Improved Method for Sliding Window Printed Arabic OCR

Prof. Wajdi S. Besbas [1], Dr. Mohamed R. Sunni [2], and Anisa F. Elbokhare [3]

***Abstract*—** In this paper an improved method of printed Arabic character recognition is presented. It is segmentation-free character recognition. A sliding widow with the size of a reference character is used to select a sub-image, from the document image, for the recognition. The proposed procedure starts with highest probability characters in Arabic writing, except for those characters that produce a recognition error. To make the procedure even faster, characters that have same dimensions are grouped in one cluster and recognized together. A previously created database for Arabic characters images is used in this research as reference characters. Five font types and nine font sizes are implemented. The classifications of the unknown characters are carried out by extracting their structural and transform features. The Character height, width, and number of pixels above baseline, together with the Walsh Hadamard Transform (WHT) coefficients are used to construct the features vector.

***Keywords*—**Features extraction, Structural features, WHT features, Segmentation-free OCR, Sliding window for Printed Arabic OCR.

I. Introduction

OPTICAL Character Recognition (OCR) is the process of recognizing handwritten or printed text by computer. Machine replication of human reading is an ancient dream. However OCR has become one of the most successful applications of technology in the field of pattern recognition and artificial intelligence [1]. Many commercial systems for performing OCR exist for a variety of applications, although the machines are still not able to compete with human reading capabilities. The character recognition approaches classified into either on-line or off-line category [2]. Dynamic recognition (On-line) describes the pen path over some period of time. In static recognition (Off-line), documents are available as hard copy. These documents are scanned to obtain two-dimensional character representation. The quality of the scanned image affects the accuracy of the recognition method.

A sample that covers segmentation-free word and character recognition approaches is given in [3] – [6]. An approach that recognizes Arabic alphabet is presented in [3]. The recognition is achieved by simple matching to pre-built prototypes of all the Arabic Character set. The prototype description is based on a method called "Minimum Covering Run Expression". This approach is used for the recognition of one font of the Arabic language. In [4] a knowledge based system called ORAN is presented. It uses a structural description of the text documents, and applying simple matching of candidate characters to reference prototypes, for the recognition. Database of prototypes for each character shape is built from the structural features. The system has been trained with several model documents using printed Naskh font type only. A technique for recognizing Arabic cursive words from scanned images of text is discussed in [5]. It is applied to four different Arabic typefaces. Each word transforms into a normalized polar image, then a two dimensional Fourier transform applied to the polar image. Each word is represented by a set of Fourier coefficients. In [6] a database composed of images for words of Arabic Printed Text is introduced. The purpose of this database is the multi-font, multi-size and multi-style text recognition systems in Arabic. The database is called APTI for Arabic Printed Text Images.

Arabic character recognition covers the recognition of Handwritten and Printed text. The aim of this research is to improve the previously introduced segmentation-free Printed Arabic Optical Character Recognition approach [7]. The classifications of the unknown characters are carried out by extracting their structural and transform features [8],[9]. To improve the recognition speed the highest probability characters, as given by [10], are recognized first, except for those characters that produce a recognition error.

Since there is no freely available database of printed Arabic characters that can be relied upon and used as a reference, a part of this research was the creation of a database for the characters images of the selected fonts with several sizes.

II. Printed Arabic Character Recognition

Arabic is the official language of many countries. Arabic belongs to the group of Semitic alphabetical scripts in which mainly the consonants are represented in writing, while the markings of vowels (using diacritics) is optional and is rarely used. A diacritic is placed above or below the body of the Arabic character such as:

ضَ ضًا ضُ ضّ ضِ ضٍ ضْ ضّ

Arabic characters differ significantly from other characters such as Latin and Chinese characters. These characters are represented by 28 symbols, and are written cursively in both printed and handwritten forms. These characters have two to four different forms, depending on its position in the word,

Wajdi Besbas is Professor with University of Zawia, Faculty of Science, Computer Department, Zawia, Libya.

Mohamed Sunni is lecturer with University of Zawia, Faculty of Science, Computer Department, Libya.

