Recognition of Off-line Printed Arabic Text Using Hidden Markov Models

Husni A. Al-Muhtaseb, Sabri A. Mahmoud,

Information and Computer Science Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia. e-mail: muhtaseb@kfupm.edu.sa, smasaad@kfupm.edu.sa.

and

Rami S. Qahwaji

Electronic Imaging and media communications department, University of Bradford, Bradford, UK

e-mail: R.S.R.Qahwaji@brad.ac.uk

Abstract

This paper describes a technique for automatic recognition of off-line printed Arabic text using Hidden Markov Models. In this work different sizes of overlapping and non-overlapping hierarchical windows are used to generate 16 features from each vertical sliding strip. Eight different Arabic fonts were used for testing (viz. Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic). It was experimentally proven that different fonts have their highest recognition rates at different numbers of states (5 or 7) and codebook sizes (128 or 256).

Arabic text is cursive, and each character may have up to 4 different shapes based on its location in a word. This research work considered each shape as a different class resulting in a total of 126 classes (compared to 28 Arabic letters). The achieved average recognition rates were between 98.08% and 99.89% for the eight experimental fonts.

The main contributions of this work are the novel hierarchical sliding window technique using only 16 features for each sliding window, considering each shape of Arabic characters as a separate class, bypassing the need for segmenting Arabic text, and its applicability to other languages.

Key words: Arabic text recognition, Hidden Markov Models, Feature extraction, Omni font recognition.

1. Introduction

Optical text (cursive) recognition, including handwritten text, is receiving renewed extensive research after the success in optical character recognition. Arabic text recognition, which was not researched as thoroughly as Latin, Japanese, or Chinese, is receiving a renewed interest not only from Arabic-speaking researchers but also from non-Arabic-speaking researchers. Samples of this research work are given in the references [1-10]. This has resulted in the improvement of the state of the art in Arabic text recognition in recent years. Higher recognition rates were reported and more practical data is being used for testing new techniques. In addition to the traditional applications like check verification in banks, office automation, and postal address reading, there is a large interest in searching scanned documents that are available on the internet and for searching handwritten manuscripts.

Reference may be made to [11-14] for surveys on Arabic Optical Text Recognition. Lorigo and Govindaraju addressed off-line Arabic handwriting recognition in [15]. Trenkle et al addressed advances in Arabic text recognition in [16] and Srihari and Ball presented an interesting and useful assessment of Arabic handwriting recognition technology [17]. They discussed the state of the art in off-line Arabic handwriting recognition, specified the most needed data, and discussed the technology gaps in Arabic handwriting recognition.

Due to the advantages of Hidden Markov Models (HMM) many researchers have used them for Arabic text recognition [18-32]. HMM offer several advantages. To name a few, there is no need for segmenting the Arabic cursive text, they are resistant to noise, they can tolerate variations in writing, and the HMM tools are freely available. Some researchers used HMM for handwriting word recognition [18, 19, 27, 28, 31]. Other researchers used it for text recognition [22, 23, 30, 32]. HMM was used for off-line Arabic handwritten digit recognition [33, 34] and for character recognition in [30, 35]. The techniques used in [33-34] are based on extracting different types of features of each digit as a whole, not on the sliding window principles used by the majority of researchers using HMM. For their technique to be applicable to Arabic text recognition, it has to be preceded by a segmentation step which is error-prone. Bazzi et al presented a system for bilingual text recognition (English/Arabic) [22, 36] using the sliding window principles and extracting different types of features. Dehghani et al used it for online handwritten Persian characters in [35] and for handwritten Farsi (Arabic) word recognition in [37].

Other researchers addressed the different stages of an Arabic text recognition system. Examples include: a database for Arabic handwritten text recognition in [38], a database for Arabic handwritten checks in [39], preprocessing methods in [40], segmenting of Arabic text in [41], different types of features are used in [42-44], and multiple classifiers in [45, 46].

It is worth mentioning that no generally accepted database for Arabic text recognition is freely available for researchers. Hence different researchers of Arabic text recognition use different data, and hence the recognition rates of the different techniques may not be comparable. This raises the need for researchers to make their data available for other researchers as a first step and to work on producing a comprehensive database for Arabic text recognition. In this respect we will make our data available for other interested researchers.

In this work, we followed the sliding window principle used by many researcher to extract the features to be used with the HMM [18, 20, 21, 23, 25, 47]. In our work we employed the sliding window technique used with HMM in speech recognition. Researchers, using the sliding window principle, differ in the number of features, type of features, window sizes, window overlapping, and HMM parameters.

Arabic text is cursive and is written from right to left. The Arabic alphabet has 28 basic letters, as shown in Figure 1. An Arabic letter may have up to four basic different shapes depending on the position of the letter in the word: whether it is a standalone, initial, terminal, or medial form. Letters of a word may overlap vertically with or without touching. Different Arabic letters have different sizes (height and width). Letters in a word can have short vowels (diacritics). These diacritics are written as strokes, placed either on top of, or below, the letters. A different diacritic on a letter may change the meaning of a word. Each diacritic has its own code as a separate letter when it is considered in a digital text. Readers of Arabic are accustomed to reading un-vocalized text by deducing the meaning from context. Figure 2 shows some of the characteristics of Arabic text related to character recognition. It shows a base line, overlapping letters, diacritics, and three shapes of the *Lam* character (terminal, medial, and initial).

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي

Figure 1: basic letters of Arabic.

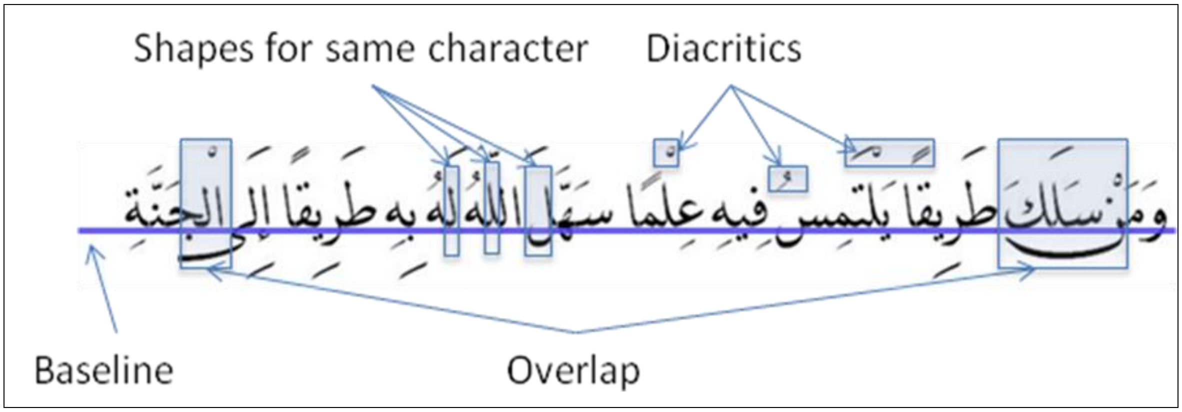

Figure 2. An example of an Arabic sentence indicating some characteristics of Arabic text.

