A Data Base for Arabic Handwritten Text Recognition Research

Somaya Al-Ma'adeed, Dave Elliman, and Colin Higgins
School of Computer Science and Information Technology, University of Nottingham, UK

Abstract: *In this paper we present a new database for off-line Arabic handwriting recognition, together with several preprocessing procedures. We designed, collected and stored a database of Arabic handwriting (AHDB). This resulted in a unique databases dealing with handwritten information from Arabic text, both in terms of the size of the database as well as the number of different writers involved. We further designed an innovative, simple, yet powerful, in place tagging procedure for the database. It enables us to extract at will the bitmaps of words. We also built a preprocessing class, which contains some useful preprocessing operations. In this paper, the most popular words in Arabic writing were found for the first time using a specially designed program.*

Keywords: *Arabic, handwriting, recognition, database, preprocessing, cursive script.*

Received April 15, 2003; accepted July 18, 2003

1. Introduction

Offline handwriting character recognition is the automatic transcription by computer, where only the image of that handwriting is available. Much work has been done on the recognition of Latin characters, both of separated and of cursive script. Work on recognizing Arab characters is still limited. The importance of recognizing Arabic characters also applies to non-speaking Arabic countries such as Farsi, Curds, Persians, and Urdu-speaking who use the Arabic characters in writing, although the pronunciation is different. Among those who have worked in this field are the following. Abuhaiba *et al.*, dealt with some problems in the processing of binary images of handwritten text documents [1]. Almuallim and Yamaguchi proposed a structural recognition technique for Arabic handwritten words [3]. A look-up table is used for the recognition of isolated hand-written Arabic characters [18]. Amin and others proposed a technique for the recognition of hand-printed Arabic characters using neural networks [4]. Obaid introduced Arabic handwritten character recognition by neural networks [15]. Saleh *et al.* describe an efficient algorithm for coding handwritten Arabic characters [19].

It is noted that most of the above work is done assuming that the Arabic handwritten word is already segmented into separated characters before recognition. Also, all of the work done on handwritten Arabic data was done with little training and test data, written by a single writer.

In this paper, we deal with the design, storage, and retrieval steps, as well as the preprocessing of offline handwritten Arabic words. Off-line Arabic character recognition involves many steps that cannot be separated from each other. In this paper the first organized database for Arabic handwritten text and words collected are presented.

An important application aspect of handwriting recognition is in domains such as bank cheques [8] and postal addresses [5]. Here there is no control over the author, writing instrument, or writing style. For example, an arbitrary handwritten word might be produced by a felt pen and could include isolated, touching, overlapping characters, cursive fragments, or fully cursive words [10]. However, these difficulties are offset by the constraint that the input words come from a fixed vocabulary.

A standard database of images is needed to facilitate research in handwritten text recognition. Some of the previous databases are summarized in [14, 22]. In [11, 13, 23] one can find databases for English off-line handwriting recognition. For machine-printed Arabic, the Environmental Research Institue of Michigan (ERIM) has created a database of machine-printed Arabic documents. These images are extracted from typewritten and typeset Arabic books and magazines [20].

Applications, which usually include some pattern recognition, require the use of large sets of data. Since there is a lack of Arabic databases available, and in order to train and test systems that are able to recognize unconstrained handwritten Arabic text, the AHDB database was built**.** This Data Base contains Arabic Words and texts written by a hundred different writers. The following sections contain steps on building such Data Base. Because the AHDB contains the most popular written Arabic words, the next

section will discuss in details how this was created. The following sections will discuss how the forms are designed and scanned. There are different approaches to form dropout. Some approaches use separate cleaning steps, while others use combined cleaning methods for both foreground and background [7]. In [6], they mentioned three common approaches to form dropout: symbolic subtraction of an image, color filtering, and thresholding. The approach we propose here in this paper is dropout by color filtering using hardware (optical filtering), which is faster than the other techniques and more accurate than dropout by symbolic subtraction. Sections 4 and 5 will discuss how the AHDB is stored and sorted in separated directories for easier data retrieval.

2. Arabic Words Counting

This step was aimed to find the most popular words in Arabic writing. First, the Arabic texts were copied from several sites from the Internet with different contents and about different subjects. Secondly, a program was written to count the repeated words in text files. The text files contained more than thirty thousand words. Finally, the words were summed up and sorted using Microsoft Excel worksheet. From the performed experiment, the twenty most used words in written Arabic were sorted and illustrated in Table 1. From the table you can see that the most popular words in Arabic writing are different from those in English. As in English the most popular word is "the" while in Arabic the most popular word is "in".

Table 1. The twenty most used words in written Arabic.