Anisa Elbokhare was postgraduate student at University of Tripoli, Faculty of Science, Computer Department, Tripoli, Libya. She is now Part time assistant lecturer at University of Tripoli.

http://dx.doi.org/10.15242/IIE.E1215031

isolated form (IF), at the beginning (BF), in the middle (MF), and at the end (EF) of a word. These different forms increase the number of classes from 28 to 100 as showing in Table I.

TABLE I
ARABIC CHARACTERS AND THEIR FOUR POSSIBLE FORMS

EF	MF	BF	IF	EF	MF	BF	IF
ـض	ـضـ	ضـ	ض	ـا			ا
ـط	ـطـ	طـ	ط	ـب	ـبـ	بـ	ب
ـظ	ـظـ	ظـ	ظ	ـت	ـتـ	تـ	ت
ـع	ـعـ	عـ	ع	ـث	ـثـ	ثـ	ث
ـغ	ـغـ	غـ	غ	ـج	ـجـ	جـ	ج
ـف	ـفـ	فـ	ف	ـح	ـحـ	حـ	ح
ـق	ـقـ	قـ	ق	ـخ	ـخـ	خـ	خ
ـك	ـكـ	كـ	ك	ـد			د
ـل	ـلـ	لـ	ل	ـذ			ذ
ـم	ـمـ	مـ	م	ـر			ر
ـن	ـنـ	نـ	ن	ـز			ز
ـه	ـهـ	هـ	ه	ـس	ـسـ	سـ	س
ـو			و	ـش	ـشـ	شـ	ش
ـي	ـيـ	يـ	ي	ـص	ـصـ	صـ	ص

Some of these characters have the same shape but are distinguished by the presence of one, two or three dots (secondaries) above or below them such as:

ش س ـنـ ثـ بـ تـ يـ ح خ ج

In some cases, a pair of characters may be combined to form another character, which is referred to as a ligature. The only mandatory ligature is (Lam Alef لا), others are optional such as (ل and ح in الحروف) and (ت and م in تميز). Ligatures complicate the segmentation task of the Arabic OCR systems.

Arabic characters of a word are connected along a baseline. In general, horizontal projection of the text shows clearly the baseline position, while vertical projection can be used for segmenting sub-words and characters, as illustrated by Fig. 1. The baseline represents essential information before feature extraction can be performed. It is proved to be a good reference for the Arabic characters feature selection [3].

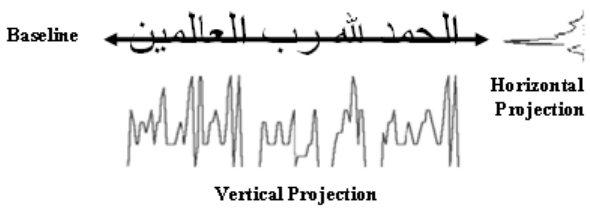


Fig. 1 Vertical and horizontal projections for the input text image

III. CHARACTERS DATABASE SPECIFICATION

The details of the database construction are given in [7], it consists of 45 folders.

- Five font types are selected to construct this database; they are "*Times New Romans*", "*Arial*", "*Arabic Transparent*", "*Simplified Arabic*" and "*Andalus*".
- Nine folders for each font type, which correspond to the nine font sizes (8, 10, 12, 14, 16, 18, 20, 24 and 28).
- Each folder consists of 70 files that represent the Arabic character set in their four possible forms (isolated, beginning, middle, end).
- Each folder also contains a file consists of the characters structural and WHT features (features vector) of the 70 characters images of that font size. As an example, the complete features vector for character (غـ = ga) is given by Table II, and Fig. 2 illustrates the character image, showing the structural features. The padding with zeros is used to make the size of the character image equals to a power of 2 such as 8x8, 16x16, or 32x32, to allow the application of WHT to the character image.

The created database contains images for all characters with their different shapes, sizes and font types (3447 character images).