In this paper we addressed the automatic recognition of off-line printed Arabic text. We used Arab Standardization and Metrology Organization (ASMO) character sets ASMO-449, ASMO-708 and ISO 8859-6 which define 36 Arabic letters. We also added four forms of *Lam-Alef* (لا) which is a sequence of two letters written as one set (hence resulting in 40 letters). Most of the letters can take four different shapes (from 1 to 4) but one letter has only one shape while others have two. Hence, the total number of shapes is 126.

Table 1 shows the basic Arabic letters with their categories. We group them into 3 different classes according to the number of shapes each letter takes. Class 1 consists of a single shape of the *Hamza* which comes in stand-alone state (Number 1 in Table 1). *Hamza* does not connect with any other letter. The second class (medial category) presents the letters that can come either standalone or connected only from the right (Numbers 2-5, 7, 9, 15-18, and 35-39 in Table 1). The third class (Class 4) consists of the letters that can be connected from either side or both sides, and can also appear as standalone (Numbers 6, 8, 10-14, 19-33, and 40 in Table 1).

no	Stand-alone	Terminal	Medial	Initial	Shapes	Class
1	ء	ء	ء	ء	1	1
2	آ	ـآ	ـآ	آ	2	2
3	أ	ـأ	ـأ	أ	2	2
4	ؤ	ـؤ	ـؤ	ؤ	2	2
5	إ	ـإ	ـإ	إ	2	2
6	ئ	ـئ	ـئـ	ئـ	4	3
7	ا	ـا	ـا	ا	2	2
8	ب	ـب	ـبـ	بـ	4	3
9	ة	ـة	ـة	ة	2	2
10	ت	ـت	ـتـ	تـ	4	3
11	ث	ـث	ـثـ	ثـ	4	3
12	ج	ـج	ـجـ	جـ	4	3
13	ح	ـح	ـحـ	حـ	4	3
14	خ	ـخ	ـخـ	خـ	4	3
15	د	ـد	ـد	د	2	2
16	ذ	ـذ	ـذ	ذ	2	2
17	ر	ـر	ـر	ر	2	2
18	ز	ـز	ـز	ز	2	2
19	س	ـس	ـسـ	سـ	4	3
20	ش	ـش	ـشـ	شـ	4	3
21	ص	ـص	ـصـ	صـ	4	3
22	ض	ـض	ـضـ	ضـ	4	3
23	ط	ـط	ـطـ	طـ	4	3
24	ظ	ـظ	ـظـ	ظـ	4	3
25	ع	ـع	ـعـ	عـ	4	3
26	غ	ـغ	ـغـ	غـ	4	3
27	ف	ـف	ـفـ	فـ	4	3
28	ق	ـق	ـقـ	قـ	4	3
29	ك	ـك	ـكـ	كـ	4	3
30	ل	ـل	ـلـ	لـ	4	3
31	م	ـم	ـمـ	مـ	4	3
32	ن	ـن	ـنـ	نـ	4	3
33	ه	ـه	ـهـ	هـ	4	3
34	و	ـو	ـو	و	2	2
35	لآ	لآ	ـلآ	لآ	2	2
36	لأ	لأ	ـلأ	لأ	2	2
37	لإ	لإ	ـلإ	لإ	2	2
38	لا	لا	ـلا	لا	2	2
39	ى	ـى	ـى	ى	2	2
40	ي	ـي	ـيـ	يـ	4	3

Table 1. Shapes of Arabic alphabets

Although an Arabic letter can have up to 4 different shapes, each letter is saved using only one code. A computer built-in driver uses contextual analysis to decide the right shape to display, depending on the previous and next characters. Presenting each Arabic letter with a single unique code irrespective of its shape and position is an international standard that helps a lot in searching, sorting, communications, and information retrieval.

Several aspects of our technique resulted in the high recognition rates. Our technique is based on a novel hierarchical sliding window technique which is reported for the first time in the literature. We represent each sliding strip by 16 features from one type of simple features for each sliding window, while in [22, 36] 80 features of four types of features are used. Our technique considers each shape of an Arabic character as a separate class (not combining multiple shapes in one class as is done by other researchers). The number of classes thus becomes 126 compared with 40 classes if all the shapes of a character are considered as separate classes. Our technique bypasses the need for segmentation of Arabic text, which is error-prone, and is applicable to other languages.

This paper is organized as follows. Section 2 addresses data preparation. Feature extraction is addressed in Section 3, where the details of the feature extraction phase are reported. Hidden Markov Models is addressed in Section 4. Training, recognition, and experimental results are addressed in Section 5. Finally the conclusions are presented in Section 6.

2. Data preparation

The data used in this work was extracted from the books of *Saheh Al-Bukhari* and *Saheh Muslem* [48, 49]. The text of the books represents samples of Standard Arabic. The extracted data consists of 2766 lines of text, consisting of 46062 words totaling 224109 characters including spaces. The average word length of the text is 3.93 characters. The length of the smallest line is 43 characters. The longest line has 89 characters.

Eight files with the same text were created, each with one of the eight used fonts (viz. Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic). We considered each shape as a separate class for recognition, as each shape of the same character is different from the other shapes of the same character. Out of the 2766 lines, the first 2500 lines were used for training, and the remaining 266 were used for testing, in order to have enough samples of each class for training. Table 2 shows a sample for each font.

