Optical Character Recognition of Arabic Printed Text

Safwa Taha, Yusra Babiker, and Mohamed Abbas
Electrical and Electronics Engineering Department
University of Khartoum
Khartoum, Sudan
safwa91@yahoo.com,usrababiker91@gmail.com,and mabbas@hotmail.com

Abstract— **Optical character recognition (OCR) systems improve human- machine interaction. They are widely used in many areas such as editing and storing previously printed or handwritten documents. Much of research has been done regarding the identification of Latin, Japanese and Chinese characters. However, very little investigation has been performed regarding Arabic recognition. Probably the reason is limitation of IT activities in Arabic speaking countries and the difficulty and complexity of Arabic characters identification compared to the others. More difficulties are introduced from the cursive nature of Arabic text. In this paper, a technique has been employed to segment printed Arabic text in order to separate the Arabic characters and then extracting powerful features for each to be recognized. In-order to recognize characters, those features are then compared with a pre-prepared database fields. Although the database was prepared from characters written in Time New Roman font, experimental results show the relatively high accuracy of the method developed when it is tested on several sizes of several fonts beside Time New Roman font.**

Keywords- Optical character recognition (OCR); OCR; Arabic characters; Segmentation; Feature extraction; Recognition

I. INTRODUCTION

Optical character recognition (OCR) is the process of automatic conversion of scanned images of machine printed or handwritten documents into computer processable code [1]. Off-line printed text recognition is a sub-task of OCR whose domain can be machine-printed material and handwritten text.

Arabic is a popular script. It is estimated that there are more than one billion Arabic script users in the world. If OCR systems are available for Arabic characters, they will have a great commercial value [1].

Due to the cursive nature of Arabic letters Arabic OCR systems require a highly sophisticated segmentation stage. Therefore, much of the previous work that has been performed in this field was considering isolated Arabic letters only.

Generally, Arabic OCR systems are classified into two major types depending on the method of segmentation been used: segmentation–free systems and segmentation–based systems. Segmentation–free systems [6,7] recognize the text words without segmenting it into words, characters or primitives. These systems avoid the segmentation procedure overhead.

Segmentation–based systems are classified into two types recognition-based [1], and dissection segmentation [8]. Recognition-based segmentation applies some analysis in the beginning to determine some points of interest called fragments. Segmentation depends on a variable size window that enlarges and moves according to the recognition. Error propagation is introduced. If the system failed in recognizing a letter the window will keep enlarging each time trying to get a result, hence the system may lose a recognition result of one or more letters. The special case of such scenario is a misclassification of an entire word.

Dissection segmentation approach segments the word into primitives or characters. Primitives are possibly smaller than characters like strokes, intersection points, loops, dots and ligatures. The recognition systems recognize these primitives and combine them to generate a word which can be characters. Segmenting a word into characters without errors is a very hard job because characters inside a text are fully connected and there are no clear parameters to detect the exact letters' boundaries.

The added value of this work that it's based on a novel developed segmentation technique which is a dissection segmentation method that implements a very powerful descriptor, 'junction line'. A pre-analysis is performed on the upcoming image to define areas of inertest which are junction lines between letters. These areas are used as signs for letter starts and ends. The algorithm shows a high accuracy in identifying the proper boundaries of letters to be recognized.

The feature extraction stage has been supported by strong features. The number of holes per a letter and the number of connected components per a letter [2] are reliable features that lead to grouping Arabic letters into 8 different groups [3]. The grouping of letters has enabled the system to search only inside each corresponding group. Thus, the comparison process is now only inside a single group of letters instead of the total number of letter patterns.

The boundary shape feature is also included as a descriptor for each letter using the chain code [2], and [5]. Hence, the system has a strong description for each Arabic letter.

The structure of the proposed system is composed of major four stages (1) image acquisition and preprocessing, (2) segmentation, (3) feature extraction and (4) classification.

