadow projection steps.

When the recognition rate is considered, both systems produced a satisfactory and approximately the same result in samples A,B, and C of test set (see figure 5). The low recognition rate result in sample type D is not unexpected, since we have supplied the networks such an information that the system couldn't distinguish the HAAC out of so many similar features.

5. Conclusions

This paper has successfully applied BP and SOFM algorithms to a large, complex task. The results appear to be the state of the art in HAAC recognition. It showed that the networks can be learned on a shadow codes of HAAC that have minimal preprocessing (as opposed to other elaborate feature extraction methods [13][14]). Because of the redundant nature of the data and because of the constraints imposed on the networks, the learning time was relatively short considering the size of the learning set. Scaling properties were far better than one would expect just from extrapolating results of BP and SOFM on smaller artificial problems.

Extension of the work is possible for the recognition of Arabic dotted characters. Information regarding stable features can group in a feature vector and combine with an exact matching classification criterion. Then applying the introduced method to identify the dotted characters, enables all the Arabic characters to be recognized.

References

1. R. Plamondom, 2000, On-line and off-line handwriting recognition: A comprehensive survey. IEEE Transaction on Pattern Analysis and Machine Intelligence, 22(1), pp. 212-218.
2. Amin A., 1997, Handbook of Character Recognition and Document Image Analysis". World Scientific Pub. Co.
3. Y. Al-Ohali et. al.,2000, Arabic Character Recognition: Progress and Challenges. Journal of King Saua University. 12, Comp & Inf. Sci. 85-116.
4. Zeki Khedher, G. Abandah, 2002, Arabic Character Recognition using Approximate Stroke Sequence. (http://www.almiskat.com/khedher/papers_e/paper_e13htm).
5. J. Cowell, 2000, Character Recognition of Arabic and Latin Scripts. Proceeding of the conference on Information Visualization, pp. 51-56.
6. P. Gallinari, 2003, An MLP architecture for offline handwritten digit recognition. Int'l J. Doc. Anal. Rec. 5(4).
7. M. S. Kamel, 1999, Multiple Classification Hierarchical Architecture for Handwritten Arabic Character Recognition. Pattern Analysis 6(5) pp. 2834-2838.
8. Rafid A. Khalil, 1997, A unified systolic architecture of artificial neural networks and applications to handwritten Arabic alphanumeric character recognition, Ph. D. Thesis, Univ. of Technology, Baghdad, Iraq.
9. M. A. Sharaidah, 2000. Recognition of Handwritten Arabic Characters via Neural Networks, M. Sc. Thesis, Jordan university.
10. Amin, 1997, Off line Arabic character recognition – a survey. In ICDAR 97 pp. 596-599.
11. D. Hush, and B. Horne, 1993, "Progress in Supervised neural networks". IEEE Signal processing Magazine, 8-39, January 1993.
12. M. Hamada, 2000, A hybrid handwritten word recognition using self-organizing feature map, discrete HMM, and evolutionary programming. In Int'l Joint Conference on Neural Networks, pp.515-520 .
13. H. Baasli, 2003, A noval feature extraction technique for the recognition of segmented hardwritten characters. In ICDAR pp. 137,141.
14. Amin, 1998, Preprocessing and Structural Feature Extraction for a Multi-font Arabic/ Persian OCR. Proceedings of 5th International Conference of Document Analysis and Recognition, vol. 1, pp.145-158.