	Arabic Word	Meaning in English
1	??	In
2	h	From
3	•′	Is
4	kA	On
5	ℓ	To
6		That
7	»X	That
8	h	About
9	a	With
10	K	Which
11	?X	That
12	??	Or
13	???	Was
14	)	Finish
15	??	He
16	t	No
17	I	She
18	?′	God
19	W	Servant
20	f	Before

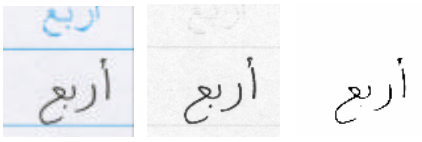
3. Form Design

The form was designed in six pages. The first three pages were filled with ninety-six words, sixty-seven of which are handwritten words numbers that can be used in handwritten cheque writing. The other twenty-nine words are from the most popular words in Arabic writing determined in section 2. The fourth page is designed to be filled with three sentences of handwritten words, numbers, and quantities that can be written on cheques. The fifth page is lined, to be filled by the writer in freehand on any subject of his choice.

The color of the forms was chosen to be light blue and the script written with any black ink. The scanner software can mask blue, green, and red. Thus one can print forms in green and filled with blue, red or black ink with the same result as blue form with black writing. In the first three pages, the spaces for handwritten words are equal so there is no restriction on the writer concerning the length of the writing.

4. Forms Scanning

Several ways of scanning were examined as following: the first forms were scanned in color mode (600 dpi), and then the blue channel mode was applied using photo shop. The images still had a grey shadow in place of the blue color in forms. Then a stamp filter was applied with good results (Figure 1). However, there are two disadvantages of color scanning: the first being that it is time-consuming. The second being that it takes up unnecessary storage space. Finally, forms were scanned in black and white using the blue channel as a mask (hardware mask). The result looked like the result in the first step but was much faster and used less memory (Figure 2). One hundred and five forms were scanned using the Hewlett Packard 6350 scanner. The images were scanned at 600 dpi. About one minute was required for scanning and color drop out for each image.



(a) (b) (c)

Figure1. Color drop out using software: (a) scanned image, (b) after applying blue channel mode, and (c) the image after stamp filter.

Figure 2. Color drop out using hardware.

5. Data Storage

Every word image is saved with its name and a number to identify the writer. Common file types in bitmap format are JPEG, GIF, TIFF and WMF. The TIFF format was chosen for the AHDB. The reason being that TIFF format can store complex information for the

CMYK color model and use the JPEG compression technique, making this one of the most robust and well-supported image formats available [9].

For easier retrieval of hand written images, the Arabic handwritten data was sorted and saved in five subdirectories as following:

1. *wrd_no:* contains words used for numbers and quantities in cheque filling (Figure 2).
2. *Wrd_mst:* contains the most popular words in Arabic writing (Table 1).
3. *Chq:* contains sentences used in writing cheques in Arabic words (Figure 3).
4. *Page:* contains free handwriting pages in any area of writer interest (Figure 4).
5. *Form_Wrd:* contains the first three pages of the forms. The first page is stored as "the_number_of_the_form_a". For example; the first page of the first form (for the first writer) stored as 001_a. The second page stored as "the_number_of_the_form_b". For example; the second page of the first form stored as 001_b. And third page stored as "the_number_of_the_form_c". For example; the third page of the first form stored as 001_c.

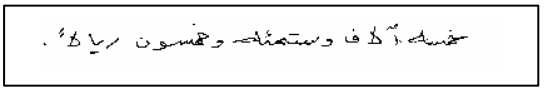


Figure 3. An example of sentences used in writing cheques in Arabic.

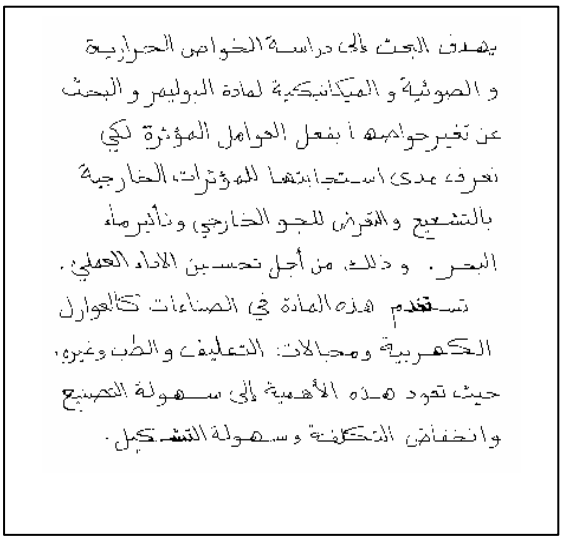


Figure 4. An example of free handwriting.