TABLE II
FEATURES VECTOR FOR CHARACTER غـ

Character	Height	Width	Padding	Number of pixel Above baseline	*Selected WHT spectral coefficients*											
					R_1C_1	R_1C_2	R_2C_1	R_1C_3	R_3C_1	R_1C_4	R_4C_1	R_1C_5	R_5C_1	R_1C_6	R_6C1	R_1C_7
غـ	10	5	16	8	32	32	14	20	-24	20	-6	-6	12	-6	-6	6

Where R_nC_m means the spectral coefficient at row_n $column_m$.

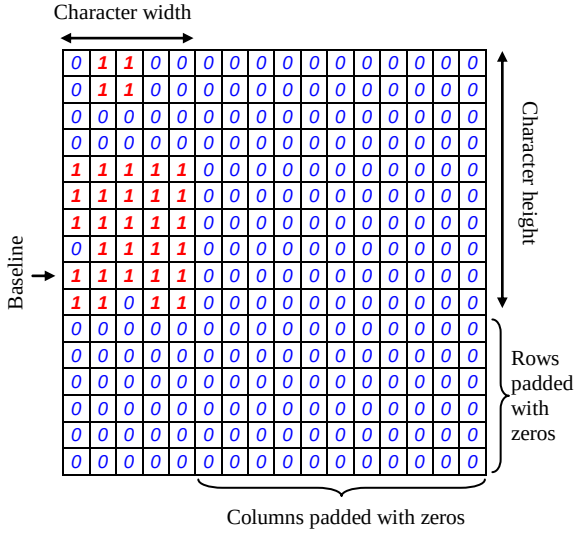


Fig. 2 The structural features for (غـ = ga) character image

Notes:

1. There are originally 100 files in each folder. This number is reduced to 61 files. Because, after removing redundant pixels, some of the characters have similar forms in their different positions (isolated, beginning, middle, end) such

as: Character "ط" has the same form at all positions in the word, and Character "ش" has the same image in the beginning and middle form and has the same image in isolated and end form.

2. Some characters such as ى , لا , ـلا , ـة , ة , ـأ , إ , أ and ـى are added. This makes the total number of files are 70. *For the "Andalus" font type the actual number of files for each font size is 103, because characters that have similar forms in different positions are much less than that for the other font types.*

3. Other characters such as ـبا , لآ , آ , ؤ , ـئ , ئ , ـنـ and ء can be added to the database.

IV. The Previously Proposed Method

The operations performed by our previously proposed method are divided into "Preprocessing the text document" and "The character recognition technique" [7].

A. Preprocessing the text document

Step1. Converting RGB and Gray-scale images to Binary images.

Step2. Drifting correction and baseline detection [11]. Fig. 3a shows a drifted document image, and Fig. 3b shows the result of drifting correction, it also shows the detected baselines using horizontal projection.

Step3. Font type and font size detection [12].

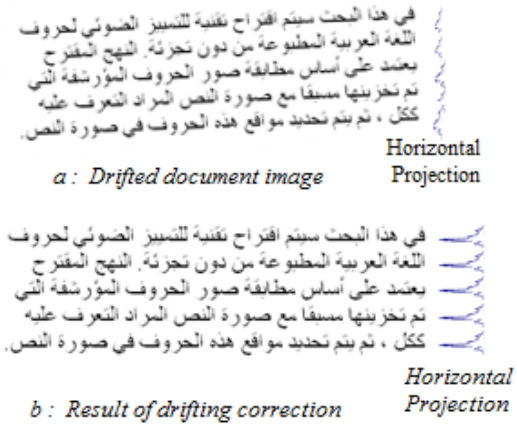


Fig. 3 Drift correction and horizontal projection

B. The character recognition technique

The recognition is carried out by scanning only the baselines of the text image.

Step1. Select a character from the database, and read its width, height, baseline position and its WHT spectrum coefficients.

Step2. Form of the sliding window and scan the text image. The baseline position of the window should be placed on the baseline of the text image and start scanning.