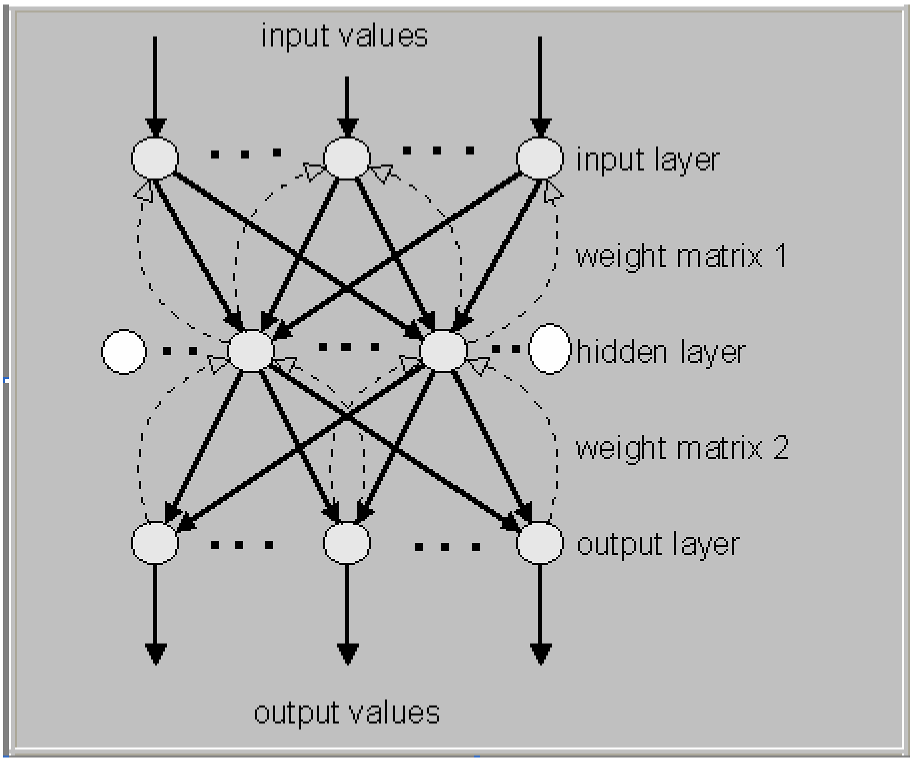


Figure 1. The network topology used in BP algorithm.