Font Name	Sample
Arial	حسن الخلق من الإيمان
Tahoma	حسن الخلق من الإيمان
Akhbar	حسن الخلق من الإيمان
Thuluth	حسن الخلق من الإيمان
Naskh	حسن الخلق من الإيمان
Simplified Arabic	حسن الخلق من الإيمان
Traditional Arabic	حسن الخلق من الإيمان
Andalus	حسن الخلق من الإيمان

Table 2. Samples of all fonts used.

For each file the text was formatted to appear as a white font color in a black background. Moreover, each image in the ‘tif’ file has been side-reversed through a mirroring tool to speed up the training and recognition testing processes as shown in Figure (3). The same effect can be done by changing the index so that the window will slide from right to left. However, what we are suggesting is more efficient. The images of the text lines were all normalized to have a height of 80 pixels.

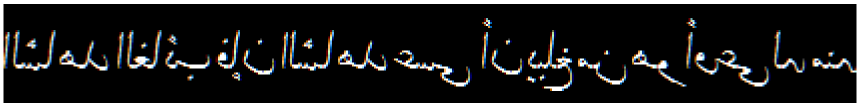

Figure 3 An example of a side reversed line using a mirroring tool.

Fifteen more lines of text were added to assure the inclusion of a sufficient number of all shapes of Arabic letters. These lines consist of 5 copies of the minimal Arabic script that has been prepared by the authors for preparing databases and benchmarks for Arabic text recognition research [50].

We prepared two dictionary books for each font. The first one represents the dictionary to be used in training and testing, where we coded each shape of a letter by its unique code. The second one includes coded Arabic characters, using English characters as the Hidden Markov Model Tool does not accept Arabic text as a dictionary.

3. Feature Extraction

A technique based on the sliding window principle was implemented to extract text features. A window with variable width and height was used. Horizontal and vertical overlapping windows were experimented with. In many experiments we tried different values for the window width and height, vertical and horizontal overlapping. Then different types of windows were utilized to get more features of each vertical segment and to decide on the most proper window size and the number of overlapping cells vertically and horizontally. The direction of the text line is considered as the feature extraction axis. Figure 4 shows the sliding window technique used in this research.

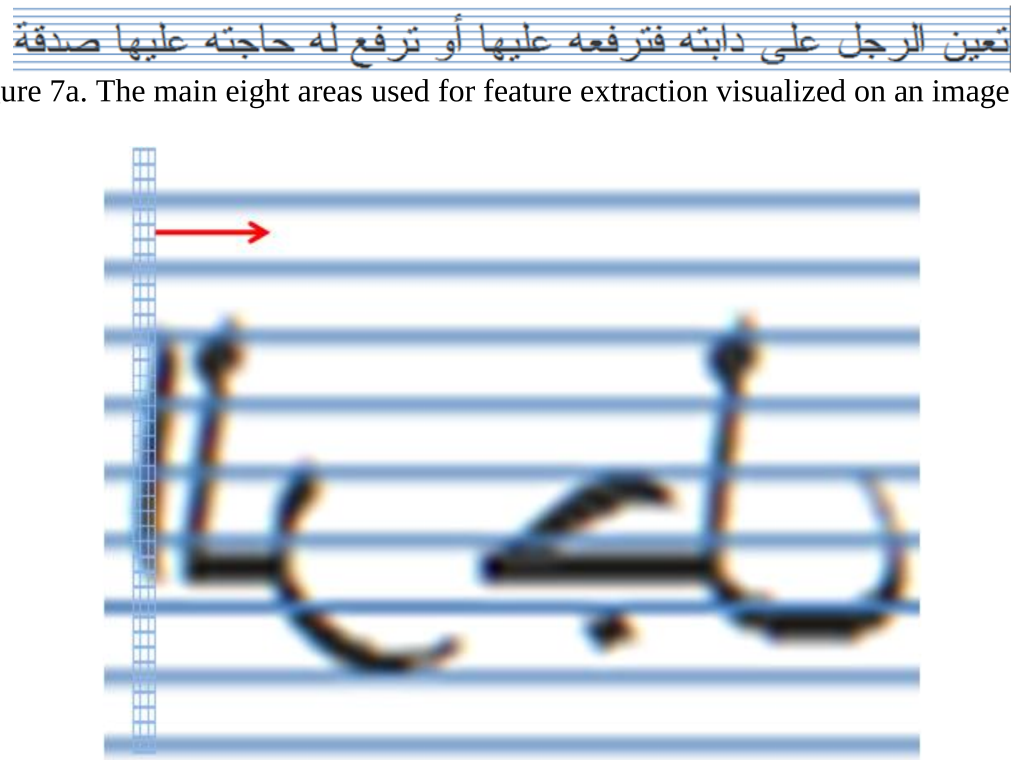
تعين الرجل على دابته فترفعه عليها أو ترفع له حاجته عليها صدقة

Figure 7a. The main eight areas used for feature extraction visualized on an image line

Figure 7 Sliding window.

Figure 4 Areas used for feature extraction and sliding windows.

Starting from the first pixel of the text line image, a vertical segment of 3 pixels width and a text line of height (T_{LH}) is taken. A window of 3 pixels width and $T_{LH}/8$ height was used to estimate the number of black pixels in the windows of the first level of the hierarchical structure. Eight vertically non-overlapping windows are used to estimate the first 8 features (features 1 to

8). Four additional features (features 9 to 12) are estimated from four vertically non-overlapping windows of 3 pixels width and TLH/4 height (windows of the second level of the hierarchical structure). Then an overlapping window with 3 pixels width and $T_{LH}/2$ height (windows of the third level of the hierarchical structure) with an overlapping of $T_{LH}/4$ is used to calculate three features (features 13 to 15). The last feature (feature 16) is found by estimating the number of black pixels in the whole vertical segment (the window of the fourth level of the hierarchical structure). Hence, 16 features were extracted for each horizontal slide. To calculate the following features, the vertical window is moved horizontally, keeping an overlap of one pixel. Sixteen features were extracted from each vertical strip and served as a feature vector in the training and/or testing processes. It has to be noted that the window size and vertical and horizontal overlapping are made settable, and hence different features may be extracted using different window sizes and vertical and horizontal overlapping. The advantages of our technique are: extracting a small number of one type of features; implementing different sizes of windows; using a hierarchical structure of windows for the same vertical strip; bypassing the need for segmentation of Arabic text; and applicability to other languages. These sixteen features have been chosen after extensive experimental testing. Table 3 illustrates features and windows used in the feature extraction phase.

