Proceedings of the 2002 International Arab Conference on Information Technology (ACIT'2002). Doha, Qatar. December 2002.

ABJAD: AN OFF-LINE ARABIC HANDWRITTEN RECOGNITION SYSTEM

RAMZI AHMED HARATY and HICHAM EL-ZABADANI
Lebanese American University
P.O. Box 13-5053 Chouran
Beirut, Lebanon 1102 2801
Phone: 961 1 867621 ext. 1285
Fax: 961 1 876098
Email: rharaty@lau.edu.lb

ABSTRACT

In this work we present a system for the recognition of handwritten Arabic text using neural networks. This work builds upon previous work done by [1]. That part dealt with the vertical segmentation of the written text. However, faced with some problems like overlapping characters that share the same vertical space, we tried to fix that problem by performing horizontal segmentation. In this research we will use two basic neural networks to perform the task; the first one to identify blocks that need to be horizontally segmented, and the second one to perform the horizontal segmentation. Both networks use a set of features that are extracted using a heuristic program. The system was tested with over 1500 characters (each character has on average about 50 rows) and the rate of recognition obtained was over 90%. This strongly supports the usefulness of proposed measures for handwritten Arabic text.

Keywords: Arabic Text, Neural Networks, and Recognition System.

1. INTRODUCTION

Whether we like it or not, the world is undergoing an information technology revolution. People are forced into contact with computers and our dependence upon them continues to increase. That is why computers should be easier to use. Nowadays, as most of the world's information processing is done electronically, it becomes more efficient if we make the transfer of information between people and machines more simple and reliable [2][3][4].

The recognition of Arabic characters represents a significant challenge due to the large set of features in the Arabic script. Each of the 28 letters has on average four shapes depending on its position within a word (start, middle, end or isolated; see Table 1). Also, because Arabic words are written cursively from right to left and contain several connected letters, character segmentation is necessary before starting the recognition phase. Some words contain broken parts because some characters cannot be connected to the others. Vowel diacritics and writing style add another degree of complexity to the recognition task.

In this paper, we are working on the second part of a system that is divided into three parts. The first part deals with the vertical segmentation of Arabic handwritten text. It was done and tested by [1]. The second part which is the main topic of this paper, deals with the horizontal segmentation of the words that could not be segmented vertically. The third part is still being worked on, deals with the classification of all characters produced by the first two parts.

ABJAD uses two multi-layer perceptron neural networks to perform the horizontal segmentation. The first one determines which block should be horizontally segmented, and the second one determines which rows are valid segmentation points; and thereby, will finish the horizontal segmentation.

The input of the second part of ABJAD is the output of the first part (see Figure 1a), which is formed of both single characters and blocks of overlapping characters. The second part will identify each block as separate character or a block that needs to be segmented horizontally (see Figure 1b).

The remainder of this paper is divided into 4 sections. Section 2 describes the proposed approach used to identify ligature blocks. The design of the heuristic and neural network components is also presented. Section 3 describes the proposed approach used to segment horizontally Arabic handwritten text. Section 4 presents the experimental results of this approach and finally a conclusion is drawn in section 5.

2. LIGATURE IDENTIFICATION

Before attempting to segment horizontally, the system should be able to recognize every single block as either ligature or character block. However, there are several major problems related to ligature recognition:

2.1 OBSTACLES

- **Variety in Size:** The same character may be written in different sizes without changing the meaning of the character.

- **Variety in Shape:** Characters may vary in shape, i.e., in line thickness, color, or stroke direction. This would cause some problems like touching characters, filled holes, or broken characters.

- **Variety in Style:** Every writer has his own writing style, i.e., different writers or the same writer in different conditions could write the same character in a different style.

- **Similarity in Block Shapes:** Sometimes two different blocks, i.e., ligature and character block, could have the same shape. This may lead to some errors in the recognition process.

2.2 PROPOSED TECHNIQUE

Before attempting to start ligature recognition, there are some steps that need to be done like data collection, data analysis, scanning, binarization, and finally block classification.

2.2.1 DATA COLLECTION

Samples were randomly collected from various students at the Lebanese American University and Gezairi Transport Company. People were asked to write character and ligature blocks. These samples were then scanned at 100 pixels per inch, and saved in monochrome Windows Bitmap format.