The rest of this paper is organized as follows: some background knowledge about the characteristics of Arabic script is given in Section II below. The model of the OCR system and the first two stages are explained in Section III .A detailed description of the algorithms used in feature

extraction stage. Section V presents the database preparation and the classification stage. Experimental results are discussed in Section VII. Finally, the suggested future work is presented in Section VIII.

II. BACKGROUND

An in-depth understanding of the characteristic of Arabic is essential for the implementation of its OCR system. This knowledge helps to discover the suitability of existing techniques to the system and may also lead to the development of new techniques.

Most Arabic characters have dot(s) zigzag(s) associated with the character and these components can be above, below or inside the character. Many characters have a similar shape, the position or numbers of these secondary components make the only difference as in TABLE I.

TABLE I
Characters contain dots

	Characters contain dot/s		
	One	***Two***	***Three***
Upper dot/s	ن,ف,غ,ظ,ض,ز,ذ,خ	ق,ت	ش,ث
Lower dot/s	ج,ب	ي	-

There are 28 characters in the Arabic alphabet. Each character has 2 to 4 different forms which depend on its position in the word or sub-word. As a result, there are 100 classes to be recognized. TABLE II shows the forms of Arabic letters.

TABLE II
Arabic letters have different forms depending on their position in the word.

Letter Name	Isolated	Initial	Middle	End
Alif	ا	ا	ـا	ـا
Ba’	ب	بـ	ـبـ	ـب
Ta’	ت	تـ	ـتـ	ـت
Tha’	ث	ثـ	ـثـ	ـث
Jeem	ج	جـ	ـجـ	ـج
Ha’	ح	حـ	ـحـ	ـح
Kha’	خ	خـ	ـخـ	ـخ
Dal	د	د	ـد	ـد
Thal	ذ	ذ	ـذ	ـذ
Ra’	ر	ر	ـر	ـر
Zy	ز	ز	ـز	ـز
Seen	س	سـ	ـسـ	ـس
Sheen	ش	شـ	ـشـ	ـش
Sad	ص	صـ	ـصـ	ـص
Dhad	ض	ضـ	ـضـ	ـض
T’ah	ط	طـ	ـطـ	ـط
The’ah	ظ	ظـ	ـظـ	ـظ
Ain	ع	عـ	ـعـ	ـع
Ghain	غ	غـ	ـغـ	ـغ
Fa	ف	فـ	ـفـ	ـف
Qaf	ق	قـ	ـقـ	ـق
Kaf	ك	كـ	ـكـ	ـك
Lam	ل	لـ	ـلـ	ـل
Meem	م	مـ	ـمـ	ـم
Noon	ن	نـ	ـنـ	ـن
Ha’	ه	هـ	ـهـ	ـه
Waw	و	و	ـو	ـو
Ya	ي	يـ	ـيـ	ـي

Arabic is a cursive-type language, which is written form right to left, and so recognition should follow this way, Figure 1 illustrate this. An Arabic word can have one or more sub word, see Figure 2. Moreover, Arabic letters may horizontally overlap and stack on each other. These introduce problems for both the segmentation of the word and the classification of the letter, see Figure 3.

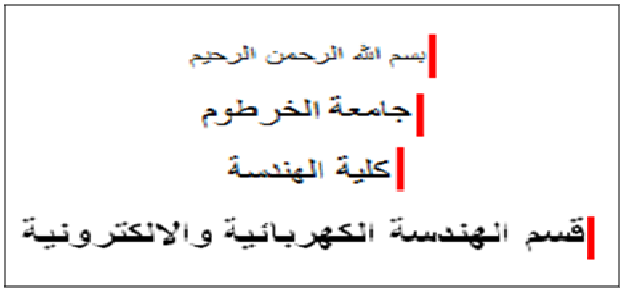

Figure 1 A sample text presents the writing direction style and the cursive nature of the Arabic text

