Applied Mathematics, 2013, 4, 1313-1319
http://dx.doi.org/10.4236/am.2013.49177 Published Online September 2013 (http://www.scirp.org/journal/am)

A Fast Recognition System for Isolated Printed Characters Using Center of Gravity and Principal Axis

Ahmed M. Shaffie, Galal A. Elkobrosy
Department of Engineering Mathematics and Physics, University of Alexandria, Alexandria, Egypt
Email: amshaffie.oq@gmail.com

Received January 22, 2013; revised February 22, 2013; accepted March 2, 2013

ABSTRACT

The purpose of this paper is to propose a new multi stage algorithm for the recognition of isolated characters. It was similar work done before using only the center of gravity (This paper is extended version of "A fast recognition system for isolated printed characters using center of gravity", LAP LAMBERT Academic Publishing 2011, ISBN: 978-3-8465-0002-6), but here we add using principal axis in order to make the algorithm rotation invariant. In my previous work which is published in LAP LAMBERT, I face a big problem that when the character is rotated I can't recognize the character. So this adds constrain on the document to be well oriented but here I use the principal axis in order to unify the orientation of the character set and the characters in the scanned document. The algorithm can be applied for any isolated character such as Latin, Chinese, Japanese, and Arabic characters but it has been applied in this paper for Arabic characters. The approach uses normalized and isolated characters of the same size and extracts an image signature based on the center of gravity of the character after making the character principal axis vertical, and then the system compares these values to a set of signatures for typical characters of the set. The system then provides the closeness of match to all other characters in the set.

Keywords: OCR; Pattern Recognition; Confusion Matrix; Image Signature; Word Segmentation; Character Fragmentation

1. Introduction

Character recognition is not a difficult task for human beings. The purpose of optical character recognition (OCR) systems is to recognize the characters from images. OCR systems have been divided into two categories, namely: on-line and off-line techniques. On-line techniques are mainly dependant on the motion of hand while the characters are being written; hence this technique is mainly used in the recognition of hand written documents. One of the main problems of that technique is that it cannot be used for already written documents and for printed characters and its need for special digitizers or PDA where sensors pick up the pen-tip movements as well as pen-up/pen-down switching. There is multiple papers exist explaining it, for which, a well formed survey is given by Nouboud *et al.* [1]. On-line techniques provide better results than off-line techniques as it uses a significantly larger set of information which is not available for off-line techniques which are only dependent on stored images. However, off-line techniques are the only techniques available for hard copy and printed papers. Cowell *et al.* [2] give a procedure for an OCR system. This paper uses the same schedule but it uses the feature of center of gravity instead of counting pixels in rows and columns as Cowell *et al.* [2]. The reason for using this different technique is because the pixel counting method is a very exhaustive technique as it requires passing through every pixel and provides no methods of improving the recognition and to decrease the confusion between characters as proposed in this paper by using the center of gravity. Tabassam Nawaz *et al.* [3] use the same schedule of Cowell [2] but change the method of extracting the features to be the number of consecutive ones and zeros using a method called "Chain Code". However, this method also requires a lot of calculations and processing. Multiple different techniques exist for character recognition such as "structural information" as number of holes and strokes, but these techniques cannot be used for every character set and have to be revised completely for each different character set as there is a lot of style of characters such as English, Arabic and

Chinese characters, every one of them has its own characteristics. In the Arabic character set, one of the main features is dots. Up to 3 dots can exist for Arabic characters, and hence no one criteria can be used to apply for all of these character sets. Some techniques for these styles are applied for Arabic characters by Cowell *et al.* [4] using “thinning” and “feature extraction”, however, that technique was slow and cannot be modified to another character sets easily. One of the main problems of the techniques used by OCR systems is that the character is wrongly identified, so the features can be tested by building a confusion matrix Cowell [5] to determine whether this technique is good for this character set or it will lead to a problem in the recognition phase. And it has been derived a way to resolve this conflict Cowell [6]. The proposed work here gets its importance from its scale and rotation invariant and of course because its calculations are minimum while its accuracy is very good and can be tuned by doing more calculations to get more accuracy.