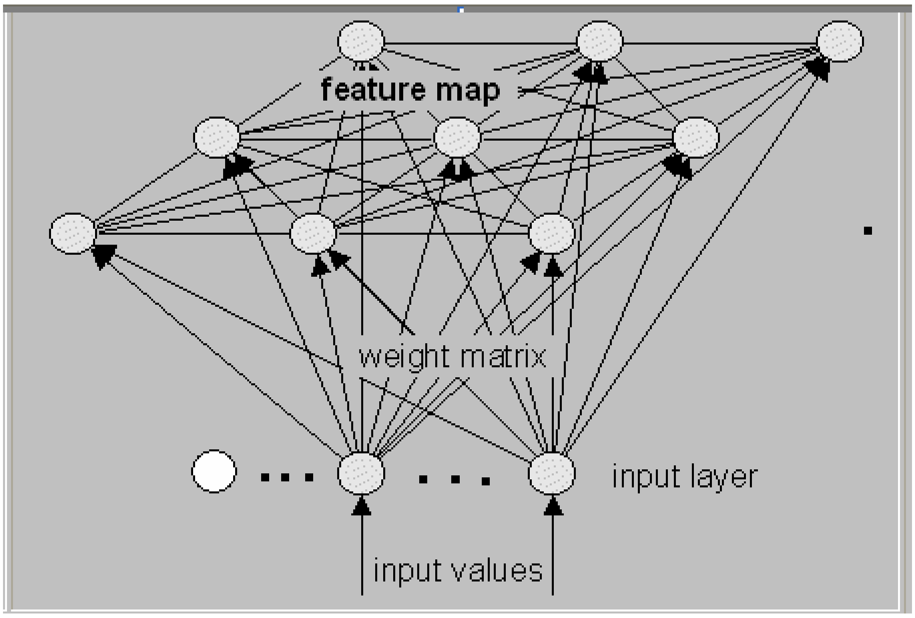
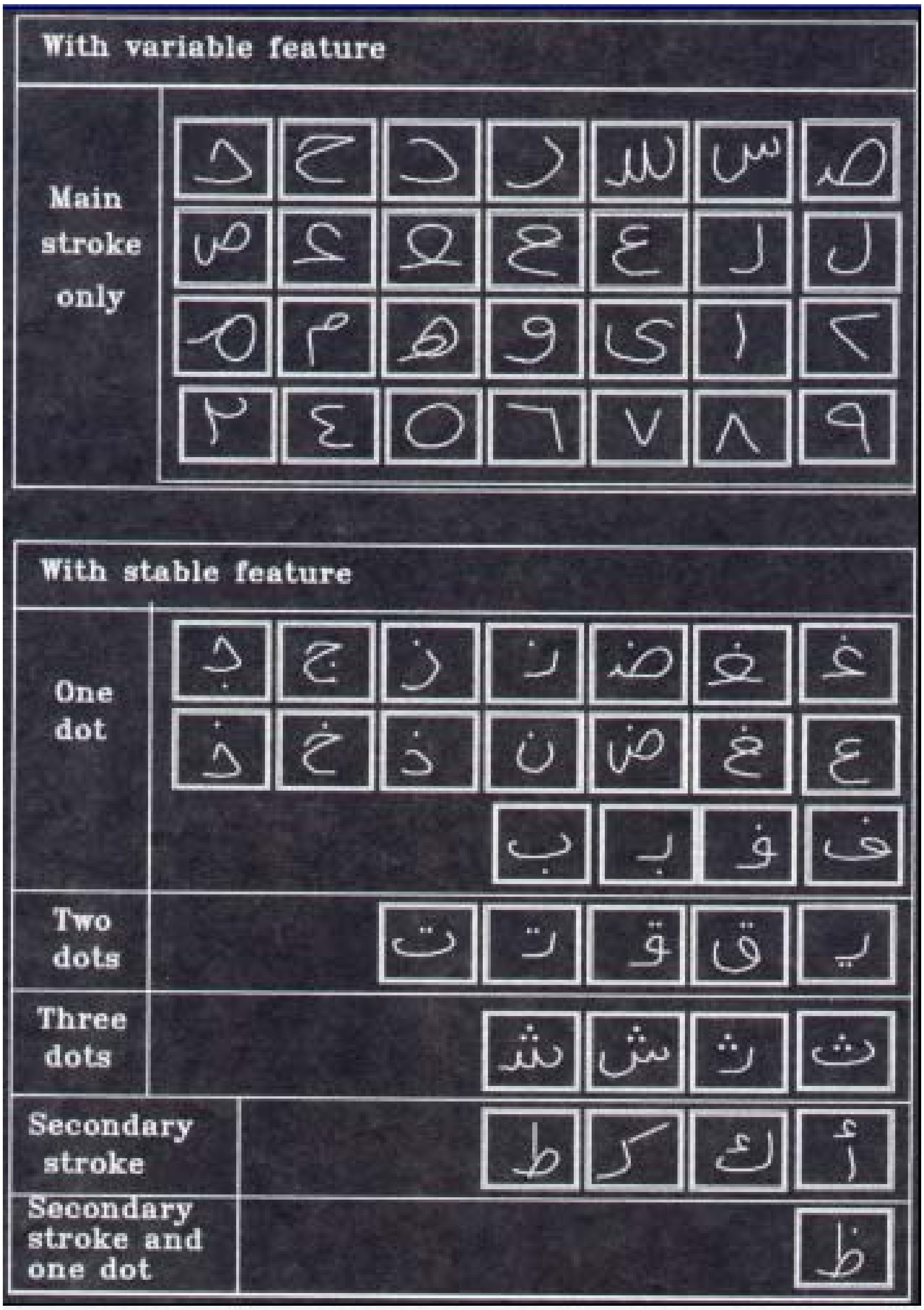