Features F_{16}	Features F_{15}	Features F_3 to F_4	Features F_9 to F_{12}	Features F_1 to F_8
$F_{16} =$ $F_{13} + F_{14}$		$F_{14} =$ $F_{11} + F_{12}$	$F_{12} =$ $F_7 + F_8$	F_8 (sum of black pixels in 8th vertical rectangle)
				F_7 (sum of black pixels in 7th vertical rectangle)
	$F_{15} =$ $F_{10} + F_{11}$		$F_{11} =$ $F_5 + F_6$	F_6 (sum of black pixels in 6th vertical rectangle)
				F_5 (sum of black pixels in 5th vertical rectangle)
		$F_{13} =$ $F_9 + F_{10}$	$F_{10} =$ $F_3 + F_4$	F_4 (sum of black pixels in 4th vertical rectangle)
				F_3 (sum of black pixels in 3rd vertical rectangle)
			$F_9 =$ $F_1 + F_2$	F_2 (sum of black pixels in 2nd vertical rectangle)
				F_1 (sum of black pixels in 1st vertical rectangle)

Table 3 illustrates features and windows used in the feature extraction phase.

4. Hidden Markov Model (HMM)

Several research papers have been published using HMM for text recognition [18, 22, 30, 32, 51-53]. In order to use HMM several researchers computed the feature vectors as a function of an independent variable. This simulates the use of HMM in speech recognition where sliding frames/windows are used. The same technique is utilized in off-line text recognition where the independent variable is in the direction of the line length [22, 36]. In this paper we extract the features of an Arabic text by using the sliding windows principle to calculate the features based on sliding vertical strip which covers parts of the character. However, our technique differs from the general trend by implementing a hierarchical window structure with different window sizes and horizontal and vertical overlapping. In addition, we extract only 16 simple features (of one type) per vertical strip compared to 80 features (four types of features) used in [22, 36]. As was done in [22, 36] we bypass the need for segmenting Arabic text, and our technique is applicable to other languages. We use the same HMM classifier without modification as implemented in HTK [54]. However, we implement our own parameters of the HMM. We allowed transition to the current, the next, and the following states only. This structure allows nonlinear variations in the horizontal position. HTK models the feature vector with mixture of Gaussians. It uses the Viterbi algorithm in the recognition phase which searches for the most likely sequence of a character given the input feature vector.

In this paper a left-to-right HMM for our Arabic handwritten text recognition is implemented. Figure 5 displays the case of a 7-state HMM, showing that we allowed transition to the current, the next, and the following states only. This is in line with several research studies using HMM [22, 36]. This model allows relatively large variations in the horizontal position of the Arabic text. The sequence of state transition in the training and testing of the model is related to each text segment feature observations. In this work we have experimented with using different numbers of states and dictionary sizes, and selected the best performing ones. Although each character model could have a different number of states, we decided to adopt the same number of states for all characters in a font. However, the number of states and dictionary sizes for each font, in relation to the best recognition rates for each font, are different for each font.

In this work, each Arabic text segment is represented by a 16-dimensional feature vector as described earlier.

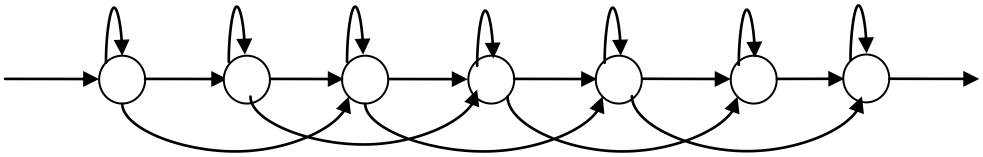

Figure 5 Seven-state Hidden Markov Model (HMM)

5. Training and Recognition

In order to have enough samples of each class for each font, in the training phase, 2500 lines were used for training and the remaining 266 lines in testing. There is no overlap between training and testing samples. A file that contains the feature vectors of each line was prepared. The feature vector contains the sixteen features extracted for each vertical strip of the image of the text line by the method described previously. All feature vectors of the vertical strips of the line are concatenated to give the feature vectors of the text line. The group of the feature files of the first 2500 lines represents the observation list for training. The group of the remaining 266 feature files represents the observation list for testing.

5.1 Training

A large number of trials were conducted to find the most suitable combinations of the number of suitable states and codebook sizes. Different combinations of the number of states and size of codebook were tested. The states that were experimented with range from 3 to 15. The sizes of codebook that were experimented with are 32, 64, 128, 192, 256, 320, 384, and 512. Table 4 shows the best combinations we experimentally found to give the best recognition rates for each font.

Font Name	Number of Sates	Codebook size
Arial	5	256
Tahoma	7	128
Akhbar	5	256
Thuluth	7	128
Naskh	7	128
Simplified Arabic	7	128
Traditional Arabic	7	256
Andalus	7	256

Table 4 combinations of number of states and size of codebook used for different fonts.

5.2 Classification

The results of testing 266 lines are summarized in Table 5. The table also shows the effect of having a unique code for each shape of each character in the classification phase (Columns 2 & 3) and then combining the shapes of the same character into one code (Columns 4 & 5). In all cases there are improvements in both correctness and accuracy in combining the different shapes of the character after recognition into one code. The following two equations were used to calculate correctness and accuracy.

$$Correctness\% = \frac{samples - (substitutions + deletions)}{samples} \times 100$$

$$Accuracy\% = \frac{samples - (substitutions + insertions + deletions)}{samples} \times 100$$

Table 6 summarizes the classification results for the Arial font. The results of all other fonts are summarized in Table 7. As the resultant confusion matrices are too large to display in row format (at least 126 rows X 126 columns are needed), we summarize the confusion matrix in a more informative way by collapsing all different shapes of the same character into one entry and by listing error details for each character. This will actually be the result after converting the recognized text from the unique coding of each shape to the unique coding of each character (which is done by the contextual analysis module, a tool we built for this purpose).

The following subsections discuss the classification results for the Arial font and a summary of the results for all other fonts are summarized in Table 7 which shows the average correctness and accuracy for all fonts (viz. Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic).