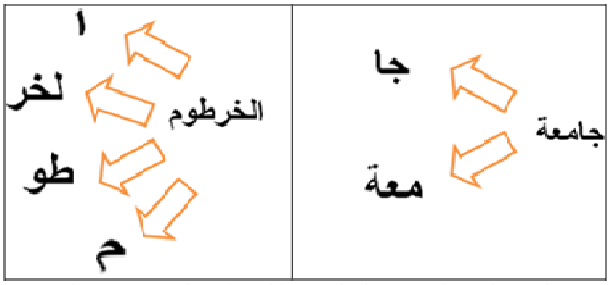

Figure 2 Samples of Arabic words that contains sub words

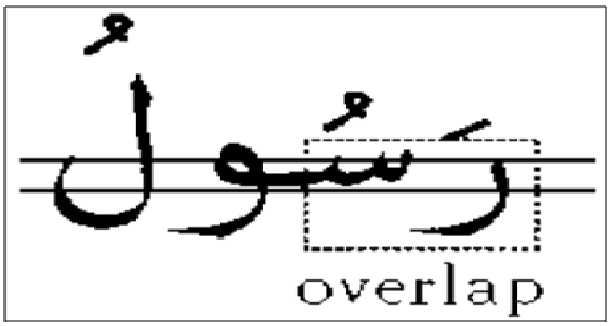

Figure 3 an example of overlapping Arabic word

To simulate the human vision analogy, every single OCR system uses specific features that describe the target language characters. Generally, features extraction methods can be categorised into structural, statistical, and global transformation. Structural features method features are usually extracted based on the text topologies of an Arabic text which include loops, the intersection points, dots, zigzags, height, width, number of crossing points [9-12]. Statistical features methods are quick and effective, but may be affected by noise. The statistical features used for Arabic text recognition include: zoning, characteristic loci, crossings and moments [13-15]. Global Transformation methods aim to shorten the

text representation in order to get better results. The global transformations methods used for Arabic text recognition include: horizontal and vertical projections, coding using Freeman chain codes, Hough transform, Gabor transform [16, 17].

Classification in an OCR system is the main decision making stage. Based on the extracted features, the classifier attempts to identify the pattern that represents the input features. The classification methods can be divided into three types: structural, statistical or neural network classifiers. The structural classifier identifies the primitives of the character first and then identifies the character by a set of primitives. Statistical classification methods map a fixed length of feature vectors with a partitioned space. One of the most efficient statistical classification methods is the hidden Markov models (HMMs) [6,7].

Artificial Neural Network (NN) is one of the most successful classifier methods used in the pattern recognition. However, to improve the intelligence of these ANNs, huge iterations, complex computations, and learning algorithms are needed, which also lead to consume the processor time. Therefore, if the recognition accuracy is improved, the consumed time will increase and vice versa.

Machine learning classification techniques have the ability to learn from their errors and to recognize new patterns according to their experience. Thus, they are highly recommended for OCR systems of handwritten texts [3, 14]. However, Printed texts have consistent parameters, therefore, no need for machine learning techniques which introduce considerable overhead.

Although the amount of research into machine-print recognition appears to be tailing off as many research groups turn their attention to handwriting recognition, it is suggested that there are still significant challenges in the machine-print domain. One of these challenges is to deal effectively with noisy, multi-font data, including possibly hundreds of fonts. Therefore, this work has been implemented to focus on the printed text using a structural classier that avoids any overheads introduced by machine learning techniques.

III. SYSTEM MODEL AND SEGMENTATION

The first stage of the model consists of image acquisition and pre-processing phases as shown in Figure 4. A flatbed scanner of 300dpi is used to acquire images.

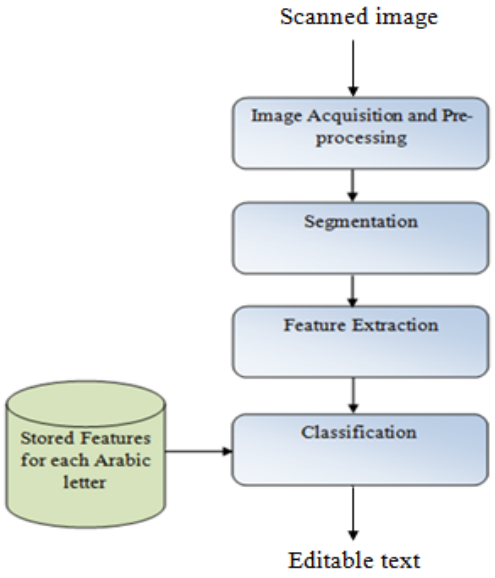

Figure 4 The system model

The purpose of pre-processing phase is to enhance the acquired image by using multiple processes such as thresholding, binarization and noise elimination [4].