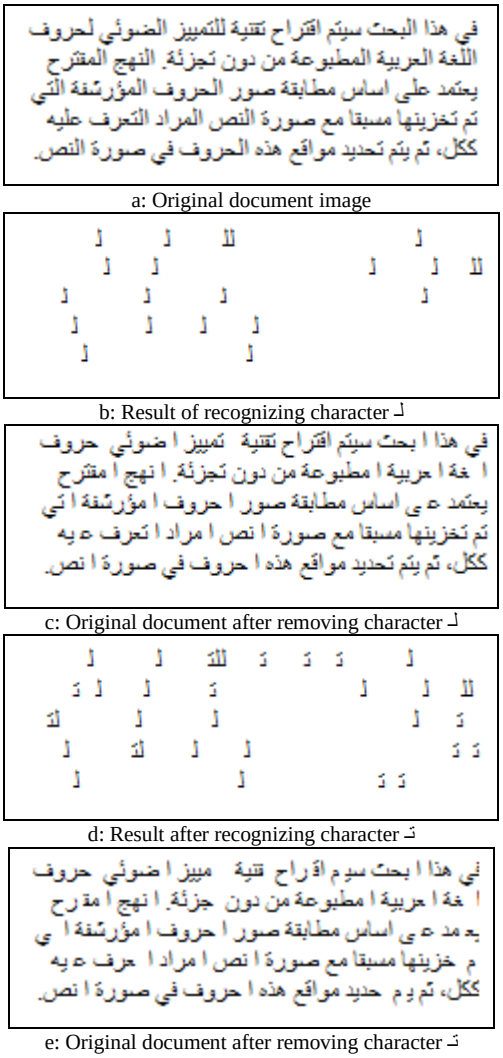
Step3. Extract the character features. Find the WHT for the text under the sliding window. Record the position of the window if the selected coefficients are similar to that of the reference character, and remove all the pixels under the window. Then advance the window to the next position on the baseline.

Step4. Repeat step 3, for the next line of text, until end of document image.

Step5. Put the selected (reference) character at the recorded positions in the output document.

Step6. Repeat starting from step 1.

Fig. 4 illustrates a sample that represents the result of recognizing characters (ـلـ and ـتـ) by the proposed method. The experimental results for the selected font types and font sizes showed that, there are some errors due to the implementation of the recognition technique using the characters in their alphabetical order.



a: Original document image

b: Result of recognizing character ـلـ

c: Original document after removing character ـلـ

d: Result after recognizing character ـتـ

e: Original document after removing character ـتـ

Fig. 4 Practical illustration of Arabic character recognition

V. Improved Arabic Character Recognition

This section describes the improvements that are made to the method introduced in the previous section. These improvements are with respect to execution time and reducing the possibility of errors.

A. Sorting characters according to their probabilities

The probabilities of Arabic characters in Arabic writing are

given in [10]. Where it is found that letter (ا) occurs in 19.65% of the writing, and letter (ل) occurs in 12.99%. To reduce implementation time of our previously proposed method in [7], Arabic characters are arranged according to the highest probabilities as given in [52]. But, there were some miss-recognized characters, and as a result of errors analysis it is found that, although the probability of (ح) is higher than that of (خ - ج), the characters (خ - ج) in all their positions, isolated-beginning-middle-end, should be recognized before the character (ح). The same is true for (ش,س) and (ض,ص) and (ث,ت) . A very high recognition rate is achieved after sorting the characters order of recognition as given in Table III.

TABLE III
SORTING ARABIC CHARACTERS TO REDUCE RECOGNITION ERRORS

1	أ	25	غـ	49	حـ
2	إ	26	عـ	50	جـ
3	ـأ	27	غ	51	نـ
4	ك	28	ع	52	حـ
5	لـ	29	غـ	53	ج
6	ل	30	عـ	54	ح
7	ـا	31	ذ	55	خ
8	ا	32	د	56	ک
9	م	33	ز	57	هـ
10	مـ	34	ر	58	ھ
11	يـ	35	ش	59	ہ
12	ـي	36	س	60	ه
13	ي	37	شـ	61	ى
14	و	38	سـ	62	ی
15	ث	39	ق	63	لا
16	ثـ	40	ف	64	ﻻ
17	ـة	41	قـ	65	ض
18	ة	42	فـ	66	ص
19	ن	43	ف	67	ضـ
20	ت	44	ب	68	صـ
21	تـ	45	بـ	69	ط
22	خـ	46	ق	70	ظ
23	ـخ	47	ج		
24	ـع	48	خ		