Table 5 Summary of Results per font type with and without shape expansion

	With Expanded shapes		**With Collapsed shapes**	
Text font	**% of Correctness**	**% of Accuracy**	**% of Correctness**	**% of Accuracy**
Arial	99.89	99.85	99.94	99.90
Tahoma	99.80	99.57	99.92	99.68
Akhbar	99.33	99.25	99.43	99.34

Thulth	98.08	98.02	98.85	98.78
Naskh	98.12	98.02	98.19	98.09
Simplified Arabic	99.69	99.55	99.84	99.70
Traditional Arabic	98.85	98.81	98.87	98.83
Andalus	98.92	96.83	99.99	97.86

a. Arial font Classification

Table 6 shows the classifications results for the Arial font. The correctness percentage was 99.94 and the accuracy percentage was 99.90. Only four letters out of 43 had some errors. The letter ح has been substituted by the letter ج four times out of 234 instances. The only difference between the two characters is the dot in the body of the letter ج. The second error consists of two replacements of the letter ه by the letter ء out of 665 instances. The third error was substituting the ligature لأ by a blank four times out of 40. The fourth error was substituting the ligature لله once by ه out of 491 times. Other than the substitutions, 10 insertions were added (two of them were blanks). The blank problems were reported by several researchers including [22].

Table 6 Classification Results for Arial Font.

Let	**Samples**	**Correct**	**Errors**	**% Recognition**	**% Error**	**Del**	**Ins**	**Correctness**	**Accuracy**	**Error Details**
Sil	532	532	0	100.0	0.0	0	0	100.0	100.0	
ء	83	83	0	100.0	0.0	0	0	100.0	100.0	
آ	10	8	2	80.0	20.0	0	0	80.0	80.0	
أ	484	484	0	100.0	0.0	0	0	100.0	100.0	
ؤ	14	14	0	100.0	0.0	0	0	100.0	100.0	
إ	157	157	0	100.0	0.0	0	0	100.0	100.0	
ئ	43	43	0	100.0	0.0	0	0	100.0	100.0	
ا	2114	2114	0	100.0	0.0	0	1	100.0	100.0	
ب	409	409	0	100.0	0.0	0	0	100.0	100.0	
ة	234	234	0	100.0	0.0	0	0	100.0	100.0	
ت	420	420	0	100.0	0.0	0	0	100.0	100.0	
ث	124	124	0	100.0	0.0	0	0	100.0	100.0	
ج	170	170	0	100.0	0.0	0	0	100.0	100.0	
ح	234	230	4	98.3	1.7	0	0	98.3	98.3	جـ4
خ	113	113	0	100.0	0.0	0	0	100.0	100.0	
د	344	344	0	100.0	0.0	0	1	100.0	99.7	
ذ	97	97	0	100.0	0.0	0	0	100.0	100.0	
ر	702	702	0	100.0	0.0	0	0	100.0	100.0	
ز	46	46	0	100.0	0.0	0	0	100.0	100.0	
س	640	640	0	100.0	0.0	0	0	100.0	100.0	
ش	119	119	0	100.0	0.0	0	0	100.0	100.0	
ص	**415**	**415**	**0**	**100.0**	**0.0**	**0**	**0**	**100.0**	**100.0**	

Let	Samples	Correct	Errors	% Recognition	% Error	Del	Ins	Correctness	Accuracy	Error Details
ض	93	93	0	100.0	0.0	0	0	100.0	100.0	
ط	68	68	0	100.0	0.0	0	0	100.0	100.0	
ظ	15	15	0	100.0	0.0	0	0	100.0	100.0	
ع	818	818	0	100.0	0.0	0	0	100.0	100.0	
غ	44	44	0	100.0	0.0	0	0	100.0	100.0	
ف	495	495	0	100.0	0.0	0	0	100.0	100.0	
ق	467	467	0	100.0	0.0	0	0	100.0	100.0	
ك	288	288	0	100.0	0.0	0	0	100.0	100.0	
ل	2136	2136	0	100.0	0.0	0	2	100.0	99.9	
م	1005	1005	0	100.0	0.0	0	0	100.0	100.0	
ن	1023	1023	0	100.0	0.0	0	0	100.0	100.0	
ه	665	663	2	99.7	0.3	0	0	99.7	99.7	ء-2
و	937	937	0	100.0	0.0	0	0	100.0	100.0	
لآ	5	5	0	100.0	0.0	0	0	100.0	100.0	
لأ	40	36	4	90.0	10.0	0	0	90.0	90.0	Blnk-4
لإ	14	14	0	100.0	0.0	0	0	100.0	100.0	
لا	207	207	0	100.0	0.0	0	4	100.0	98.1	
ى	413	413	0	100.0	0.0	0	0	100.0	100.0	
ي	1159	1159	0	100.0	0.0	0	0	100.0	100.0	
Blnk	4637	4637	0	100.0	0.0	0	2	100.0	100.0	
ﷲ	491	490	1			0	0	99.8	99.8	ـه1
Ins			10							ـلا4 ـل2 ـد1 ـا1 Blnk-2
Total	**22524**	**22511**	**13**	**99.94**	**0.06**	**0**	**10**	**99.94**	**99.90**	

b. Classification of other fonts

Table 7 summarizes the results of Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic fonts. Arial font was included for comparison purposes. The table shows the average correctness and accuracy of all these fonts.