Modern scanners have a defined page borders, hence an assumption is made; the scanned document has no homogeneous transformation.

The proposed segmentation algorithm consists of three levels of segmentation: lines segmentation, sub-words segmentation and characters segmentation.

Lines segmentation involves horizontal projection of the image rows to find the empty rows between rows that contain text as in TABLE III.

TABLE III
Lines segmentation algorithm

Step 0	If the current row index is smaller than the max rows index Build up the horizontal projection of this row. If its value equals 0 Go to step1 Else Go to step 2 Else Go to step 3
Step 1	Store the corresponding row index in array1.
Step 2	Go to the next raw. Go to step 0.
Step 3	In array1, search for more than 5 consecutive indexes If find it Store those indexes in array2

The algorithm output (array2) contains the indexes of separating lines in the processed paragraph.

Words segmentation is the next level to follow after the lines segmentation. Vertical projection profile of image's

columns is performed to each text line in-order to divide it into sub-words.

Characters segmentation is the third level of the segmentation stage its algorithm is based on an important feature of cursive Arabic text (junction line).

Junction line is a small horizontal line that links between two consecutive characters in one sub-word. Fig. 6 shows junction lines.

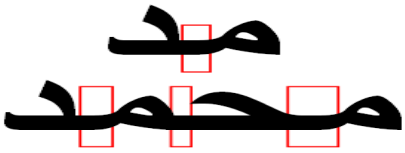

Figure 5 Examples of junction lines between characters in words

Since each character in the word except the first and last character is placed between two junction lines, the proposed algorithm detects each junction line to extract every single character.

The extracted characters are then passed to the recognition stage which consists of two phases feature extraction and classification.

IV. Feature Extraction

In this stage, the proposed system has implemented many features due to the complexity of Arabic characters. Those features can be classified into structural and statistical features.

The implemented structural features are three features. The first two features are the number of connected components and the number of holes which are shown in Figure 6 and Figure 7.

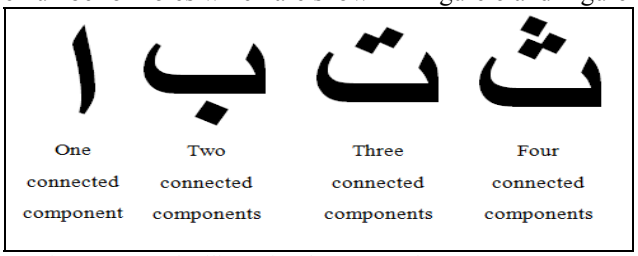

Figure 6 Examples illustrating the connected components concept

The third structural feature is the chain code. It is a technique used to represent a boundary by a connected sequence of straight line segments of specified length and direction. The direction of each segment is coded by using a numbering scheme such as the one shown in **Figure 8**. Chain codes based on this scheme are refereed as freeman chain codes [2], [4], and [5].

Each character is represented by a unique vector relative to its boundary shape. An example is illustrated in Figure 9 which represents the kaf letter 'ك ' vector:

[7,6,6,6,6,6,6,6,6,6,6,6,6,4,4,4,4,4,4,4,4,4,4,4,2,2,2,6,6,6,7,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,2,2,2,2].