B. Scanning for characters that have the same structural dimensions in one go

Characters that have the same dimension (height, width, number of pixels above baseline) will have the same sliding window dimension, which requires an WHT with the same dimension (i.e. 8x8, or 16x16, 32x32) to be implemented for character recognition. As a result of that these characters are scanned in one go, which leads to the reduction of time taken by the recognition procedure to scan and recognize the document characters. Examples of such characters are (غـ , شـ) in their beginning form, character (ـغـ) in its middle form and (د) in its isolated form, for "Times New Roman" font size 12, as shown in Table IV. For smaller font sizes, the group of characters that have the same dimensions are even larger. Characters (نـ , تـ) in their beginning form, character (ـغـ) in its middle form and (د) in its isolated form have the same dimensions for "Times New Roman" font size 8, as shown in Table V.

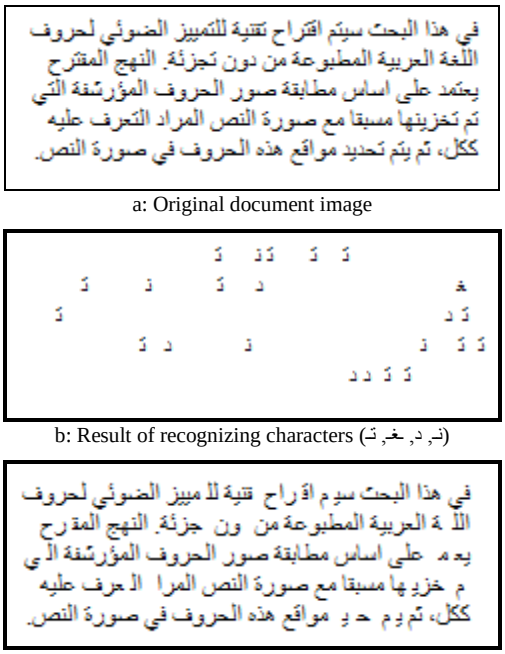
TABLE IV
CHARACTERS WITH THE SAME DIMENSIONS IN TIMES NEW ROMAN SIZE 12

Char	Width	Height	Pixels above base line
غـ	7	8	7
شـ	7	8	7
ـغـ	4	6	5
د	4	6	5

TABLE V
CHARACTERS WITH THE SAME DIMENSIONS IN TIMES NEW ROMAN SIZE 8

Char	Width	Height	Pixels above base line
تـ	3	5	4
ـغـ	3	5	4
د	3	5	4
نـ	3	5	4

The following figure illustrates the result of recognizing characters (نـ, د, ـغـ, تـ) by scanning the document image only once. All four characters have the same dimensions.



a: Original document image

b: Result of recognizing characters (نـ, د, ـغـ, تـ)

e: Original document after removing characters (نـ, د, ـغـ, تـ)

Fig. 5 Practical illustration of the Improved Arabic OCR

C. Some characters combination recognized better as one character image

In some font sizes, some combinations of characters are recognized as one character image better than recognizing each character separately. As an example of such combination, in "Times New Roman" font size 28 the pixels of the combination (الـ) are overlapped and should be recognized as one character image. Fig. 6 illustrates this case.

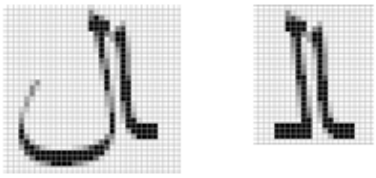


Fig. 6 Pixels overlapping in some characters combination

VI. Experimental Results

This method is applied to all selected font types and font sizes, using the document shown in Fig. 5a. The results with respect to the recognition time, in seconds, are given in Tables (VI - X).

