Journal of Engineering and Applied Sciences 12 (Special Issue 10): 8851-8857, 2017
ISSN: 1816-949X

Arabic Characters Identification Depending on Extracted Features of the Projection-Based Segmentation

Fryal Al-Razaq, Safa Al-Murieb and Sura Mohammed
Information Technology College, Babylon University, Hillah, Iraq

Abstract: The need to establish powerful Arabic characters recognition becomes high in these days, especially; many Latin character recognition systems reach to high accuracy whereas Arabic ones still lower. The identification of Arabic characters is particularly difficult because the Arabic text has a property of cursive nature which necessitates segmenting the Arabic text into isolated characters, the matter which represents the major challenge for any Arabic character identification system. The goal of this study is to increase the accuracy of Arabic identification systems. This study composes of four phases; the first one is calle d pre-processing to treat some problems that occurs in scanning, the second one is segmentation process phase. This phase involves the challenge area on that system. So, we try in this study to deal with this problem. The third phase is features extraction we try to implement some recognition techniques on the Arabic characters, the last one is identification phase based on the extracted features. In this study, a developed a techniques to detect the skew angle and using different descriptors to recognize the Arabic characters and comparing the accuracy of each one.

Key words: OCR, feature extraction, character recognition, projection, phase, Arabic

INTRODUCTION

Pattern recognition is considered an important field in computer vision whereas it contributes significantly in many different areas and applications such as in the Optical Character Recognition (OCR) (Leila *et al*., 2012). The maximum goal of any character identification system is making simulation to the vision of person. The aim of character identification system is transferring a scanned text to a machine-editable form (Haraty and Ghaddar, 2004; Lawgali, 2015).

The systems of character identification have role in the process of automation and its advancing, they can progress the computer interaction by the human in most applications such as in automation process in office or email routing, archiving, making e-Books, verification process, banking systems, etc. (Chan *et al*., 2006; Hamami and Berkani, 2002).

In the Arabic text processing that was printed, there is a difficulty due to the Arabic text nature as a cursive and syntax sensitive text. The following Arabic text characteristics are as limitation (Aljarrah *et al*., 2012; AL-Shatnawi *et al*., 2011).

The Arabic text composed of (28 characters) with the 10 numbers. Whereas regarding to the position of a character in a word, its shape is varied in addition to the fact that most characters have two or four varied styles. So that, the classes number that can be recognized is increased to 112 classes instead of 28 classes:

- There are some supplement special characters
- In Arabic text space can be either inter-words or intra-word space
- Some characters have holes like (..., wow, faa, kaff)
- The formation of vowel symbols are an indispensable part of Arabic script
- Arabic letters usually overlap with neighboring letters
- Many Arabic characters have dots and different character can have the same body shape with the differences in the of dots number
- There is a ligature in the Arabic text

MATERIALS AND METHODS

Suggested system: The suggested system composes of four major phases:

- Pre-processing
- Character detection (i.e., segmentation)
- Characteristics (feature) extraction
- Classifying and identifying

The main stages of the system is shown in the following block diagram as in Fig. 1 where the goals of the phases are illustrated in Table 1.

Corresponding Author: Safa Al-Murieb, Information Technology College, Babylon University, Hillah, Iraq

Pre-processing: When scanning a document, the image can be influenced to different factors such as skew and noise.

Table 1: Goals of system's stages

Phases	Goals
Preprocessing	Noise removal, tilt correction, redundant space removal
Segmentation	To divide the scanned text into well defined, clear characters
Feature extraction	To extract the relevant features of the normalized character for unique identification of the character
Classification and recognition	To correctly classify and recognize the text