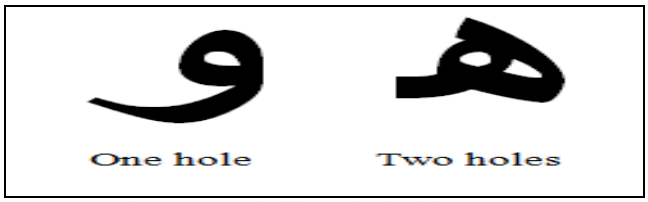

Figure 7 Examples illustrating the holes concept

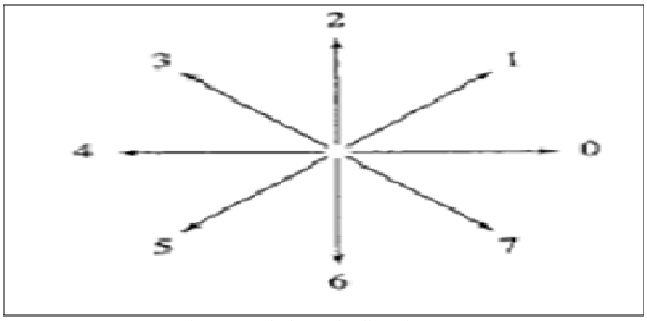

Figure 8 Chain code directions

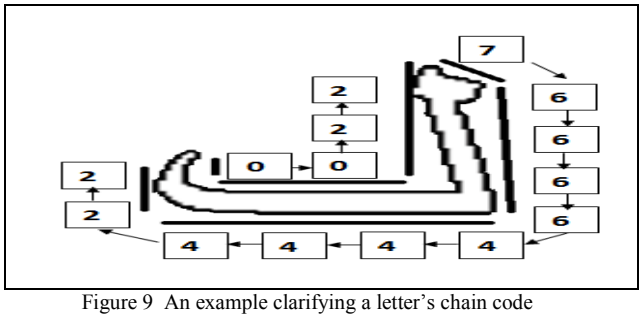

Figure 9 An example clarifying a letter's chain code

The implemented statistical features are the horizontal and vertical Projection. They can be a good descriptor to uniquely identify letters, for example for the 40x43 image below both Y-projection (rows) vector and X-projection (columns) vector are unique, which is simply the sum of white points in each row and vector respectively. Figure 10 explains these features.

e.g,Ypr=[0,0,0,0,0,0,0,0,0,0,0,1,2,4,4,3,4,5,6,10,27,27,27,18,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0].

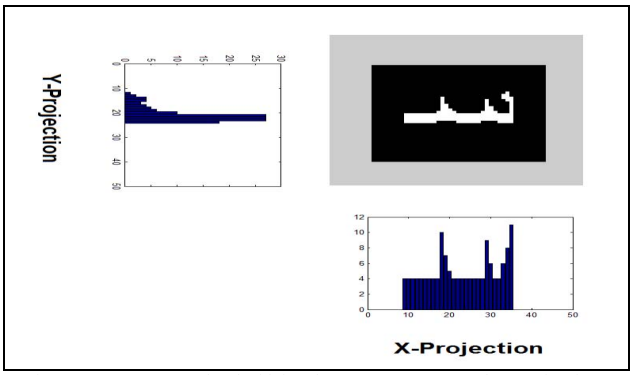

Figure 10 An example clarifying letter seen 'س' X and Y projections

V. DATABASE AND CLASSIFICATION

The database was generated and saved before the testing procedure, the patterns in TABLE I was prepared. Templates were written using Time New Roman font type. Templates have been processed with the same image processing techniques that were used to process input image as clarified in Section III.

Features that have been clarified in the previous section were extracted from each pattern in TABLE I, and saved in the database.

The database was constructed from eight look-up tables, grouped according to the number of connected components and the number of holes as TABLE IV shows.

TABLE IV
Letters Groups

2 0	ب , ج , خ , ذ, ز, غ, ك, ن
2 1	ض ظ ف
3 0	ت ,ي
4 0	ث , ش
1 0	ا,ح, د , ر, س , ع ل م
1 1	ص , ط , و
1 2	هـ
3 1	ق

Each look-up table is a structure that was constructed from the following fields:

- Letter name.
- Letter value, each character has a corresponding unique value assigned to it.
- Letter horizontal and vertical projection profiles.
- The chain norm vectors of the letter.

The database entries were generated automatically using a developed piece of code. In-order to generalize the system scope, the developed code can be used to generate database of other font types.