TABLE VI

Font type	**Times New Roman**								
Font size	8	10	12	14	16	18	20	24	28
Recognizing each character separately	1.94	2.55	2.88	8.30	6.49	8.66	11.13	35.41	31.81
Recognizing characters with same dimensions in one scan	1.67	2.19	2.42	7.86	5.44	7.30	10.28	32.29	29.72
Saved time	0.27	0.36	0.46	0.44	1.05	1.36	0.85	3.12	2.09
Percentage of saved time %	13.73	14.12	15.76	5.27	16.15	15.71	7.58	8.83	6.58

TABLE VII

Font type	**Simplified Arabic**								
Font size	8	10	12	14	16	18	20	24	28
Recognizing each character separately	2.44	2.14	3.47	6.81	5.88	8.98	11.16	37.20	34.11
Recognizing characters with same dimensions in one scan	1.80	1.81	2.95	6.06	5.30	7.77	9.45	32.13	32.91
Saved time	0.64	0.33	0.52	0.75	0.58	1.21	1.71	5.07	1.20
Percentage of saved time %	26.29	15.32	14.87	11.01	9.84	13.56	15.27	13.65	3.53

TABLE VIII

Font type	**Andalus Arabic**								
Font size	8	10	12	14	16	18	20	24	28
Recognizing each character separately	3.41	3.05	3.78	5.97	9.98	12.24	19.55	24.77	51.80
Recognizing characters with same dimensions in one scan	2.11	1.99	2.48	4.59	7.24	9.50	15.45	20.34	41.53
Saved time	1.30	1.06	1.30	1.38	2.74	2.74	4.10	4.43	10.27
Percentage of saved time %	38.08	34.89	34.30	23.02	27.53	22.35	20.94	17.86	19.82

TABLE IX

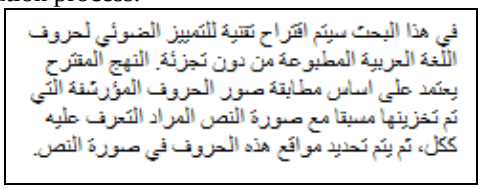
Font type	**Arabic Transparent**								
Font size	8	10	12	14	16	18	20	24	28
Recognizing each character separately	1.69	1.72	3.38	5.94	4.99	8.13	11.53	39.61	36.17
Recognizing characters with same dimensions in one scan	1.41	1.61	2.97	5.44	4.70	7.30	10.49	37.33	34.42
Saved time	0.28	0.11	0.41	0.50	0.29	0.83	1.04	2.28	1.75
Percentage of saved time %	16.65	6.35	12.03	8.41	5.66	10.19	9.08	5.76	4.84

TABLE X

Font type	**Arial**								
Font size	8	10	12	14	16	18	20	24	28
Recognizing each character separately	2.00	2.20	2.91	6.39	6.73	8.89	10.89	38.05	41.84
Recognizing characters with same dimensions in one scan	1.56	1.84	2.56	5.89	5.53	7.86	9.95	36.75	39.34
Saved time	0.44	0.36	0.35	0.50	1.20	1.03	0.94	1.30	2.50
Percentage of saved time %	21.90	16.34	11.84	7.82	17.86	11.60	8.60	3.41	5.97

From the above tables, it is clear that the recognition time is smaller for small font size characters compared to large font sizes. The reason for that is, the document image with small font size characters has fewer number of lines compared with large font size characters, as demonstrated by Fig. 7.

In small font size the number of characters that have the same dimensions is greater than the number of characters in large font sizes, and as a result the percentage of time saved by the improved method is larger for smaller font sizes, as illustrated by Fig. 8. Because transforming the characters using WHT, to extract their features, requires large transform matrix for large font sizes which in turn requires larger time for the transformation process.