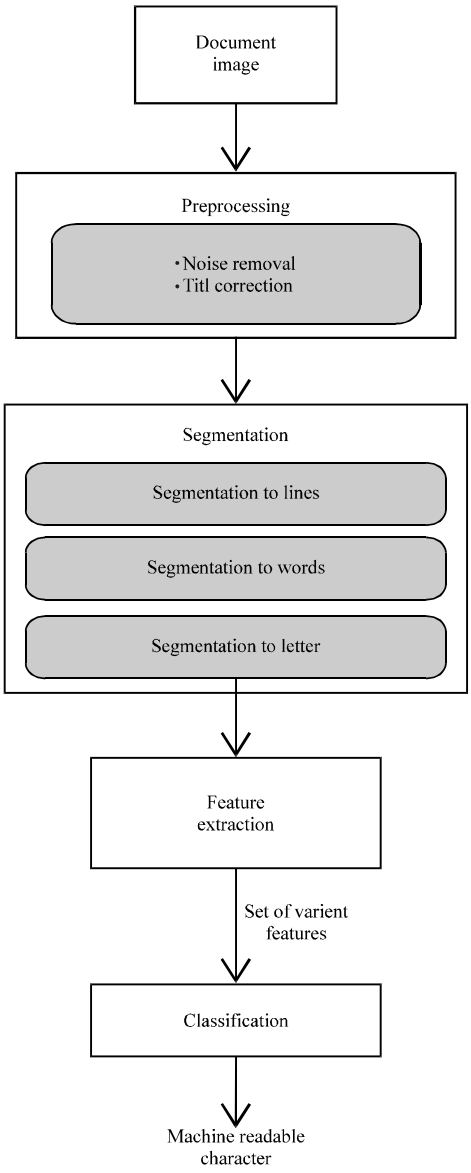

Fig. 1: System's block diagram

Whereas the presence of skew and noise will lead to a very poor segmentation as the Arabic characters are normally connected with an imaginary line which is denoted by (baseline).

Thus, to make the subsequent stages (i.e., segmentation, feature extraction and recognition) easier, it is preferable to remove the noise and discover and correct the skewness in the image of scanned text firstly (Ashkan *et al*., 2006). In this stage, the document is subjected to:

- Thresholding whereby foreground pixels are distinguished from background pixels using a threshold value
- Noise removal process; a process in which pixels that represent noise are identified and corrected, the median filter was used to remove noise from the text
- Redundant space removal whereby the redundant space around the document is cropped

Tilt correction in this process, first, the base line is detected and the tilt angle (θ) is evaluated, after that, the image is rotated by angle ($-\theta$). To correct the angle of tilt, the projection profile based technique was used.

As the baseline has most blackest pixels so that, this property can be useful in allocating the base line in the text. After that, projection and rotation are used for finding the angle of tilt. The technique is as follow: firstly, the start angle was zero with doing the horizontal and vertical projections in order to count the black pixels summation of the rows and columns with recording the maximum of those two sums in addition to the angle (θ).

After that, the angle is decreased by 1 with rotating the document image in clockwise direction and also do projection. If the last computed sum greater than the previous sum, the maximum summation and angle are updated, otherwise trying another angle. The previous steps are repeated until the angle becomes 90.

Then, the document image is rotated either by angle (θ) or angle (θ+90) regarding to the horizontal or vertical maximums, respectively. The following algorithm illustrate the process.

Segmentation: After the image is smoothened the script's segmentation into set of characters is done. In OCR, the segmentation stage is the most difficult, so that, misrecognition or rejection can be caused by a poor segmentation (Al-Hamad, 2013; Hamid and Haraty, 2001).

Character segmentation and classification depends largely on the topographic features as well as extracting the contextual information from them (Shaikh *et al*., 2009; Aouadi and Echi, 2016).

For analysing the cursive writing, the structured description is needed in order to identify the Arabic characters in addition to characters segmentation in the word and individual characteristics detection (Zeki *et al*., 2011; Sarfraz *et al*., 2003; Sahlol *et al*., 2014).

The segmentation stage begins with the whole document, then the following steps are done: first, segmentation of the document to lines; then segmentation of line into words at the end, segmentation the word into characters.