Figure 2. The network topology used in SOFM algorithm.

Table 1. Arabic alphanumeric character classification


With variable feature
Main stroke only
With stable feature
One dot
Two dots
Three dots
Secondary stroke
Secondary stroke and one dot

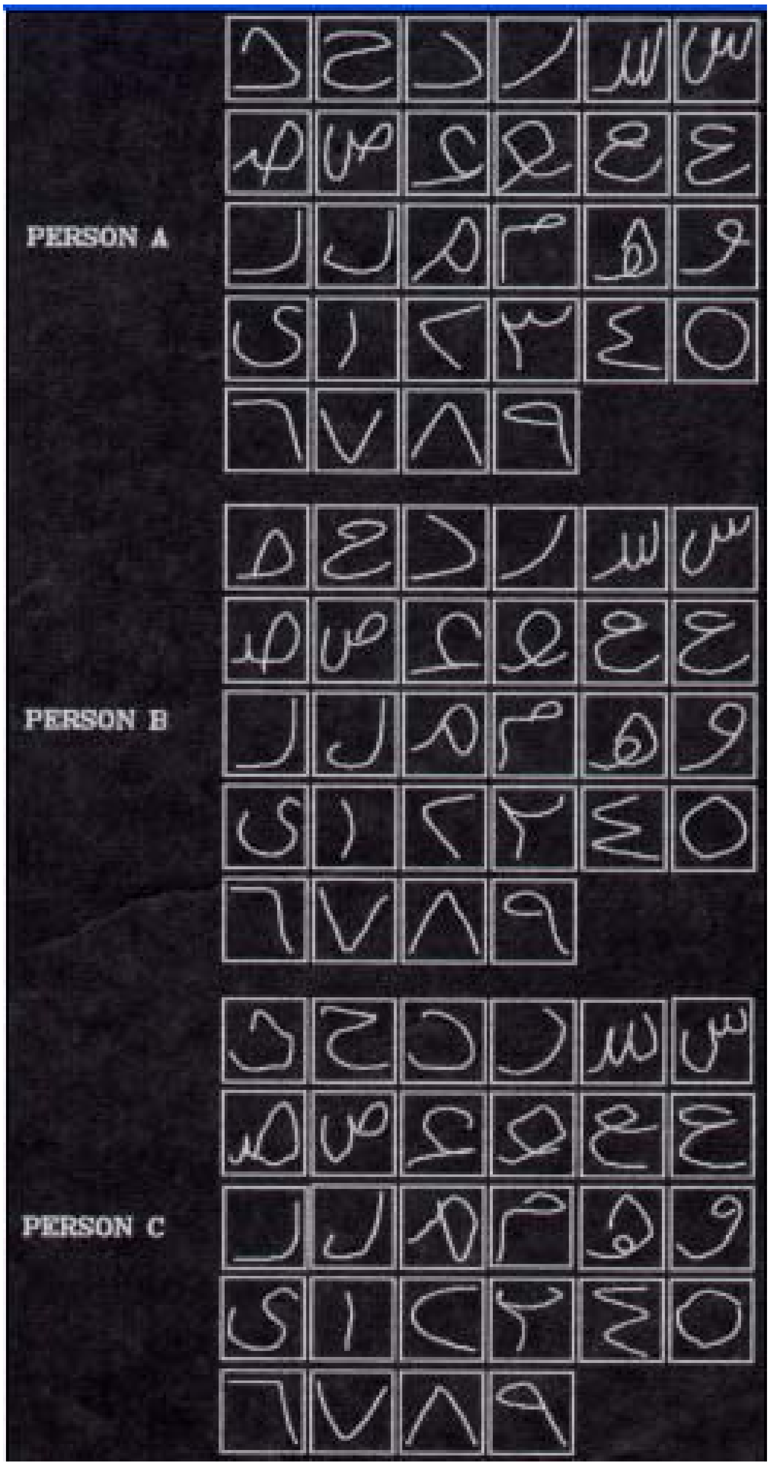
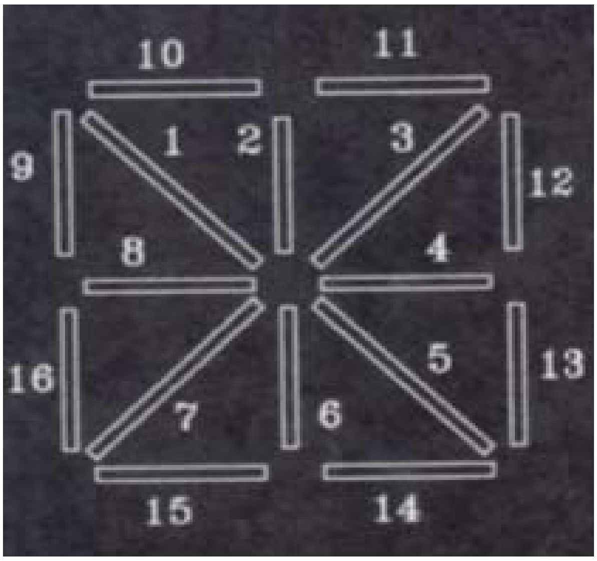
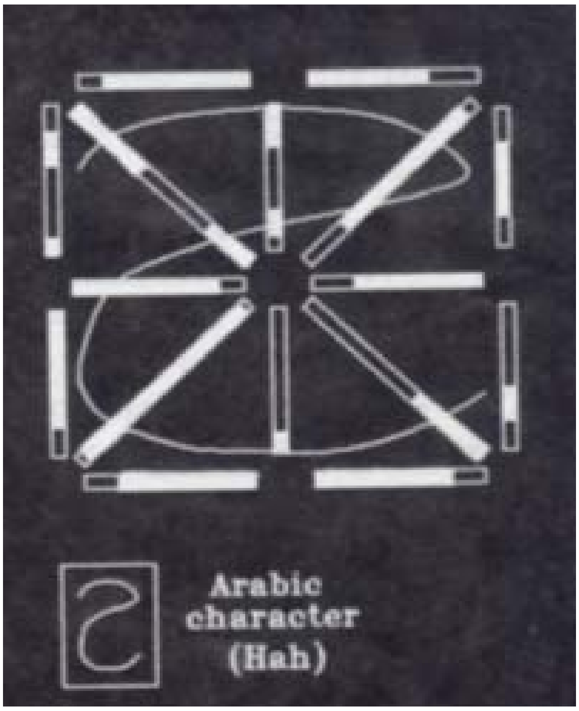


Figure 3. Some samples of Arabic alphanumeric character written by different people used to train BP network.



