MSA UNIVERSITY ENGINEERING JOURNAL

Fast Technique to Recognize Closed Characters in Arabic Handwriting

Ramadan Gad [1], Shaimaa Kandeel [1, a], Mohamed Al-Adawy [2], Nawal El-Fishawy [1], and Mokhtar A. A. Mohamed [1]

1 Comp. Sci. & Eng. Dept., Faculty of Electronic Engineering, Menoufia University, Menof 32952, Egypt.
2 Department of Electronics and Communications, Faculty of Engineering, Helwan University, Cairo 11795, Egypt.
E-mail: [a,*]Shimahanikandeel@gmail.com.

Abstract

Considering the distinctive features and characteristics of Arabic scripts, recognizing handwritten Arabic characters is still widely open as a major experimental task, which is not easy to overcome. This study attempts to develop a fast and effective technique for recognizing this difficult type of written script by using an iterative algorithm to extract closed Arabic characters. Some Arabic characters have circles that may be blurred, incomplete, unformed, or distorted. In addition to ligatures with different shapes, the technique is based on the development of a proper Circular Hough Transform. This study used IFN/ENIT, which is a standard database for extracting closed Arabic characters as designed in classes. The first class is closed blurred circles (ق, م , ع ,و). The second class is unformed circles (ق, ع , ة). The third one is incomplete circles (ه,ة, ط , ظ, ص, ض). Finally, the fourth class is distorted circles (ط, ظ). This methodology achieved an outstanding accuracy of approximately 96.67%, a very interesting work for the total four classes related to the 11 closed Arabic characters (out of the 28 characters of the Arabic alphabet)..

Keywords: Arabic closed characters, Circular Hough Transform, Handwritten recognition.

ENGINEERING JOURNAL Volume 2 Issue 2
Received Date December 2022
Accepted Date February 2023
Published Date March 2023
Online at https://msaeng.journals.ekb.eg/article_291928.html
DOI: 10.21608/msaeng.2023.291928

1. Introduction

Arabic character recognition has attracted considerable research efforts over the past three decades. The main motivation behind such interest is that handwritten styles of Arabic characters are found in several similar languages, such as Urdu, Pushto, and Persian.

Character recognition systems can be employed in many real-life applications. These systems can contribute to the advancement of the recognition automation process. This recognition process improves the human–computer collaboration in many applications related to the Arabic language. Another important application is bank check readings. An offline handwritten recognition system is very important for signature verification and recognition of forms filled by the user. Such recognition systems can help different users convert handwritten contents to computerized text using mobiles, touch screens, and/or digitizers.

However, automatic processing of handwritten Arabic characters is still a wide-open field for research. Arabic handwritten recognition has not been fully tackled and handled [1]. Many researchers focused on English characters, but the Arabic language needs more effort and attention because it is difficult to identify and read.

This study mainly aims to employ Circular Hough Transform (CHT) to recognize the Arabic closed characters, such as (غ ـغغ ع ـعع ة ف ق و م ه هه ص ض ط ظ ـه ـهـ). The problem with these closed characters is that they are not in the correct form during human drawing. Some are blurred or open, incomplete, distorted, or unformed closed characters. Another problem is that these letters appear in different forms according to their location within any word and type of drawing [2]. For example, an Arabic character called "Haa" can be drawn as "هـ" at the beginning of any word. Its shape in the middle will be like "ـهـ." Its shape in the end will be "ـه" or "ه" depending on the word type. The cursive nature of the Arabic characters increases the difficulty of achieving the required accuracy during recognition [3] ,[4], [5].

This study contributes by using CHT for the first time with Arabic characters in the IFN/ENIT database. In addition, the study helps in recognizing closed and circular handwritten Arabic characters. The results have proven very high and acceptable outcomes. Most studies only used Linear Hough Transform (LHT) in skew and slant characters in the Arabic handwriting. Moreover, they used the Hidden Markov Model (HMM), projection profile with histogram, and morphological algorithms to obtain acceptable outcomes. These closed and handwritten Arabic characters can be found frequently in many words (e.g., Muhammed "محمد," Ahmed "أحمد," Allah "أستغفر الله," and "الله,").

2. Structure

This paper is further arranged as follows. Section 3 presents the related works of Arabic handwriting recognition. Section 4 discusses the proposed handwriting system. Then, Section 5 explains the experimentation and the results of the system. Finally, Section 6 discusses the conclusion and future work.

3. Related work

This section is devoted to shedding light on numerous articles that introduced many methods to improve the efficiency of recognizing Arabic characters. Such methods include LHT [6], [7], [8], [9], [10] , HMM [11], [12], [13], [14], histogram with Gabor filter [15], histogram with projection profile [16], [17], Otsu's model [9], [17], [18], heuristic rules differentiating [19], discrete cosine transform [20], parallel thinning algorithm [21], artificial immune system [22], and/or dynamic Bayesian network [23]. In addition, morphological algorithms [18] were applied to estimate document skew angles using the IFN/ENIT database. The following paragraphs will describe briefly the sample of the literature closest to our proposal.

Boukharouba et al. in [8] applied LHT with a simple labeling algorithm and directional histogram. His research aims to solve the skew angles, skew pages, and baseline detection by utilizing the IFN/ENIT database. The results of this article with IFN/ENIT are unsatisfactory at all for many reasons. They focused only on lower edges and ignored the upper edges. However, they found uncontrolled problems with lower edges. Their method needs a large scale of storage and expensive computations as they used a small sample set of databases. The results, with isolated characters, such as (ر ز ن و), are unsatisfactory, and limitations with short Arabic writing also exist.