2. An Overview of the Proposed System

The paper’s approach in recognition makes use of five phases as outlined below:

- Read input image.
- Line and characters segmentation.
- Normalize character to a standard size, 100 × 100 pixel resolution has been used in the implementation.
- Extract the character signature.
- Compare the character signature with the signature templates of the character set.

2.1. Text Image and Text Line Segmentation

The segmentation of the image is done at two levels. First, the text in the image is split into lines of text using the horizontal projection technique (*i.e.* location of horizontal lines of zero density of pixels, given the line of text from the horizontal projection technique, indicates the beginning of a segment and the subsequent location of another zero density line of pixels indicates the end of a segment, thus an entire segment is located). Second, each line of text is split into characters using vertical projection technique (*i.e.* location of vertical lines of zero density of pixels, given the line of character from the vertical projection technique, indicates the beginning of a segment and the subsequent location of another zero density line pixels indicates the end of a segment, thus an entire segment is located) [7]. **Figure 1** illustrates the text image to text lines segmentation and the text line to individual character segmentation.

2.2. Normalization of the Fragmented Characters

One of the important stages in the process is to insure that the input character has the same dimensions and the same orientation as the characters used to create the signature or the configuration file, so the procedure start by unified the orientation of the characters by using the character principle axis, so the normalization started by getting the principle axis of the character and rotate the character image to make the principle axis vertical, and then trimming the white parts of the character image then it is scaled to 100 × 100 pixels. **Figure 2** shows the case of Arabic character alef and mim. The figure on the right shows the scanned character as scanned and the figure on the left shows the character after it expanded so that it has the size required and therefore touches each side on the 100 × 100 square as shown [2].

2.3. Get the Center of Gravity

The signature of each character is produced by getting the center of gravity of the normalized character as if the character is a uniform body and the center of gravity coordinate XG, YG is calculated using the following formula

$$XG = \text{thegma } xi/n$$

$$YG = \text{thegma } yi/n$$

where:

n is the number of pixels.

x, y is the coordinate of the black pixels in the image of the character.

This approach is applied to the Arabic characters as

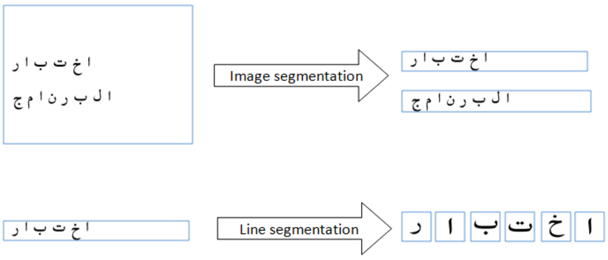


Figure 1. Image and line text segmentation.

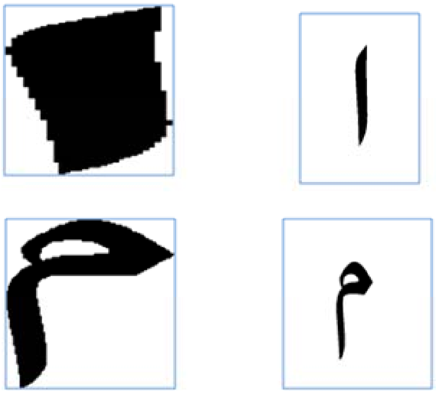


Figure 2. Normalized characters and other original form.

the algorithm treats each character as a body and attempts to get the center of gravity for it; a major challenge with this technique is that more than one character have the same center of gravity or that two centers of gravity are approximately the same. A solution for this problem is provided by dividing the character into 4 bodies and in some cases into 9 bodies and getting the centers of gravity of the divided bodies and then use these points to identify the character. It is found that, when the character is divided, there is no conflict between characters signature and so this signature can be used successfully in recognition. **Figure 3** shows the center of gravity of a character and then show the center of gravity after the character is divided into 4 and 9 segments.