Algorithm tilt correction:

Input: Image of script
Output: Image with tilt correction
Begin
Let q be the rotation angle
Start from q = 0
Maxv = 0; maxh = 0
For q = 0-90
Rotate (Image, q)
Project horizontally at q degree
Project vertically at q degree
If (sumh>maxh) and (sumh>sumv)
maxh = sumh
θ_h = q
End//if
End//for
If maxv>maxh
Rotate (Image, θ_v+90)
Else
Rotate (Image, θ_h)
End//if
If (sumv>maxv) and (sumv>sumh)
maxv = sumv
qv = q
End//if
End

A number of efforts have been devoted to segmentation problem and many algorithms have been proposed on literature.

In this study, three modules were used that do the segmentation, since with considering all possibility of the text of contain diacritics or not also the spacing between the line and words. The first module responses to divided the text into lines and save that lines into separate folder. Firstly, the minimum spacing between lines is computed, then tracing the whole text from the top to down search to the black pixels. If we found the first black one we consider that as the top point of the line. Then, we trace the under point until we found spacing more the minimum one that compute firstly.

The second part includes that segment line into words. The segmented line from the previous module consider as the input for this module. The module compute the minimum spacing between words then cut the word according to that. It starts from right to left and take the first black point as the start point of the word.

Then, it traces the word vertically until it finds spacing more than the minimum spacing, it take this point as the last point of the word. Finally, it cut the word according to these two points and save the word and its position and order on the line within the file name.

The ultimate module involves segmenting the word into its letters. It takes the word, than compute the horizontal projection of that word and then computes the vertical ones according to the horizontal. It concludes the point that should letter cutting from it. In this study, a projection-based method was used whereas the projection was used vertically of the middle region in lieu of making fully projection to the word.

Four regions of text line are identified, they are (upper, middle, baseline and lower) where the zone of baseline has maximum density with black pixels while the middle one is any region that in the above of baseline as shown in Fig. 2. The begin of any new character is considered when any place comes after the connection area (between two characters) that has a larger value. This algorithm was implemented and it introduced a very good accuracy especially with the non-ligatured fonts.

Finally, it save these letters into separated folder and save within file name the position letter on the word the line. Figure 3 clarifies how information of letters (Line, word, its position) are saved after doing the three modules of segmentation. The algorithm works as follows.

Algorithm baseline detection:

Input: Image if scanned documents
Output: Baseline zone
Begin
Project vertically and repeat the above process to segment line to individual parts or words
For each word
Project horizontally and determine the rows with the highest values
This zone of large rows values is the baseline zone
Middlezone = 2* baseline thikness
If vertical projection (middle zone area)
>1/2 (baselline zone) or <2/3 (baseline zone)
The area is a connection area
Else Isolate the character
End//if
End//for
End

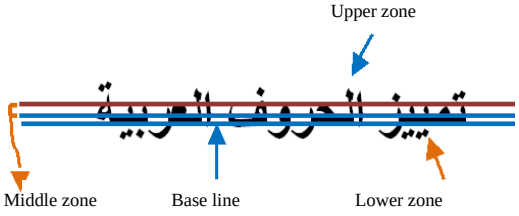

Fig. 2: Line zone

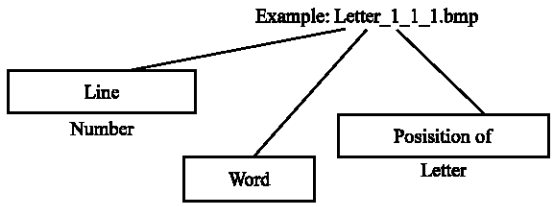

Fig. 3: An example of letter file name and its information

Feature extraction: Since, that pattern recognition is concerned with the automatic detection or classification of objects, it is an integral part of image processing and finds its applications in biometric and biomedical image diagnostics to document classification, remote sensing, and so on. So that, a set of relevant features must be required, so that, such features are extracted after segmentation phase, they characterize with their invariant with the transformation operations such as rotation, translation and scaling. The relevant characteristics are used for discriminating among characters (Amin, 2002; AbdelRaouf *et al*., 2008).