Classification is responsible for classifying the incoming pattern by comparing between the extracted features from the feature extraction phase and the features stored in the look-up table. The result is then sent to a text file.

VI. EXPERIMENTAL RESULTS

This phase is accomplished by two main parts. The first is performed in letters while the second is performed in words.

In the first part, a testing sample of 1240 images is prepared. These images contain the Arabic letters written in multiple fonts as well as in multiple sizes. The selected fonts are the Arial, Courier New, Traditional Arabic and Tahoma. The purpose of this test is to ensure the system ability to recognize other fonts that are similar or different from the system templates font, Time New Roman. Recognition percentage in each case has been illustrated in Figure 11.

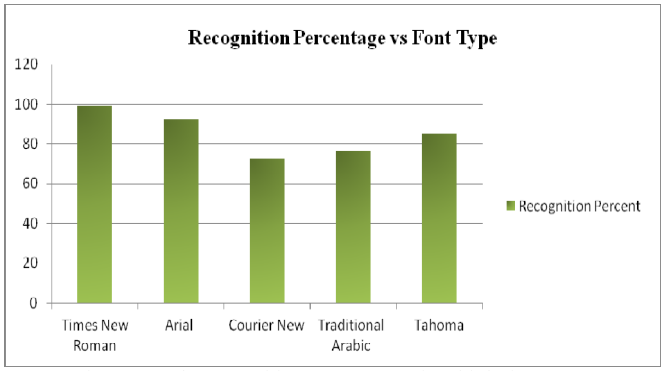

Figure 11 The recognition percentage of multiple fonts

The relatively high recognition rate of time new roman font's letters is due to that the database stored fields were extracted from Time New Roman templates. The degradation in the other fonts caused by the variation in letter's shapes from font to font as shown in **Error! Reference source not found.** Another reason for misclassifications come from that inside one classification group, some letters have a similarity in the main body shape. In group '202', letters"خ ,غ " have similar shape of main bodies. The same scenario is in group '101', letters pair" ع ,ح " and letters pair "د, ر" in small sizes have similar boundary shapes that will affect the final decision of classification.

Figure 12 Example of letters that have different shapes according to the font type.

In the second part, selected words samples are passed to the system. The segmentation algorithm shows a very high accuracy, the proposed line segmentation and sub-word segmentation gave excellent results, see **Error! Reference source not found.** However the character segmentation algorithm has introduced some problems, since it does not differentiate between the junctions that occur between letters and some other junctions that occur in the main body of the letter. Hence it affected the recognition rate negatively. Letters that are composed from junction lines are exposed to this problem. This appears obviously in the case of letter seen 'س' and letter 'د'. The first case is recognized as two alef 'ا'and lam 'ل'.

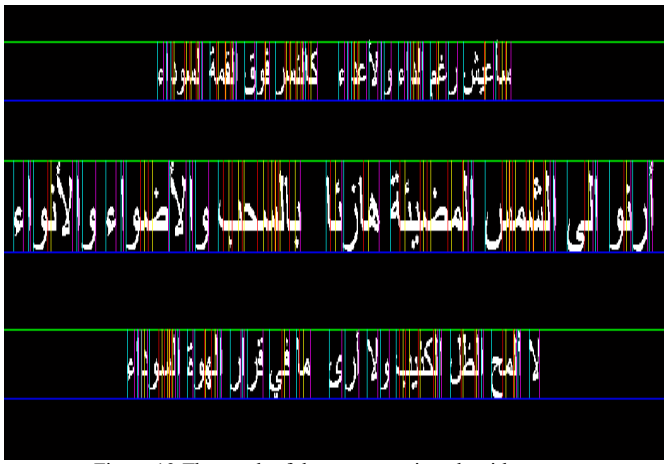

Figure 13 The result of the segmentation algorithm

The selected samples of words show misclassification results due to the misclassification of letters reasons. Another reason is the overlapping problem that introduced by letters: “ح”,”خ”,”ج”,“ر”,“ز”,”ع”,”غ”and“و”. However there are many fonts including Time New Roman font that do not suffer from this problem.