	Arial		Tahoma		Akhbar		Thuluth		Naskh		Simplified Arabic		Traditional Arabic		Andalus	
Let	Correctness	Accuracy	Correctness	Accuracy	Correctness	Accuracy	Correctness	Accuracy	Correctness	Accuracy	Correctness	Accuracy	Correctness	Accuracy	Correctness	Accuracy
S	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ء	100	100	100	98.8	96.3	96.3	100	97.4	100	100	100	100	100	100	100	100
آ	80	80	100	100	60	60	100	100	100	100	40	40	100	100	100	100
أ	100	100	100	100	99.6	99.6	99.8	99.8	100	100	96.3	96.3	98.5	98.5	100	100
ؤ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
إ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ئ	100	100	100	100	100	100	100	100	97.7	97.7	100	100	100	100	100	100
ا	100	100	100	99.9	99.1	99	100	100	98.5	98.4	100	99.8	98.7	98.5	100	100
ب	100	100	100	100	99.3	99.3	93.9	93.9	89.3	88.4	100	100	92.5	92.5	100	99.8
ة	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ت	100	100	99.8	99.8	100	100	97.8	97.6	98.8	98.8	99.8	99.8	99.1	99.1	100	100
ث	100	100	100	100	98.4	92.6	100	100	95.1	95.1	100	100	99.2	99.2	100	100
ج	100	100	99.4	99.4	100	100	97.1	97.1	96.5	96.5	100	100	91.8	91.8	100	100
ح	98.3	98.3	94.4	94.4	100	100	82	82	87.6	87.6	100	100	84.2	84.2	100	100
خ	100	100	100	100	100	100	98.2	98.2	90.3	89.4	100	100	98.2	97.4	100	100
د	100	99.7	100	99.1	100	99.7	99.1	98.8	100	100	100	99.7	99.7	99.4	100	99.7
ذ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ر	100	100	100	100	100	100	94.9	94.9	98.6	98.6	100	100	99.9	99.9	100	100
ز	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
س	100	100	100	100	99.8	98.9	99.8	99.8	99.2	99.2	100	100	99.7	99.7	100	100
ش	100	100	100	100	99.2	99.2	99.2	99.2	100	100	100	100	100	100	100	100
ص	100	100	100	100	100	100	99.8	99.8	98.8	98.8	100	100	98.8	98.8	100	100
ض	100	100	100	100	100	100	98.9	98.9	100	100	100	100	100	100	100	100
ط	100	100	98.5	97.1	100	100	97.1	97.1	97.1	97.1	100	100	100	100	100	100
ظ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ع	100	100	100	100	99.8	99.8	99.5	99.5	98.7	98.7	100	100	99.9	99.9	100	100
غ	100	100	100	100	97.7	97.7	97.7	97.7	100	100	100	100	100	100	100	100
ف	100	100	100	100	99.6	99.6	100	99.2	99.6	99.6	100	100	100	100	100	100
ق	100	100	100	100	98.9	98.9	100	100	99.1	99.1	99.8	99.8	100	100	100	100
ك	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
ل	100	99.9	100	97.9	98.1	97.9	99.8	99.7	98.2	98.1	100	99.7	99.6	99.6	100	77.7
م	100	100	100	100	99.9	99.9	98.2	98.1	90	89	100	100	94.1	94	100	100
ن	100	100	100	100	99.4	99.3	99.1	99.1	96.7	96.5	99.9	99.9	98.1	98	100	100
ه	99.7	99.7	100	100	99.7	99.7	99.1	99.1	99.7	99.7	100	100	99.1	99.1	100	100
و	100	100	100	100	100	100	99.8	99.8	99.9	99.9	100	100	100	100	100	100
لآ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
لأ	90	90	100	100	100	100	100	100	97.5	97.5	100	100	100	100	100	100
لإ	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
لا	100	98.1	100	100	100	100	100	100	99	99	100	100	100	100	100	100
ى	100	100	100	100	100	100	89.1	89.1	99.5	99.5	98.6	98.6	98.6	98.6	100	100
ي	100	100	100	100	98.4	98.4	99.7	99.7	97.7	97.7	100	100	98.4	98.4	100	100
B	100	100	100	100	100	100	99.3	99.2	99.4	99.4	100	99.6	99.9	99.9	99.9	99.9
ﷲ	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	99.8	100	100
T	**99.9**	**99.9**	**99.9**	**99.7**	**99.4**	**99.3**	**98.9**	**98.8**	**98.2**	**98.1**	**99.8**	**99.7**	**98.9**	**98.8**	**100**	**97.9**

Table 7 summarizes the results of Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic fonts.

c. Do the suggested features work for English?

Although features presented in this work were designed for Arabic script, the question of whether the same features would work for English arises. To validate this matter, an English text of 1230 lines was prepared using Microsoft Sans Serifs font as an example. The text file is converted into an array of image lines. The same features used for Arabic text images were extracted for the English text. The first 1100 lines of the English text images were used for training, and the remaining 130 lines were used for testing. The classification results were 98.92% for the correctness and 98.90% for the accuracy. Out of 4921 characters there were two deletions, 51 substitutions, and one insertion. This shows the applicability of our features for English text as well. All these errors are summarized in Table 8. The table contains the letters that had classification errors. All other letters were correctly classified. It is to be noted that we applied the same model for Arabic text recognition without change or enhancement for a proof of concept.

	m	**n**	**o**	**Blank**	**Capital i**	**Del**
Small L	0	0	0	0	0	2
n	16		0	0	0	0
r	2	3	0	0	0	0
t	0	0	0	13	0	0
/	0	0	0	0	1	0
c	0	0	16	0	0	0
Ins	**0**	**0**	**0**	**1**	**0**	

Table 8. All errors appeared in the English test.

6. Conclusions

This paper presents a technique for automatic recognition of off-line Arabic text recognition based on estimating simple and effective features that are suitable for use with the HMM (which is normally employed for speech recognition). We analyzed the performance of the HMM with different numbers of features, different sizes of sliding windows, different numbers of states and different dictionary sizes. We applied the technique for eight Arabic fonts (viz. Arial, Tahoma, Akhbar, Thuluth, Naskh, Simplified Arabic, Andalus, and Traditional Arabic). After a large number of experiments, we selected the number of features, the number of states and the dictionary size for each font according to each font's highest recognition rate. The technique is scale- and translation-invariant. The experimental results indicate the effectiveness of the proposed technique in the automatic recognition of off-line Arabic text with different types of fonts.

A database of 2766 lines was used in the training and testing phase. 2500 lines were used in training and the remaining 266 in testing. The experimental results, discussed earlier, show the effectives of our features. We used a small number of simple and effective features that can be computed quickly. This was repeated for all vertical strips with an overlap of one pixel. Only sixteen features were extracted from each vertical strip of the text line image. We applied our technique to eight different Arabic fonts. They all gave acceptable recognition rates (accuracy percentages were: Arial 99.9, Tahoma 99.68, Akhbar 99.34, Thuluth 98.78, Naskh 98.09, Simplified Arabic 99.7, Traditional Arabic 98.83, Andalus 97.86).