3. The Implementation of the System

The system described above has been fully implemented using Java in two processes. First, learning process at which the configuration file is created. The configuration file contains the characterset signature and determine the number of segments used to get the signature, the input of this process is the full character set and then the system normalizes every character and gets the center of gravity based on the number of segments, the output is the configuration file which contains the number of segments and each character signature. **Figure 4** shows the structure of the proposed learning process.

Second, the recognition process of the input image; the program separates the image first into lines, then to individual characters and asks for the configuration file that the program will use to recognize the characters in the image and then classify the characters by getting the nearest character in the configuration file by calculating squared equilidian distance to all the character set and get the minimum distance. **Figure 5** shows the structure of the proposed OCR system.

4. The Confusion Matrix

The closeness of match between every character can be calculated and is put in a matrix called the "confusion

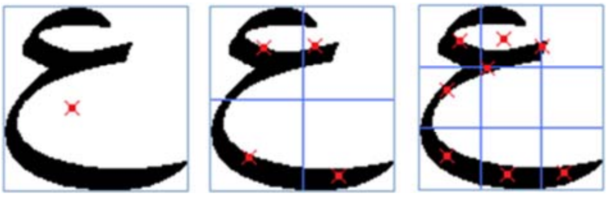


Figure 3. The center of gravity location of ein Arabic character usnig 1, 4, 9 segmentation model.

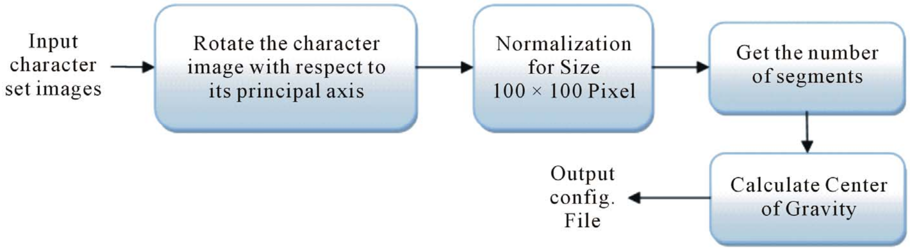

Input character set images
Rotate the character image with respect to its principal axis
Normalization for Size 100 × 100 Pixel
Get the number of segments
Calculate Center of Gravity
Output config. File

Figure 4. The structure of the proposed learning process.

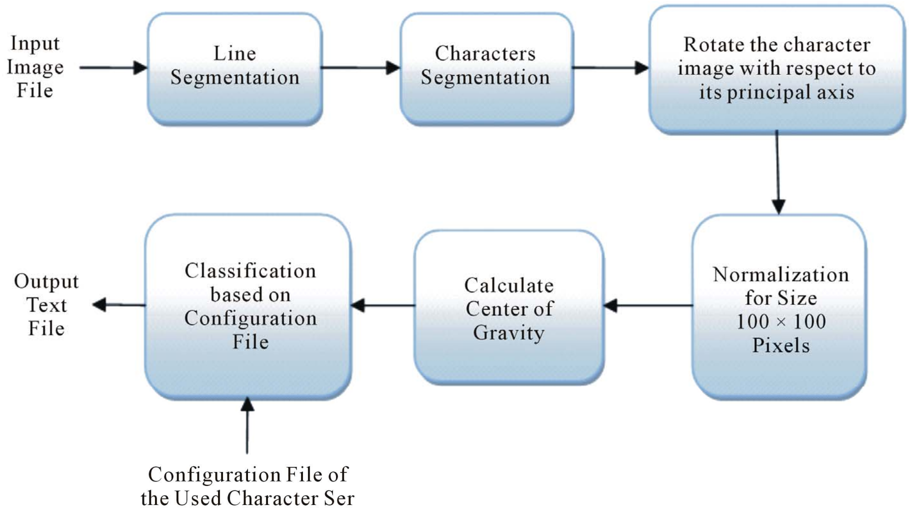

Input Image File
Line Segmentation
Characters Segmentation
Rotate the character image with respect to its principal axis
Normalization for Size 100 × 100 Pixels
Calculate Center of Gravity
Classification based on Configuration File
Output Text File
Configuration File of the Used Character Ser

Figure 5. The structure of the proposed OCR system.

matrix" [2,5]. The application has a tab to get the confusion matrix for the selected configuration file to provide indications about the suitability of using this configuration file in recognition, but in some cases there are more than one character that has nearly the same signature and so the number of segments can be changed in order to solve this conflict.