2.2.2 FEATURE EXTRACTION

This stage transforms the block to be recognized into a sequence of features that will be used as the input of the neural network in a later stage. In order to decide whether a block is a ligature or character one, the list of features listed in Table 2 were used.

2.2.3 VERIFICATION USING ANN

To find the optimum ANN architecture to solve this problem, various networks with different types, number of hidden layers and processing elements (PEs) per each layer were tried. The ANN with the smallest number of PEs, minimum estimated generalization error, and that learned best to identify correct segmentation points was chosen. The best ANN architecture found was a multilayer perceptron that consists of 3 layers. With 35 inputs, 1 output and 1 hidden layer that contains 10 PEs, the network was built. The 35 inputs were the features that we collected in previous sections and the output was the decision of the network about whether the block is a ligature or character block (Table 3).

3. HORIZONTAL SEGMENTATION

There are no major differences between horizontal and vertical segmentation, both perform the same tasks. However, instead of extracting features from the original Character Block (BC), features are extracted from a Transposed Character Block (TBC) as shown in figure 2.

3.1 FEATURE EXTRACTION

Table 4 shows the major features used in horizontal segmentation. Holes, corner points, fork points, and end points were part of the features.

3.2 ANN ARCHITECTURE

The ANN architecture used was the same as the one used for ligature identification with some few modifications. Since horizontal segmentation needs more feature than ligature identification, the ANN was more complex. In the next few sections we will discuss only the modifications. The best ANN architecture reached consisted of 52 inputs, 1 output, and 2 hidden layers. The 52 inputs were feature attribute of a pre-segmentation point and the output was the validity of the point. The ANN architecture is summarized in Table 5.

4. EXPERIMENTAL RESULTS

4.1 LIGATURE IDENTIFICATION SEGMENTATION RESULTS

The testing set for ligature identification was about 2250 exemplars. About 42% of the testing set consisted of ligature blocks, and the rest were character blocks. The output range of the ANN was between -0.9 and 0.9. A positive value indicated that

a block is a valid ligature block; a negative value indicated that a block is a character block.

An algorithm checked the results and defined the identification of blocks as ligature or characters blocks. As shown in figure 17, 79% of all the blocks were identified correctly (57% consisting of character blocks identified correctly and 22% consisting of ligature blocks identified correctly).

The recognition rate in ligature identification would be 79% according to figure 3, while the error rate would be 22%. Table 6 describes the number of ligature blocks identified correctly and incorrectly and the number of character blocks that were identified correctly and incorrectly.

4.2 HORIZONTAL SEGMENTATION EXPERIMANTAL RESULTS

The testing set for ligature identification was about 13,000 exemplars. About 10% of the testing set consisted of valid segmentation points, and the rest were invalid segmentation points. As we mentioned above, the output range was between -0.9 and 0.9. A positive value indicated that a point is a valid segmentation point; a negative value indicated that a point is an invalid segmentation point. As shown in figure 18, 91% of all points were identified correctly as valid or invalid points (88% consisting of invalid points identified correctly and 3% consisting of valid points identified correctly).

The recognition rate in horizontal segmentation would be 91% according to figure 4, while the error rate would be 9%. Table 7 describes the number of valid segmentation points identified correctly and incorrectly and the number of invalid segmentation points that were identified correctly and incorrectly.

Out of 1300 segmentation points that should be identified as valid, the ANN identified 390, which is about 1/3 of the total number of valid segmentation points.

Figure 5 shows the range of segmentation points that should be identified as valid. It is enough that at least one point out of all segmentation points included in this range would be identified by the ANN as valid segmentation point. This makes the result of the ANN a good result, since 1/3 of the total valid segmentation points were identified by the ANN. This will guarantee that in each ligature block, at least one valid segmentation point will be identified by the ANN, which is enough to cut the ligature into two separate characters.

4.3 COMPARISON OF SEGMENTATION RESULTS

Many researchers have used various techniques for the segmentation of characters in handwritten words. Segmentation accuracy rates of above 90% were achieved by [5]; however, the authors were only dealing with printed Latin alphanumeric characters. [6] obtained segmentation accuracies of 83% for handwritten zip codes (no alphanumeric). [7] achieved an 85.7% accuracy using a heuristic algorithm for the segmentation of words on 50 envelopes from real mail pieces. Finally, experiments conducted by [8], segmenting cursive handwriting produced a 75.9% accuracy rate using an ANN-based method.