In any identification system, the features extraction is an important phase whereas the classifier's performance depends heavily on this stage (Abdullah *et al*., 2008; Sawant and Baji, 2016).

In this stage, a set of invariant features (Hu's moments, Zernike moments and Fourier descriptors) for both isolated and connected characters under investigation are extracted and compared with corresponding ones in the database to identify the character under investigation.

Finally, the extracted features are used in classification and recognition stage to classify the character. Characters can be classified according to their computed features to different classes. Firstly, 7 Hu's moment were used, they are:

$$\mu_{pq} = \sum_{x=0}^{N-1}\sum_{y=0}^{M-1}(x-x_c)^p (y-y_c)^q f(x,y) \quad (1)$$

Where:

$$x_c = \frac{m_{10}}{m_{00}} \text{ and } y_c = \frac{m_{01}}{m_{00}} \quad (2)$$

However, certainlize moments are only translation invarient. In order to accure invariance to scale we require normalized central moments which defined as:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\alpha}} \quad (3)$$

Where:

$$\alpha = \frac{p+q}{2}+1 \quad \text{for } p+q \geq 2 \quad (4)$$

$$\phi_1 = \eta_{20}+\eta_{02} \quad (5)$$

$$\phi_2 = (\eta_{20}+\eta_{02})^2 + 4\eta_{11}^2 \quad (6)$$

$$\phi_3 = (\eta_{30}+\eta_{12})^2 + (3\eta_{21}-\eta_{03})^2 \quad (7)$$

$$\phi_4 = (\eta_{30}-\eta_{12})^2 + (\eta_{21}-\eta_{03})^2 \quad (8)$$

$$\phi_5 = (\eta_{30}-3\eta_{12})(\eta_{30}+\eta_{12})[(\eta_{30}+\mu_{12})^2-3(\eta_{21}+\eta_{03})^2]+ (3\eta_{21}-\eta_{03})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2] \quad (9)$$

$$\phi_6 = (\eta_{20}-\eta_{02})[(\eta_{30}+\eta_{12})^2-(\eta_{21}+\mu_{03})^2]+ 4\eta_{11}(\eta_{30}+\eta_{12})(\eta_{21}+\eta_{03}) \quad (10)$$

$$\phi_7 = (3\eta_{21}-\eta_{30})(\eta_{30}+\eta_{12})[(\eta_{30}+\mu_{12})^2-3(\eta_{21}+\eta_{03})^2]+ (3\eta_{21}-\eta_{30})(\eta_{21}+\eta_{03})[3(\eta_{30}+\eta_{12})^2-(\eta_{21}+\eta_{03})^2] \quad (11)$$

But as shown in experimental results, the 7 Hu's moment don't give a good accuracy of identification. While, Fourier descriptors give a good accuracy, they describe the boundary of the character's shape, this is more useful in identification the Arabic characters.

The Fourier descriptors are generated by the Fourier transform, representing the character's boundary as a set of coordinates as:

$$s(k) = [x(k), y(k)], \; k = 0, 1, 2, ..., K-1 \quad (12)$$

$$s(k) = x(k)+iy(k) \quad (13)$$

Fourier descriptors are get from the DFT:

$$a(u) = \sum_{k=0}^{k-1} s(k) e^{-\frac{j2\pi uk}{K}} \quad u = 0, 1, 2, ..., K-1 \quad (14)$$

The experimental results shows that the Fourier descriptors introduce a better performance than moments (Hu''s or Zernike).