5. Screen Shoots of the Implemented Program

The program consists of three tabs. First, recognition tab in **Figure 6**, this tab is used for loading the image which contains the characters which has to be recognized, then a configuration file which contains the signature of the character being loaded, and then the characters are recognized and written in the selected output file. second, learning tab in **Figure 7**, at this tab, the "Load Alpha" button is used to load the character set images, and after that the "Scale & Trim" button is used to normalize all characters, then from the List of Values the "number of segments" is selected, and the center of gravity is calculated and saved in a configuration file to use it later in the recognition tab. Third, confusion matrix tab in **Figure 8**, in this tab a configuration file is loaded and the program calculates the distance between each pair of the characters in the character set to show them in the grid and to know if this configuration file can be used efficiently or not.

6. Performance Evaluation

In this part we will conduct with a blackbox evaluation for two Arabic OCR products which are our proposed OCR system and Cowell OCR system. Our proposed system and Cowell system deal with a character image with resolution 100 × 100 pixels and of course this will make the comparison process very easy and very expressive.

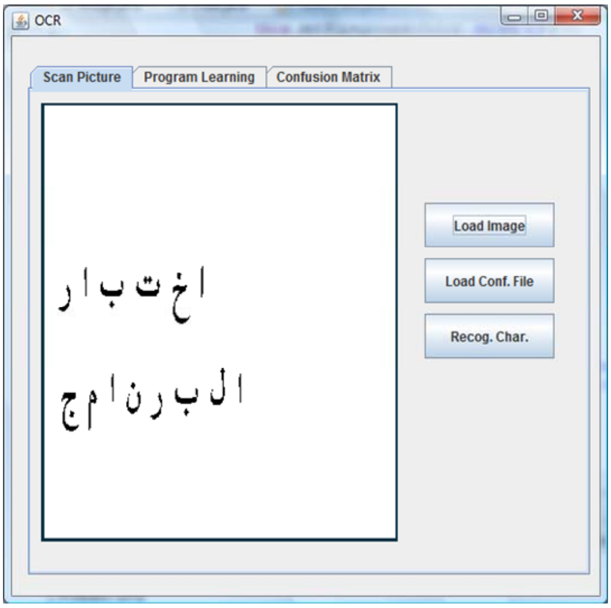
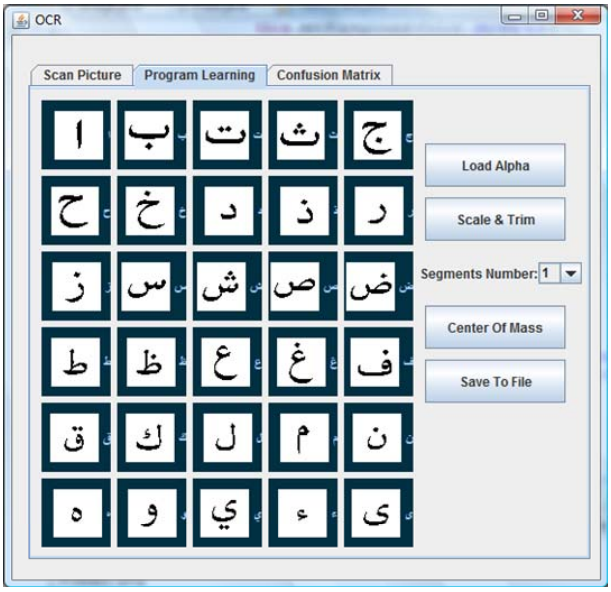
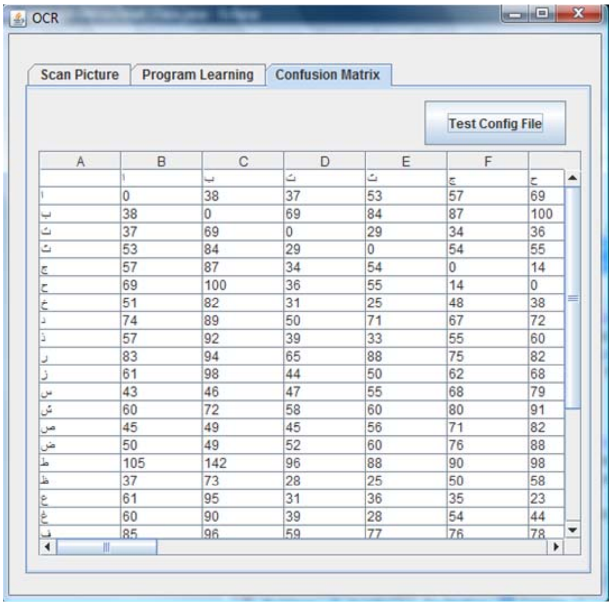