On average our segmentation accuracy using the neuro-conventional technique was about 91% for horizontal segmentation and 80% for ligature identification. Table 8 summarizes the results obtained by various researchers.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented two main Artificial Neural Networks. The first has the job of identifying ligature blocks. However, the second is responsible of horizontally segmenting all ligature blocks identified by the first one. The segmentation phase proved to be successful in both networks. The first one, dealing with ligature identification, had a recognition rate of 80%. However, the other one, which is responsible of the horizontal segmentation process, had a recognition rate of over 90%.

In future work, ligature identification and horizontal segmentation could be somehow improved. For example, extracting more features could improve both ligature identification and horizontal segmentation. Moreover, ANN's could be trained more so that it will give better and accurate results.

Finally, a classification system should be integrated into the overall system to obtain a complete Arabic handwritten recognition system.

REFERENCES

[1] R. A. Haraty and A. Hamid, *Segmenting Handwritten Arabic Text*, ACIS International Journal of Computer and Information Science (IJCIS), Volume 3, Number 4. December, 2002.

[2] B. Bos, and A. Van Der Moer, *The Bakunin Project and Optical Character Recognition*, Proceedings of OCRHD, 1993, pp. 11-15.

[3] J. B. Bellegarda, D. Nahamoo, K. S. Nathan, and E. J. Bellegarda, *Supervised Hidden Markov Modeling for On-Line Handwriting Recognition*, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Volume 5, 1994, pp. 149-152.

[4] M. Cheriet, and C. Y. Suen, *Extraction of Key Letters for Cursive Script Recognition*, Pattern Recognition Letters 14:, 1993, pp. 1009-1017.

[5] S. W. Lee, D. J. Lee, and H. S. Park, *A new Methodology for Gray-Scale Character Segmentation and Recognition*, IEEE Transaction on Pattern Analysis and Machine Intelligence, 18, 1996, pp. 1045-1051.

[6] S. N. Srihari, V. Govindaraju, and A. Shekhawat, *Interpretation of Handwritten Addresses in US Mail Stream*, Proceedings of ICDAR, 1993, pp. 291-294.

[7] K. Han and I. K. Sethi, *Off-Line Cursive Handwriting Segmentation*, Proceedings of ICDAR, Montreal, Canada, 1995, pp. 894-897.

[8] B. Eastwood, A. Jennings, and A. Harvey, *A Feature Based Neural Network Segmenter For Handwritten Words*, Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA), Gold Coast, Australia, 1997, pp. 286-290.

[9] M. Blumenstein and B. Verma, *A Segmentation Algorithm Used in Conjunction with Artificial Neural Networks for the Recognition of Real-World Postal Addresses*, Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA), Gold Coast, Australia, 1997, pp. 155-160.

Table 1. Different positions of a character.

Isolated	Start	Middle	End
م	مـ	ـمـ	ـم

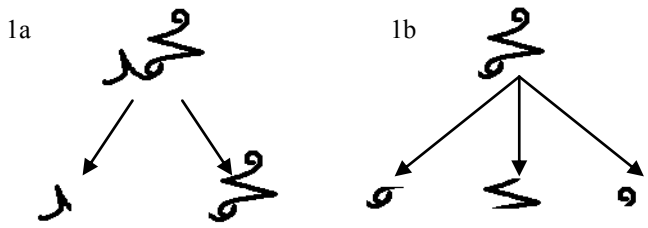

Figure 1: (1a) Input of the first part and its output. (1b) Input of the second part and its output.

Table 2 Major features extracted for each block.

Feature	Attribute
Width and Height	
Center of Gravity	
Black Pixel Density	Total Col. Density Minima
	Total Row Density Minima
	Total Col Density Maxima
	Total Row Density Maxima
Transitions	Row Max. Transitions
	Column Max. Transitions
Junctions	Total Junctions
Loops	Max. Loop Density
	Max. Loop Transition
End Points	Total End Points
Turning Points	Total Turning Points
Contours	Total Upper Contour
	Total Lower Contour

Table 3 Architecture of the ANN.

	PEs	Transfer Function
Input Layer	35	Linear
Hidden Layer 1	10	Tanh
Output Layer	1	Tanh

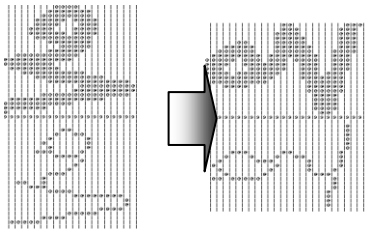

Figure 2 Ligature block and its corresponding transposed block.