RESULTS AND DISCUSSION

Experimental results and comparisons: The implementation of our system is as follows: firstly, the document is scanned and pre-processed for noise removal and tilt correction and redundant space is removed. Secondly, the text segmented to lines then the each line is segmented to words then each word will be segmented to

Table 2: Results of the experiments

Test data				Results							
				-----Lines-----		-----Words-----		------Letters------			
Font	Size	Spacing	Bold							No noise (%)	Noise 10 (%)
Normal	18	1.0	N	6	6	53	77	201	209	75.50	70.00
Normal	18	1.5	B	6	6	53	53	201	180	94.75	91.00
Naskh	18	1.5	N	6	6	53	55	201	169	90.00	84.00
Koufi	20	1.0	N	6	6	53	53	201	114	84.25	80.00
Q-Nask	18	1.0	N	3	3	25	31	114	163	75.50	70.25
Q-Nask	18	1.0	B	4	4	37	35	141	128	90.25	87.20
Weight of each part	-	-	-	-	-	-	30%	35%	35%	100.0	100.0

Fig. 4: The used data set

characters. Thirdly, for each segmented character the features from different descriptors are extracted and used in recognition phase.

With generate a data set that was used in this project. This data set contains 6 different font type, size and style, with the ration for each part of experimental, 30% of accuracy for estimating lines accuracy for words and letters 35% of total accuracy.

Figuer 4 shows different types of fonts for data set. Whereas in the first state, normal font style "Times New Roman" was used with size 18, spacing between line one and no bold. The second one likes the previous but with style bold and 1.5 spacing between lines. The third state was Naskh font with 1.5 spacing and size 18, Koufi font was used on the fourth state with 20 pixel size, 1 spacing and without bolding. The last two sets were like Quran's font but one of them bold and the other is not. In pre-processing process, skew and noise are unavoidably introduced into the image when a document is scanned due to different factors, Fig. 5 illustrates a state of skew detection and correction. The segmentation phase begins across the follows steps:

- Segment the document to lines
- Segment line into words
- Segment the word into characters

Figure 6 shows an example of text segmentation into lines and Fig. 7 shows an example of word segmentation. Table 2 shows the detail of results each of lines, words and letters column has two value the first value indicate the actual number of lines, words or letters and the second ones shows the estimate result after the segmentation process.