في هذا البحث سيتم اقتراح تقنية للتمييز الضوئي لحروف اللغة العربية المطبوعة من دون تجزئة. النهج المقترح يعتمد على اساس مطابقة صور الحروف المؤرشفة التي تم تخزينها مسبقا مع صورة النص المراد التعرف عليه ككل، ثم يتم تحديد مواقع هذه الحروف في صورة النص.

a:Times New Roman size 8


في هذا البحث سيتم اقتراح تقنية للتمييز الضوئي لحروف اللغة العربية المطبوعة من دون تجزئة. النهج المقترح يعتمد على اساس مطابقة صور الحروف المؤرشفة التي تم تخزينها مسبقا مع صورة النص المراد التعرف عليه ككل، ثم يتم تحديد مواقع هذه الحروف في صورة النص.

b: Times New Roman size 14

Fig. 7 Varying document size for deferent font sizes

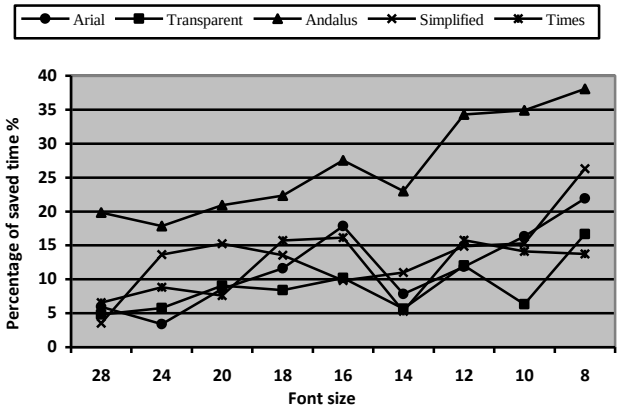


Fig. 8 Percentage of saved time using Improved Arabic OCR

VII. Conclusion

An improved segmentation-free printed Arabic character recognition technique is presented. A sliding widow with the size of a reference character is used for the recognition. A database for Arabic characters images, five font types and nine font sizes with their different positions in the word (isolated, beginning, middle, and end), is used. Walsh Hadamard Transform (WHT) is utilized for the characters images features extraction. The improved method reduced the recognition errors by re-arranging the characters independent of their alphabetical order. Reducing the execution time is achieved by re-arranging characters according to their probabilities in Arabic writing, and recognizing characters that have same sliding window dimensions simultaneously.

References

[1]. Z. Shaaban, "***A New Recognition Scheme for Machine-Printed Arabic Texts based on Neural Networks***", World Academy of Science, Engineering and Technology, 41, 2008.

[2]. B. M. F. Bushofa, M. Spann, "***Segmentation and recognition of Arabic characters by structural classification***", Image and Vision Computing, 15 (1997) http://dx.doi.org/10.1016/s0262-8856(96)01119-5

[3]. A. Zidouri1, M. Sarfraz, "***On Optical Character Recognition of Arabic Text***", The 6th Saudi Engineering Conference, Dhahran, Vol. 4, December 2002

[4]. A. Zidouri, "***ORAN System: A Basis For an Arabic OCR***", The Arabian Journal for Science and Engineering, Volume 31, Number 1B April 2006.

[5]. M. S Khorsheed , W. F. Clocksin,***"Multi-Font Arabic Word Recognition Using Spectral Features",*** 2000 IEEE, pp(543-546)

[6]. F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi, J. Hennebert, ***"Database and Evaluation Protocols for Arabic Printed Text Recognition",*** department of informatics research report , Switzerland, February 6, 2009.

[7]. W. S. Besbas, A. F. Elbokhare, **"Segmentation-free Arabic Character Recognition**", International Congress on Engineering and Information (ICEAI 2014), Beijing, China, May 22-24, 2014.

[8]. R. C. Gonzalez, R. E. Woods ,"***Digital Image Processing***", Second Edition, Printice Hall, Sep5, 2003.

[9]. D. F. Elliott, K. R. Rao, ***"Fast Transforms, Algorithms, Analysis, Applications",*** Academic Press, 1982.

[10]. Excel sheet about Frequency distribution of letters in Arabic from *http://www.al-shamaa.com/php/arabic/#ArStats*

[11]. A. AL-Shatnawi, K. Omar, "***Methods of Arabic Language Baseline Detection – The State of Art***", International Journal of Computer Science and Network Security, IJCSNS, VOL.8 No.10, October 2008.

[12]. I. S. I. Abuhaiba, "***Arabic Font Recognition using Decision Trees Built from Common Words***", Journal of Computing and Information Technology - CIT13, 3, 2005. http://dx.doi.org/10.2498/cit.2005.03.04