Several aspects of our technique resulted in the high recognition rates. Our technique is based on a novel hierarchical sliding window technique with overlapping and nonoverlapping windows which is reported for the first time in the literature. We represent each sliding strip by 16 features from one type of simple features for each sliding window, while other researchers used 80 features of four types of features (viz. intensity, vertical and horizontal derivative, and local scope and correlation) [22,36]. To the knowledge of the researchers, no other researchers have included the following letters/ligatures in their classifications: (آ, إ, أ, لآ, لأ, لإ, and ﷲ). We considered each shape of an Arabic character as a separate class, not combining multiple shapes in one class as is done by other researchers. The number of classes became 126 compared with 40 classes if all the shapes of a character are considered as separate classes. This technique does not require segmentation of Arabic cursive text which is known to be problematic where an error in segmentation results in more errors in recognitions. Hence, using this technique, segmentation was a by-product of our technique. Finally, the presented technique is language independent.

The researchers are currently exploring the use of more elaborate data and testing the system on Omni font. In addition, they are exploring post-processing techniques to enhance the recognition rates further as they feel that the extracted features and the classifier have done an excellent job in the classification phase.

Acknowledgment

We would like to thank the referees for their constructive criticism and stimulating remarks. The modification of the original manuscript to address those remarks improved the revised manuscript considerably. The first and second authors would like to thank King Fahd University of Petroleum and Minerals for supporting this research work. This work is partially supported by KFUPM internal project number IN060337.

References

[1] M.Arivazhagan, H.Srinivasan, S.Srihari, A statistical approach to line Segmentation in handwritten documents, in: Proceedings of SPIE, 2007.

[2] R. Davidson and R. Hopely, “Arabic and Persian OCR Training and Test Data Sets,” Proc. Symp. Document Image Understanding Technology, pp. 303-307, 1997.

[3] J.Femiani, M.Phielipp, A.Razdan, A System for Discriminating Handwriting from Machine Print on Noisy Arabic Datasets, in: SDIUT 05: Proceedings of the 2005 Symposium on Document Image Understanding Technology, CollegePark, Maryland,2005.

[4] A. Gillies, E. Erlandson, J. Trenkle and S. Schlosser, “Arabic Text Recognition System,” Proc. Symp. Document Image Understanding Technology, 1999.

[5] J. Jin, H. Wang, X. Ding and L. Peng, “Printed Arabic Document Recognition System,” Proc. SPIE-IS&T Electronic Imaging, vol. 5676, pp. 48-55, 2005.

[6] G. Kim, V. Govindaraju and S. Srihari, “Architecture for Handwritten Text Recognition Systems,” Advances in Handwriting Recognition, Series in Machine Perception and Artificial Intelligence, pp. 163-172, 1999.

[7] L.Lorigo, V.Govindaraju, Segmentation and pre-recognition of Arabic Hand writing, in: ICDAR 05: Proceedings of the Ninth International Conference on Document Analysis and Recognition, vol.2, IEEE Computer Society, Seoul, Korea, 2005, pp. 605-609.

[8] T.Sari, L.Souici, M.Sellami, O-line Handwritten Arabic Character Segmentation Algorithm: ACSA, in:Proceedings of the Eighth International Workshop on Frontiers in Handwriting Recognition,2002.

[9] L. Souici-Meslati and M. Sellami, “A Hybrid Approach for Arabic Literal Amounts Recognition,” The Arabian J. Science and Eng., vol. 29, pp. 177-194, 2004.

[10] S.Srihari, G.Ball, H.Srinivasan, Versatile Search of Scanned Arabic Handwriting, in: SACH06: Summit on Arabic and Chinese Handwriting, 2006.

[11] B. Al-Badr, S. Mahmoud, “Survey and Bibliography of Arabic Optical Text Recognition,” J. of Signal Processing, Vol. 41, No.1, pp.49-77 (Jan. 1995).

[12] M. Khorsheed, “Off-line Arabic Character Recognition – A Review,” Pattern Analysiss & Applications, 5:31-45, 2002.

[13] A. Eldin and A. Nouh, “Arabic Character Recognition: A Survey,” Proc. SPIE Conf. Optical Pattern Recognition, pp. 331-340, 1998.

[14] N. Amara, F.Bouslama, Classication of Arabic script using multiple Sources of information: State of the art and perspectives, International Journal On Document Analysis and Recognition 5(4)(2005),195-212.

[15] L. Lorigo, V. Govindaraju, “Offline Arabic Handwriting Recognition: A Survey”, EEE Trans. Pattern Analysis and Machine Intelligence, vol. 28, no. 5, pp. 712-724,May 2006.

[16] J. Trenkle, A. Gillies, E. Erlandson, S. Schlosser and S. Cavin, “Advances in Arabic Text Recognition,” Proc. Symp. Document Image Understanding Technology, 2001.

[17] S. Srihari and G. Ball, An Assessment of Arabic Handwriting Recognition Technology, TR-03-07 report, University at Buffalo, The State University of New York, 2007.

[18] S. Almaadeed, C. Higgens and D. Elliman, “Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach,” Proc. 16th Int"l Conf. Pattern Recognition, vol. 3, pp. 481-484, 2002.

[19] S.Almaadeed, C.Higgens, D.Elliman, On-Line Recognition of Handwritten Arabic Words Using Multiple Hidden Markov Models, Knowledge-Based Systems 17(2004), 75-79, 2004.

[20] S. Al-Qahtani and M. Khorsheed, “An Omni-Font HTK-Based Arabic Recognition System,” Proc. Eighth IASTED Int"l Conf. Artificial Intelligence and Soft Computing, 2004.

[21] S. Al-Qahtani and M. Khorsheed, “A HTK-Based System to Recognise Arabic Script,” Proc. Fourth IASTED Int"l Conf. Visualization, Imaging, and Image Processing, 2004.

[22] I. Bazzi, R. Schwartz and J. Makhoul, “An Omnifont Open-Vocabulary OCR System for English and Arabic,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, pp. 495-504, 1999.

[23] H. Bunke, S. Bengio and A. Vinciarelli, “Off-Line Recognition of Unconstrained Handwritten Texts Using HMMS and Statistical Language Models,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 26, pp. 709-720, 2004.

[24] R.El-Hajj, L.Likforman-Sulem, C.Mokbel, Arabic Hand writing Recognition Using Baseline Dependant Features and Hidden Markov Modeling, in: Proc. 9th Intl Conf. Document Analysis and Recognition, ICDAR 05, 2005, pp. 893-897.