VII. FUTURE WORK

Some refinements can be added to the proposed system so that numbers and symbols can be recognized. Also, more fonts can be added easily by adding their characters features to the database. Each font look-up tables can be generated automatically. Skew angle detection and correction and other image processing techniques can be added to the pre-processing stage to avoid the effect of rotation or other homogeneous transformation.

The work extension for handwritten recognition can be done by using one of the machine learning methods. Artificial neural network (ANN) and hidden Markov models (HMM) are the most common used.

ACKNOWLEDGMENT

The authors would also like to thank M. El-Haj, M. Abdul-Hakam, M. Hussain, S. Babiker and A. M. Ameen for their great help and support that reflected positively on this work.

REFERENCES

[1] A. Cheung, M. Bennamoun and N.W. Bergmann, “An Arabic optical character recognition system using recognition-based segmentation” Pattern Recognition 34 215-233, 2001.

[2] H. Izakian, S. A. Monadjemi, B. Tork Ladani, and K. Zamanifar,” Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes”, World Academy of Science, Engineering and Technology 43 2008.

[3] A. M. Ameen ,”An OCR System for Arabic Handwriting", University of Khartoum, 2010.

[4] N. A. Jusoh, and J. M. Zain," Application of Freeman Chain Codes: An Alternative Recognition Technique for Malaysian Car Plates", IJCSNS International Journal of Computer Science and Network S ecurity, VOL.9 No.11, 222-227, November 2009.

[5] Rafael C. Gonzalez, Richard E. Woods, and Steven L. Eddins, Digital Image Processing using MATLAB, 2004.

[6] S. A. Mahmoud, H. A .Al-mohtaseb and S. S. Qawaji, “Recognition of off-line printed Arabic text using Hidden Markov Models”, Signal processing 88, pp. 2902–2912, 2008.

[7] S. A. Hussien Al-Qahtani, “Recognizing cursive Arabic script using Hidden Markov Models”, University of King Saud, 2004.

[8] A. Amin. “OCR of Arabic text”. In The 4th International Conference on Pattern Recognition, Cambridge, UK, pp. 616–625, 1988.

[9] A. Amin and H. Alsadoun, “Hand printed Arabic character recognition system”. Proceedings of the 12th International Conference A on Pattern Recognition, IAPR ; 1994. pp 536–539.

[10] H. Goraine, M. Usher, and S. Al-Emami, “Off-Line Arabic Character Recognition”, Computer, 1992. vol. 25, pp. 71-74.

[11] M.S Khorsheed. and W.F. Clocksin, “Structural Features of Cursive Arabic Script”, Proc. British Machine Vision Conf., 1999. pp. 422-431.

[12] A., Zidouri, S Chinveeraphan, and M. Sato, “Structural Features by MCR Expression Applied to Printed Arabic Character Recognition” in 8th Int. Conf. on Image Analysis and Processing, (San Remo Italy), 1995.pp.557--562, Sept. 13-15.

[13] B. AL -Badr and S. Mahmoud “Survey and bibliography of Arabic optical text recognition”, Signal Processing. 1995. 41(1): 49-77.

[14] H. Sanossian, “An Arabic character recognition system using neural network”. Proceedings of 1996 IEEE Signal Processing Society Workshop, Kyoto, Japan, IEEE1996., pp; 340–348.

[15] I. Bazzi, R .Schwartz, and J. Makhoul, “An omnifont open-vocabulary ocr system for English and Arabic”. IEEE Trans Pattern Analysis and Machine Intelligence. 1999. vol; 21(6): 495–504.

[16] F. Zaki, S. Elkonyaly, A. Elfattah, and Y. Enab, “A new technique for arabic handwriting recognition”. Proceedings of the 11th International Conference for Statistics and Computer Science, Cairo, Egypt, 1986. pp; 171–180.

[17] S .Saadallah, and S. Yagu, “Design of an arabic character reading machine”. Workshop on Computer Processing and Transmission of the Arabic Language, Kuwait. 1985.