Figure 6. Recognition tab.



Segments Number: 1

Figure 7. Learning tab.



A	B	C	D	E	F	
	ا	ب	ت	ث	ج	ح
ا	0	38	37	53	57	69
ب	38	0	69	84	87	100
ت	37	69	0	29	34	36
ث	53	84	29	0	54	55
ج	57	87	34	54	0	14
ح	69	100	36	55	14	0
خ	51	82	31	25	48	38
د	74	89	50	71	67	72
ذ	57	92	39	33	55	60
ر	83	94	65	88	75	82
ز	61	98	44	50	62	68
س	43	46	47	55	68	79
ش	60	72	58	60	80	91
ص	45	49	45	56	71	82
ض	50	49	52	60	76	88
ط	105	142	96	88	90	98
ظ	37	73	28	25	50	58
ع	61	95	31	36	35	23
غ	60	90	39	28	54	44
ف	85	96	59	77	76	78

Figure 8. Confusion matrix tab.

6.1. Transformations Invariance

Our proposed system and Cowell system dial with a character image normalizing the character image and resizing it to resolution 100 × 100 pixel so this will make both system translation and scale invariant , but Cowell system will face a big problem if the character image is rotated a certain angle this will show that Cowell system is not rotation invariant. Our proposed algorithm is rotation invariant as we first rotate any character image to make its principal axis vertical and this will guarantee that anycharacters have the same shape will have the same principal axis and by the way have the same orientation. For example Cowell system failed torecognize the image in **Figure 9** but our proposed system can recognize all the characters in this text image.

6.2. Number of Comparisons

Of course the number of comparisons while recognizing any text image is very important as it affects the execution time of the system. In Cowell system it depends on comparing the number of black pixels in each row and in each column and treat with this numbers as the signature of the character, So when we try to recognize a single character we have to compare 200 number (100 number which denote the number of black pixels in each row and 100 number which denote the number of black pixels in each column). In our proposed system we get the signature by getting the center of mass of the character and so we have one point coordinate it means that we reduce the number of comparisons to only two comparisons. We add an option as mentioned before to increase the accuracy by dividing the character image into 4 or 9 images and get the center of mass to every part. Now in case we divided the character image to 4 parts we will have 8 comparisons and if we divided the character image to 9 parts we will have 18 comparisons. **Figure 10** shows the relation between the number of comparisons and the number of characters in the text image when we use Cowell system and our proposed system with different number of segments.

6.3. Execution Time

One of the very important factors in any OCR system is the execution time. Cowell claim that his technique can identify in the region of 100 characters per second and we test our proposed system and we get in the region of 250 characters per second. **Figures 11** and **12** show the relation between the number of characters in the page and the execution time it take in milliseconds when we use 9 segments characters and 4 segments characters respectively.