[25] M. Khorsheed, Recognising Hand written Arabic Manuscripts Using a Single Hidden Markov Model, Pattern Recognition Letters 24(2003), 2235-2242.

[26] H. Miled, N. Amara, Planar Markov Modeling for Arabic Writing Recognition: Advancement State, in: Proc. Intl Conf. Document Analysis and Recognition, pp. 69-73, 2001.

[27] M.Pechwitz, V.Margner, HMM based approach for handwritten Arabic word Recognition using the IFN/ENIT-database ,in: ICDAR03: Proceedings of The Seventh International Conference on Document Analysis and Recognition, IEEE Computer Society, Edinburgh, Scotland, pp. 890-894, 2003, 2003.

[28] R.Safabakhsh, P.Adibi, Nastaaligh Handwritten Word Recognition Using a Continuous-Density Variable-Duration HMM, The Arabian J. Science and Eng. 30(2005),95-118, 2005.

[29] S.Touj, N. Amara, H.Amiri, Arabic Handwritten Words Recognition Based On a Planar Hidden Markov Model., International Arab Journal of Information Technology 2(4)(2005),318-325.

[30] A. Hassin, X. Tang, J. Liu, and W. Zhao, Printed Arabic character recognition using HMM, Journal of Computer Science and Technology, Volume 19 Issue 4, July 2004, pp. 538-543.

[31] M. Mohamed and P. Gader, “Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Techniques,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 5, pp. 548-554, May 1996.

[32] J. Hu, S. . Lim, M. Brown, Writer independent on-line handwriting recognition using an HMM approach, Pattern Recognition 33 (2000) 133-147.

[33] S. Mahmoud, Recognition of writer-independent Off-line Handwritten Arabic (Indian) Numerals Using Hidden Markov Models, Accepted for publication in the Journal of Signal Processing.

[34] S. Mahmoud, M. Abu-Amara, Recognition of Handwritten Arabic (Indian) Numerals Using Radon Transform, submitted for publication.

[35] A.Dehghani, F.Shabani, P.Nava, On-Line Recognition of Isolated Persian Handwritten Characters Using Multiple Hidden Markov Models, in: Proc. Intl Conf. Information Technology: Coding and Computing, pp. 506-510, 2001.

[36] I. Bazzi, C. LaPre, J. Makhoul, and R. Schwartz, “Omnifont and Unlimited Vocabulary OCR for English and Arabic,” Proc. Int’l Conf. Document Analysis and Recognition, vol. 2, pp. 842-846, Ulm, Germany, 1997.

[37] M.Dehghan, K.Faez, M.Ahmadi, M.Shridhar, Hand written Farsi(Arabic) Word Recognition: A Holistic Approach Using Discrete HMM, Pattern Recognition 34(2001),1057-1065.

[38] S. Almaadeed, D. Elliman and C. Higgins, “A Data Base for Arabic Handwritten Text Recognition Research,” Proc. Eighth Int"l Workshop Frontiers in Handwriting Recognition, pp. 485-489, 2002.

[39] Y. Al-Ohali, M. Cheriet and C. Suen, “Databases for Recognition of Handwritten Arabic Cheques,” Pattern Recognition, vol. 36, pp. 111-121, 2003.

[40] F.Farooq, V.Govindaraju, M. Perrone, Pre-processing methods for Handwritten Arabic documents, in: ICDAR 05: Proceedings of the Ninth International Conference on Document Analysis and Recognition,vol.1, IEEE Computer Society, Seoul, Korea, . 267-271, 2005.

[41] R.Haraty, A.Hamid, Segmenting Hand written Arabic Text, in: Proc. Intl Conf. Computer Science, Software Eng., Information Technology, e-Business, And Applications, 2002.

[42] A.Amin, Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming, Pattern Recognition Letters 24(2003),3187-3196.

[43] M.Fahmy, S. Ali, Automatic Recognition of Hand written Arabic Characters Using Their Geometrical Features, Studies in Informatics and Control J.10.

[44] S.Mozaari, K.Faez, M.Ziaratban, Structural decomposition and statistical Description of Farsi/Arabic hand written numeric characters, In: ICDAR05: Proceedings of the Ninth International Conference on Document Analysis and Recognition, vol.1,IEEE Computer Society, Seoul, Korea, 2005.

[45] N.Farah, L.Souici, L.Farah, M.Sellami, Arabic Words Recognition with Classiers Combination: An Application to Literal Amounts, in: Proc. Artificial Intelligence: Methodology, Systems, and Applications, 2004.

[46] N. Farah, A. Ennaji, T. Khadir and M. Sellami, “Benefits of Multi-Classifier Systems for Arabic Handwritten Words Recognition,” Proc. Int"l Conf. Document Analysis and Recognition, pp. 222-226, 2005.

[47] M. Shahrezea, K. Faez, A. Khotanzad, Recognition of handwritten Persian/Arabic numerals by shadow coding and an edited probabilistic neural network. In: Proceedings of the International Conference on Image Processing, vol. 3, pp. 436–439,1995.

[48] M. Al-Bukhari, "*Al-Jame' Al-Saheeh (Sahih Al-Bukhari)*", Dar Al-Jeel, Beirut, 2005 (in Arabic).

[49] M. Al-Naysabouri, "*Al-Jame' Al-Saheeh (Sahih Muslim)*", Dar Al-Jeel, Beirut, 2006 (in Arabic).

[50] H. Al-Muhtaseb, S. Mahmoud, R. Qahwaji, “A Novel Minimal Arabic Script for Preparing Databases and Benchmarks for Arabic Text Recognition Research”, To be published, 2008.

[51] S. Almaadeed, C. Higgens, and D. Elliman, “Recognition of Off-line Handwritten Arabic Words using Hidden Markov Model Approach,” ICPR 2002, Quebec City, August 2002, pp. 481-484.

[52] M. Mohamed and P. Gader, “Handwritten Word Recognition Using Segmentation-Free Hidden Markov Modeling and Segmentation-Based Dynamic Programming Techniques,” IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 18, no. 5, pp. 548-554, May 1996.

[53] HTK Speech Recognition Toolkit, http://htk.eng.cam.ac.uk/