(a)



(b)

Figure 4 (a) 16-segment bar mask used for HAAC recognition.
(b) Illustration of encoding the character (Hah ح), with shadow projections shown in white.

The shadow code obtained by scanning bars in sequence is (69, 43, 71, 78, 42, 20, 94, 86, 40, 86, 75, 56, 26, 80, 80,76). Each number represents total length of the corresponding Shadow.

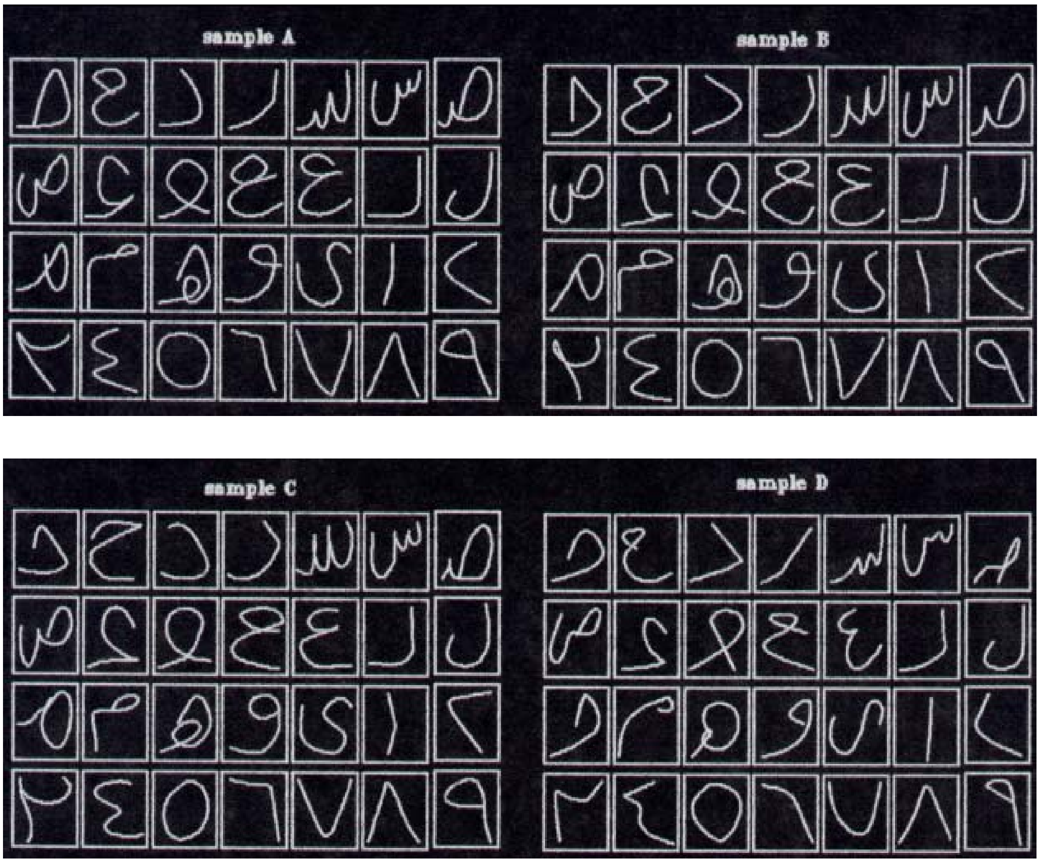


Figure 5. Four different samples of HAAC written by four people (samples A-D) used as test set for BP algorithm.

Table 2. Experimental results of BP and SOFM algorithms applied to HAAC recognition problem.

Test Set (HAAC Samples)	BP Algorithm Recognition Rate %	SOFM Algorithm Recognition Rate %
A	80	88
B	72	83
C	92	96
D	33	45