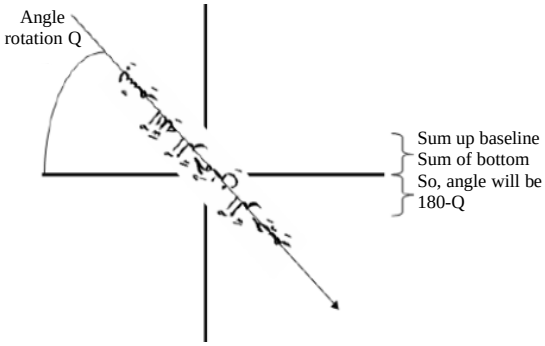

Fig. 5: A state of skew detection and correction

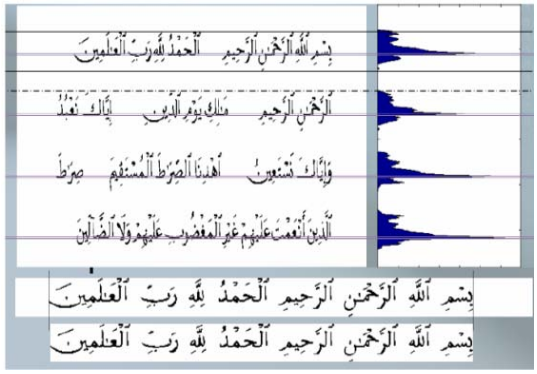

Fig. 6: Segmentation a text into lines

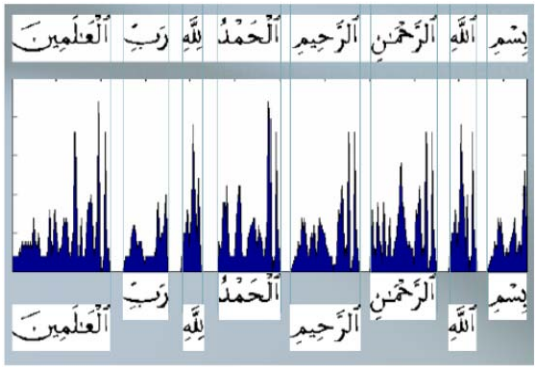

Fig. 7: Words segmentation

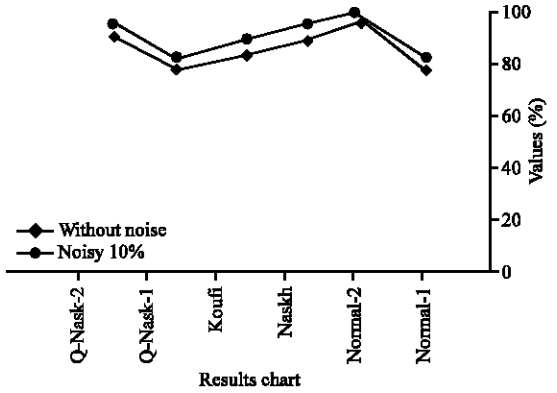

Fig. 8: Experimental result chart

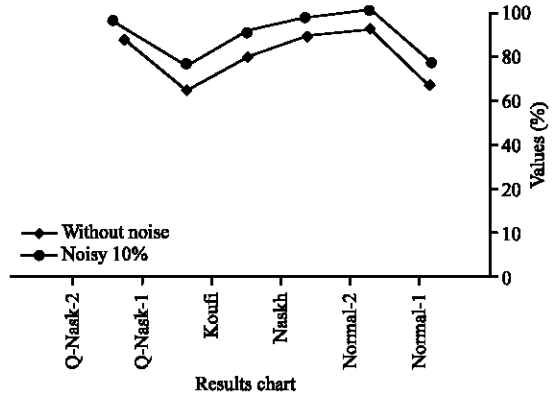

Fig. 9: Results chart of a paper text

Table 3: Accuracy with using the primary part only

Descriptors	----------No noise (%)----------		----Noise (%)----
7-Hu's moments	Whole character	Primary only	Whole character
12-Hu's moments	37	68	20
Fourier	90	96	91
Neural net with Zernike moments	92	94	91

Sometimes the estimated value was greater than the actual one that means the segmentation have more segmented part, for example, if we have letter Sin "seen", it will be segment into three parts. Another case if we have value less than the actual one that means, some letter may be segmented with each other line these letter lam and alef "la". In this case, we cannot segment these two letters lonely but we can consider as one and recognize them as this.

Figure 8 shows a chart of the results and Fig. 9 also shows the experiment on one completed paper of text. With isolated characters (Table 3) shows the accuracy of different descriptors that were used for feature extraction. While with the connected characters, a classifier that was used is the neural network and the recognition rate was 78%. As shown in Table 3, more accuracy was get if the primary part is used instead of the whole character. Fourier descriptors introduce a better performance than moments.

CONCLUSION

This study indicates the problems that are relevant to the printed Arabic characters script and much of the important efforts were made in Arabic character segmentation and identification. Furthermore, it stills an open area for many researches in the future due to the problem of segmentation. Arabic characters are very sensitive to both noise and rotation. Solving these two problems will definitely contribute tremendously in increasing the recognition rate in the Arabic character systems and hence contribute in the automation process for different application. In this study, segmentation technique was used that depends on vertical and horizontal project of the text. Since, this technique is the fastest and easiest way for segmentation. Also, different descriptors were experimented to classify the characters. The experimental results shows that the Fourier descriptors introduce a better performance than moments (Hu's or Zernike).

ACKNOWLEDGEMENTS

Researcher thanks and offers the gratitude to the Information Technology College for its support to the Postgraduates. Also, researcher presents the thankfulness to them classmate for them encourage along the research.

REFERENCES

AbdelRaouf, A., C.A. Higgins and M. Khalil, 2008. A database for Arabic printed character recognition. Proceedings of the 5th International Conference on Image Analysis and Recognition (ICIAR'08), June 25-27, 2008, Springer, Povoa De Varzim, Portugal, pp: 567-578.

Abdullah, S., A. Al-Nassiri and R.A. Salam, 2008. Off-line Arabic handwritten word segmentation using rotational invariant segments features. Intl. Arab J. Inf. Technol., 5: 200-208.