Table 4 Major features extracted for each column of BC.

Feature	Attributes

Image width and height	
Black pixel density	Black pixel density / width
	Density minima
	Density maxima
Transitions	Number of transitions crossed
Holes	Number of holes crossed
	Total hole densities / width
Endpoints	Number of endpoints crossed
Corner points	Number of corners crossed
Fork points	Number of fork points crossed
Relative index of row in image	
Upper and lower contours	Upper and lower contour index / width
	Upper and lower contour minima or maxima
Feature relationships	Index of nearest left and right feature / 100

Table 5 Architecture of the ANN.

	PEs	Transfer Function
Input Layer	52	Linear
Hidden Layer 1	30	Tanh
Hidden Layer 2	15	Tanh
Output Layer	1	Tanh

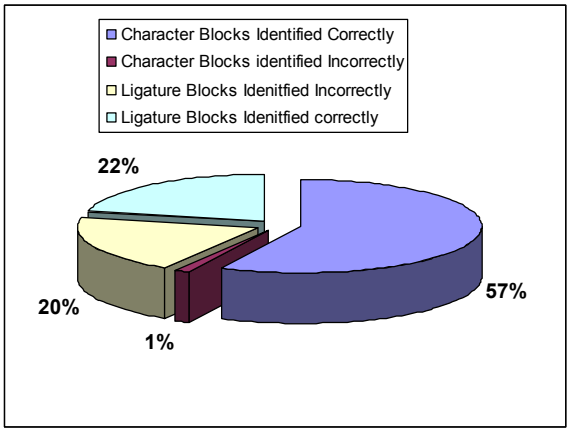

Figure 3 Percentages of the ligature identification ANN results.

Table 6 Number of correctly and incorrectly identified blocks.

	Correctly Identified	Incorrectly Identified	Total
Ligature Blocks	**495**	**450**	945
Character Blocks	**1283**	**22**	1305
Total	1778	472	2250

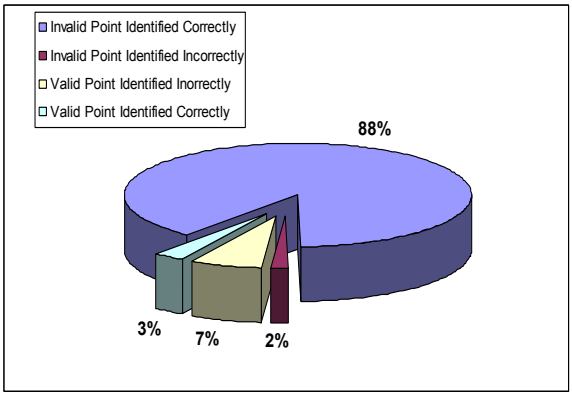

Figure 4 Percentages of the horizontal segmentation ANN results.

Table 7 Number of correctly and incorrectly identified points.

	Correctly Identified	Incorrectly Identified	Total
Valid segmentation points	**390**	**910**	1300
Invalid segmentation points	**11440**	**260**	11700
Total	11830	1170	13000

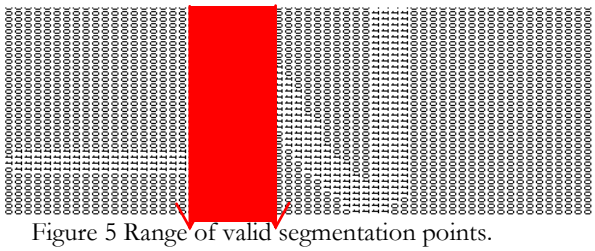

Figure 5 Range of valid segmentation points.

Table 8 Comparison of segmentation results in the literature.

Author	**Segmentation accuracy [%]**	**Data set used**	**Method used**
Blumenstein and Verma [9]	81.21	Griffith University Latin handwriting database	Neuro-conventional method
Eastwood et al. [8]	75.9	Cursive Latin handwriting from CEDAR database.	ANN-based method
Han and Sethi [7]	85.7	Latin handwritten words on 50 real mail envelopes	Heuristic algorithm
Lee et al. [5]	90	Printed Latin alphanumeric characters.	ANN-based method
Srihari et al. [6]	83	Handwritten zip codes (no alphanumeric)	ANN-based method