6. Data Retrieval

As we mentioned earlier in section 5 the data was stored in TIFF format. For retrieving the images, the system used lizard's TIFF library for Java [12]. For counting the number of segments (or blobs as it called in our system) and their dimensions, the system used and developed "Blubs class", which is fast as it deals with the stored tif raw data in the memory as shown in Figure 5.

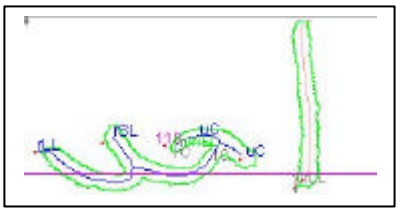


Figure 5. The blubs of the Arabic word "ahad".

7. Preprocessing Steps

The main advantage for preprocessing the handwritten word image is to organize the information to make the task of the recognition simpler. The main part of the pre-processing stage is normalization, which attempts to remove some of those variations in the images, which do not affect the identity of the word [21]. This system incorporates normalization for each of the following factors stroke width, slope, and height of the letters (see Figure 1). The normalization task reduces each word image for one consisting of vertical letters of uniform height on horizontal base line, and made of one-pixel-wide strokes. In this system, the word image is loaded and cropped. Then the slant and slope of the word is corrected and thinned. For representing the useful information contained in the image of the word features are calculated [2]. Then the word will be segmented into frames, so that the previous features in these frames could be distributed.

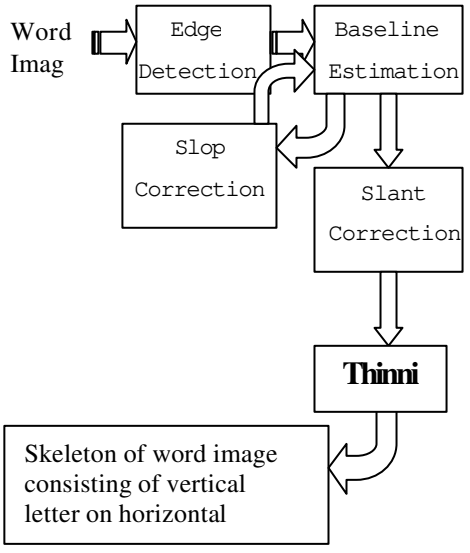


Figure 6. The preprocessing operations.

In the Java implementation of our system, every preprocess operation has a separate class. In the following subsections, the steps involved in the Arabic handwritten preprocessing that used in this research are discussed in detail.

7.1. Slope Correction

The slope, defined as the angle of the base line of a word that is not written horizontally. The heuristic, used for the base line estimation, consists of three main steps [5]:

1. Calculation of the Vertical Density Histogram

By counting the number of black pixel in each horizontal line in the image, as it is seen in the procedure Calculate the vertical histogram. Then the base line estimation follows by rejecting the part of the image likely to be a hooked descender. Such a descender is indicated by the maximum peak in the vertical density histogram. Finally the slope correction procedures are calculated by the following procedure:

2. Calculate the Slope

As following:

a. Find the lowest remaining pixel in each vertical scan line.
b. Retain only the points around the minimum of each chain of pixels and discard the points that are too high.
c. Find the line of best fit through these points.

3. Slope Correction

a. The image of the word is straightened to make the baseline horizontal by the application of a "Shear" transform parallel to the y-axis.
b. The baseline, height, and bounding rectangle of the cropped image are re-estimated, under the assumption that it is now horizontal.

7.2. Slant Correction

The slant is the deviation of strokes from the vertical axis, varying between words and between writers. The slant of a word is estimated by finding the average angle of near-vertical strokes [17]. This is calculated by finding the thinned strokes, using a thinning filter. The mode orientation of those thinned strocks close to the vertical is used as an overall slant estimate.

7.3. Thinning

Numerous algorithms have been proposed for thinning (also called skeletonizing) the plane region. This system uses a Zhang-suen/Stentiford/Holt combined algorithm for thinning binary regions [16].

7.4. Width Normalization

Before finding handwriting features for each word, the original word image can be normalized and encoded in a canonical form, so that different images of the same word are encoded similarly. The normalization task will reduce each word image to one consisting of vertical letters of uniform height on horizontal base line, and made of one-pixel-wide strokes. The width will be normalized to 64 width.

8. Conclusion

We have built the AHDB database. This contains Arabic Words and texts written by a hundred different writers. The AHDB database contains words used for the numbers and quantities in cheques filling. Also it contains the most popular words in Arabic writing (counted for the first time in this paper). It also contains sentences used in writing cheques in Arabic words. Finally, it contains free handwriting pages in any area of writer interest. This database is meant to provide a training and testing set for Arabic text recognition research. At the end, there are some useful preprocessing operation carried out on the AHDB.