Al-Hamad, H.A., 2013. Neural-based segmentation technique for Arabic handwriting scripts. Proceedings of the of 21st International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision, June 24-27, 2013, European Union Agency for Fundamental Rights (FRA), Plzen, Czech Republic, ISBN: 978-80-86943-75-6, pp: 9-14.

Al-Shatnawi, A., F. Al-Zawaideh, S. Al-Salaimeh and K. Omar, 2011. Offline Arabic text recognition system: An overview. World Comput. Sci. Inform. Technol. J., 1: 184-192.

Aljarrah, I., O. Al-Khaleel, K. Mhaidat, M.A. Alrefai and A. Alzu'bi *et al*., 2012. Automated system for Arabic optical character recognition with lookup dictionary. J. Emerging Technol. Web Intell., 4: 362-370.

Amin, A., 2002. Structural description to recognition Arabic characters using decision tree learning techniques. Proceedings of the Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), August 6-9, 2002, Springer, Windsor, Ontario, Canada, pp: 152-158.

Aouadi, N. and A.K. Echi, 2016. Word extraction and recognition in Arabic handwritten text. Intl. J. Comput. Inf. Sci., 12: 17-23.

Ashkan, M.Y., D.S. Guru and P. Punitha, 2006. Skew estimation in Persian documents: A novel approach. Proceedings of the 2006 International Conference on Computer Graphics, Imaging and Visualisation, July 26-28, 2006, IEEE, Sydney, New South Wales, Australia, ISBN:0-7695-2606-3, pp: 64-70.

Chan, J., C. Ziftci and D. Forsyth, 2006. Searching off-line Arabic documents. Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Vol. 2, June 17-22, 2006, IEEE, New York, USA., ISBN:0-7695-2597-0, pp: 1455-1462.

Hamami, L. and D. Berkani, 2002. Recognition system for printed multi-font and multi-size Arabic characters. Arabian J. Sci. Eng., 27: 57-72.

Hamid, A. and R. Haraty, 2001. A neuro-heuristic approach for segmenting handwritten Arabic text. Proceedings of the 2001 IEEE International Conference on Computer Systems and Applications, June 25-29, 2001, IEEE, Beirut, Lebanon, ISBN:0-7695-1165-1, pp: 110-113.

Haraty, R.A. and C. Ghaddar, 2004. Arabic text recognition. Intl. Arab J. Inf. Technol., 1: 156-163.

Lawgali, A., 2015. A survey on Arabic character recognition. Intl. J. Signal Process. Image Pattern Recognit., 8: 401-426.

Leila, C., K. Maamar and C. Salim, 2012. Combining neural networks for arabic handwriting recognition. Intl. Arab J. Inf. Technol., 9: 588-595.

Sahlol, A.T., C.Y. Suen, M.R. Elbasyouni and A. Sallam, 2014. A proposed OCR algorithm for the recognition of handwritten Arabic characters. J. Pattern Recognit. Intell. Syst., 2: 90-104.

Sarfraz, M., S.N. Nawaz and A. Al-Khuraidly, 2003. Offline Arabic text recognition system. Proceedings of the 2003 International Conference on Geometric Modeling and Graphics, July 16-18, 2003, IEEE, London, UK., ISBN:0-7695-1985-7, pp: 30-35.

Sawant, S. and S. Baji, 2016. Handwritten character and word recognition using their geometrical features through neural networks. Intl. J. Appl. Innovation Eng. Manage., 5: 77-85.

Shaikh, N.A., G.A. Mallah and Z.A. Shaikh, 2009. Character segmentation of Sindhi, an Arabic style scripting language, using height profile vector. Aust. J. Basic Appl. Sci., 3: 4160-4169.

Zeki, A.M., M.S. Zakaria and C.Y. Liong, 2011. Segmentation of Arabic characters: A comprehensive survey. Intl. J. Technol. Diffus., 2: 48-82.