6.4. Resolution of the Image and Font Size

Of course when we increase the resolution of the scanned image, and the font size we get more details about the character. So the effect of the font size is also shown. Our proposed algorithm will be applied on different images each one has different font size and different resolution of image. From the results, we concluded that, using 300 dpi resolutions is the best choice giving average accuracy about 98% and no need to use resolution more

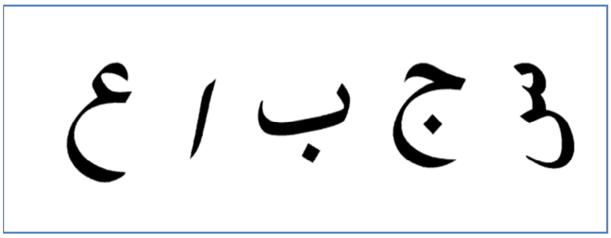


Figure 9. Text image contained some characters rotated different angles.

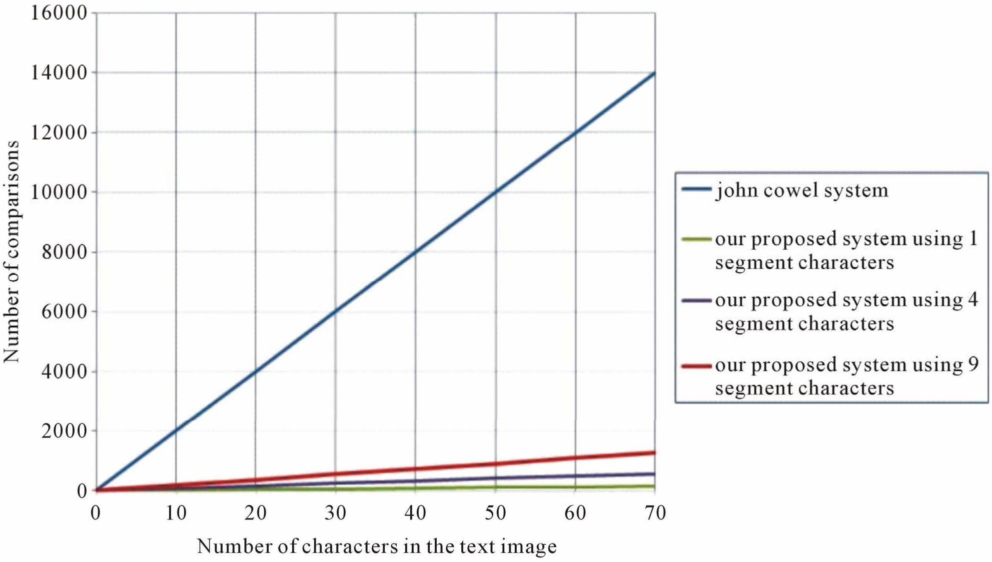


Figure 10. Comparing the number of comparisons in john cowel system and our proposed system using 1, 4 and 9 character segments.

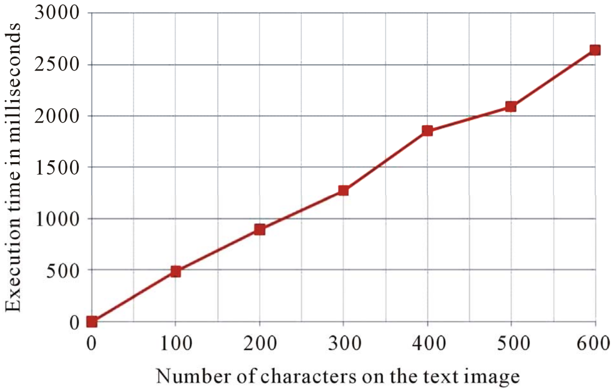


Figure 11. Number of characters execution time using 9 segment characters.