References

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

[2] Almaadeed S., Higgens C., and Elliman D., "A New Preprocessing System for the Recognition of Off-line Handwritten Arabic Words," *IEEE International Symposium on Signal Processing and Information Technology*, Egypt, December 2001.

[3] Almuallim H. and Yamaguchi S., "A Method of Recognition of Arabic Cursive Handwriting," *IEEE Trans. Pattern Analysis and Machine Intelligence (PAMI-9)*, pp. 715-722, 1987.

[4] Al-Sadoun A. H. and Fischer S., "Hand-printed Character Recognition System Using Artificial Network," *Pattern Recognition*, vol. 29, no. 4, pp. 663-675, 1996.

[5] Chen M. Y., Kundu A., and Zhou J., "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 481-496, 1994.

[6] Cracknell C. and Downton A. C., "A colour approach to form dropout," *Downton A. and Impedovo S. (Eds.) in Progress in Handwriting Recognition*, World Scientific, UK, 1997.

[7] Downton A. and Impedovo S., *in Progress in Handwriting Recognition*, World Scientific, UK, 1997.

[8] Freitas C. O., El Yacoubi A., Bortolozzi F., and Sabourin R., "Brazilian Bank Check Handwritten Legal Amount," *in Proceedings of the XIII Brizilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'00),* Brazil, 2000.

[9] Howell D., "Getting to Grips with Graphic File Format," Computer Publishing, issue 9, 2000.
[10] Hull J. J., "A database for handwritten text recognition research," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550-554, 1994.
[11] Johansson S., Leech G. N., and Goodluck H., *Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for use with Digital Computers,* Department of English, University of Oslo, Norway, 1978.
[12] LizardWorks, On-line reference, *http://www.lizardworks.com/java.html*, 2000.
[13] Marti U. and Bunke H., "A full English sentence database for off-line handwriting recognition," *in Proceedings of the 5th Int. Conf. on Document Analysis and Recognition (ICDAR'99)*, Bangalore, pp. 705-708, 1999.
[14] Nagy G., "At the Frontiers of OCR," *in Proceedings of IEEE*, vol. 7, pp. 1093-1100, 1992.
[15] Obaid A. M., "Arabic Handwritten Character Recognition by Neural Nets," *Journal on Communications*, vol. 45, pp. 90-91, 1994.
[16] Parker J. R., *Algorithms for Image Processing and Computer Vision*, John Wiley & Sons Inc., USA, 1997.
[17] Rafuel C. and Woods R. E., *Digital Image Processing*, Addison, USA, 1992.
[18] Saadallah S. and Yacu S., "Design of an Arabic Character Reading Machine," *in Proceedings of Computer Processing of Arabic Language*, Kuwait, 1985.
[19] Saleh A., "A Method of Coding Arabic Characters and it's Application to Context-free Grammar," *Pattern Recognition Letters*, vol. 15, issue 12, pp. 1265-1271, 1994.
[20] Schlosser S. G., "ERIM Arabic Document Database," On-line reference: *http://documents.cfar.umd.edu/resources/database/ERIM_Arabic_DB.html*, 2002.
[21] Senior A. W. and Robinson A. J., "An Off-line Cursive Handwriting Recognition System," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 3, pp. 309-321, March 1998.
[22] Suen C. Y., presented at the Int. Workshop Frontiers Handwriting Recognition, Montreal, Canada, April 1990.
[23] Zimmermann M., "The Homepage of the IAM Database," On-line reference, *http://iamwww.unibe.ch/~zimmerma/iamdb/iamdb.html,* 2002.

Somaya Al-Ma'adeed is a PhD student at the School of Computer Science and Information Technology, University of Nottingham, UK. She received her BSc in computer science from Qatar University and MSc in mathematics and computing from Alexandria University, Egypt. Previously, she worked as an assistant teacher at Qatar University. Her main interest is in Arabic handwritten recognition.

Dave Elliman is a professor of applied computing at the School of Computer Science and Information Technology, University of Nottingham, UK. His main interests are document recognition especially graphics symbols and linework, hand-printed character and cursive script recognition, classifiers and neuro-fuzzy systems, genetic algorithms and swarm intelligence, financial modelling, knowledge reresentation and ontologies.

Colin Higgins is a senior lecturer at the School of Computer Science and Information Technology, University of Nottingham, UK. His main interests are in cursive script recognition, pen computing and a multi-modal intelligent design aid via the designers apprentice project.