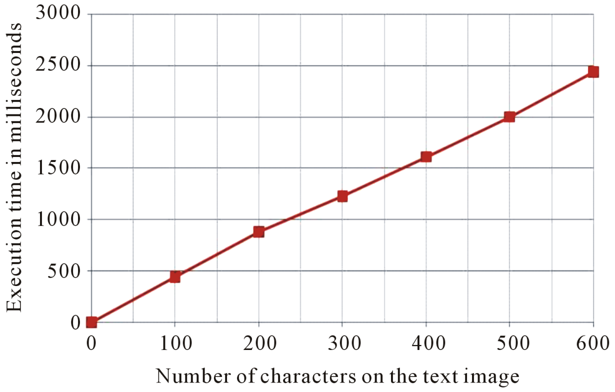


Figure 12. Number of characters execution time using 4 segment characters.

than 300 dpi because it will need more execution time. And the imagesize will be very large (more than 1.5 MB). **Table 1** shows the result of using different image resolution with different font size. The accuracy is calculated by dividing the number of right recognized characters by the total number of characters in the input text image.

7. Conclusions

This paper describes a fast recognition system based on creating images signatures which can be used for any character set. The Arabic character set is used here but this method can be used in any character set. The system used is very rapid, as it uses the center of gravity of the character and then calculates the distances to all the characters to get the nearest one which is selected as the recognized character. This method starts by normalization and done by scaling the character to standard size and rotating the character image in order to make its principal axis vertical and this normalization guarantees that this signature is scale and rotation invariance. The performance and accuracy of this technique can be tweaked by changing the number of segments each character is divided into. The confusion matrix gives an indicator to how much the characters are near to each other.

Table 1. Our proposed system accuracy using different font size and different image resolution.

Font size	Image Resolution			
	150 dpi	200 dpi	300 dpi	400 dpi
11	62%	83%	92%	96%
12	70%	84%	92%	96%
14	72%	85%	94%	98%
16	73%	85%	94%	98%
18	73%	90%	96%	98%
20	76%	90%	96%	99%
22	76%	93%	97%	99%
24	78%	93%	97%	99%
26	78%	93%	99%	99%
28	80%	93%	99%	99%
36	80%	94%	99%	100%
48	80%	96%	100%	100%
72	83%	98%	100%	100%

Additional work can be done in the part of segmentation and fragmentation of the characters especially in the Arabic character as our proposed algorithm dealt with isolated characters; however, Arabic characters in practice are not isolated. And there is a lot of features can be added to the signature such as the number of dots and its position which can help in decreasing the number of segments used and so the program will be faster and at the same time more accurate.

REFERENCES

[1] F. Nouboud and R. Palmondon, “On-Line Recognition of Handprint Characters,” *Pattern Recognition*, Vol. 23, No. 9, 1990, pp. 1031-1044. doi.org/10.1016/0031-3203(90)90111-W

[2] J. Cowel and F. Hussain, “A Fast System for Isolated Arabic Characters,” *Proceeding of the* 6*th International Conference on information Visualisation IEEE*, London, July 2002, pp. 650-654.

[3] T. Nawaz, S. A. H. S. Naqvi, H. Rehman and A. Faiz, “Optical Character Recognition System for Urdu (Naskh Font) Using Pattern Matching Technique,” *International Journal of Image Processing*, Vol. 3, No. 3, 2009, pp. 92-104.

[4] J. Cowel and F. Hussain, “Thinning Arabic Characters for Feature Extraction,” *IV 2001 Proceedings IEEE Conference on Information Visualization* 2001, London, July 2001.

[5] J. Cowell and F. Hussain, “The Confusion Matrix: Iden-

tifying Conflicts in Arabic and Latin Character Recognition," *CGIM* 2000, Las Vegas, 2000.

[6] J. Cowell and F. Hussain, "Resolving Conflicts in Arabic and Latin Character Recognition," 19*th Eurographics UK Conference*, London, April 2001.

[7] M. Al-A'ali and J. Ahmed, "Optical Character Recognition System for Arabic Text Using Cursive Mutli-Directional Approach", *Journal of Computer Science*, Vol. 3, No. 7, 2007, pp. 549-555. doi.org/10.3844/jcssp.2007.549.555.