Reference [9] presented the skew angles detection and correction for Mushaf Al-Quran image pages based on LHT, histogram projection, and gradient orientation method. Skew detection and correction in Mushaf Al-Quran are known to be different and more difficult than handwritten Arabic characters. The reason is the presence of "diacritics" (Tashkil) and its connected writing style. This research had many limitations. The algorithm, within this research, cannot detect skew angle correction with the range above 20° rotation. In addition, the algorithm cannot deal with dotting, ligatures, diacritical, marks, multiple graphemes, and cursive letters that are found in Mushaf Al-Quran image pages. The outcome results show that the algorithm works perfectly and quickly with low-resolution images through six stages. In the case of high-resolution images, the results are not accurate, and the process becomes time-consuming. To sum up, this work can be considered a preprocessing work when dealing with the recognition of Mushaf Al-Quran image pages. Thus, more research studies are needed.

Bafjaish et al. in [10] applied the methods of histogram with projection profile, LHT, and nearest neighbor method for detecting and correcting the skew angle for different document scripts of Mushaf al-Quran. They used three basic stages: detect lines, skew angle detection, and finally skew rotation. This system achieves a high speed while using LHT but consumes a lot of memory space. They have problems of poor accuracy because of using the nearest neighbor algorithm.

Reference [12] concentrated on detecting skew and slant characters of a small database set of Arabic numbers of 87 images in the IFN/ENIT database. They utilized HMM and Hidden Markov Model Toolkit without priori segmentation. They mentioned that the system is very competitive because it handles overlapping and ligature problems in the IFN/ENIT database.

Rabi et al. in [14] focused on baseline estimation and skew detection with slant correction for cursive handwritten characters and words. They employed embedded training based on HMM. They used the IFN/ENIT benchmark database, where they achieved 87.93% accuracy. They estimated the baseline where the character was divided into upper and lower zones. The feature extraction was used to divide the image into frames based on pixel densities. They use the probabilities in HMM, where complex mathematical methods are being applied.

Reference [16] presented algorithms that solved the problems of ligature, overlapping, and open characters. In addition, they divided the method of segmentation into open- and semi-open loop characters. Their algorithms were applied on two databases: IFN/ENIT and IAM. Their work cannot distinguish Arabic characters (ن ى).

Reference [17] used the unoccupied spaces between the characters to recognize and identify the words. They used the triangle model to extract the features and used the list of point selection for both histograms, vertical and horizontal. They also used the IFN/ENIT dataset but did not mention the recognition rate. Otsu's Model was used for the threshold. The threshold was used as the input for the histogram.

Essa et al. in [18] utilized the morphological Arabic ligature segmentation algorithm. The essential phase of the system was the Enhanced Technique for Arabic character recognition. They applied their algorithms on the IFN/ENIT database and Handwritten Arabic characters database. They encountered problems of connected and unconstrained characters, and one of the solutions is using more Arabic ligature classes. In addition, they faced segmentation problems, such as over segmentation, under segmentation, broken and touching characters, overlapping, and ligature. They solved the aforementioned problems by using the algorithm of Parts of Arabic Words (PAWs) segmentation. PAWs will reduce some cases of over segmentation, under segmentation, and touching characters.

4. Proposed Character Recognition System

Our proposed recognition system is divided into four major stages, as shown in Fig.1. They are Scanning Image, Preprocessing, CHT, and Voting stages. In the scanning stage, our system picks handwritten images individually from the database (IFN/ENIT) location. In the preprocessing stage, each image will be resized, its edges will be detected, and the image will finally be filtered. In the third stage, CHT is employed with voting for varying radius ranges to detect closed characters of the Arabic language. In general, Hough Transform can be used for transferring any point in the X-Y plane to parameter space.

The main contribution of this study is by developing and presenting a new methodology for recognizing closed Arabic characters. Offline Arabic character recognition is the process of converting scanned image documents (numerals, letters, and symbols) into computerized format. The following sections will clarify methods, algorithms, and each description, including the stage of our proposed system.

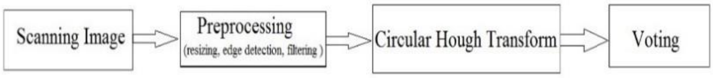

Fig. 1 The overall proposed recognition system

4.1. Stage one: Scanning stage

The scanning images are Arabic words for Tunisian towns and villages from the IFN/ENIT database, as shown in Fig.2.

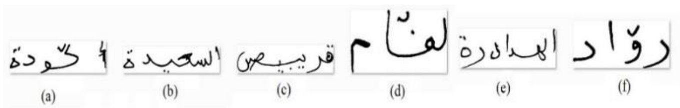

Fig. 2 IFN/ENIT sample Scanning Image

4.2. Stage two: Preprocessing stage

The preprocessing stage is responsible for converting the image into the desired format via three operations; image resizing, edge detection, and filtering. The Canny Edge detection algorithm is applied in this stage, which runs in five separate steps as follows: smoothing step, finding gradients step, Non-maximum suppression step, double thresholding step, and edge step.

The smoothing step aims to blur the image to remove noise by using Gaussian filtering. This step produces and calculates the image gradients and direction of edges. These gradient magnitudes can be calculated as a Euclidean distance, which can be measured by applying Eq. (1) (i.e., the law of Pythagoras). This equation can be simplified as in Eq. (2) by applying the Manhattan distance measuring.

$$|G| = \sqrt{Gx^2 + Gy^2}, \quad (1)$$
$$|G| = |Gx| + |Gy|, \quad (2)$$

where Gx and Gy are the gradients in the x- and y-directions, respectively.

Then, Eq. (3) must be applied to calculate the direction (i.e., slope) of the edge.

$$\theta = \arctan\left(\left|\frac{|Gy|}{|Gx|}\right), \quad (3)$$

where the magnitude *G* and the slope *θ* of the gradient are calculated.

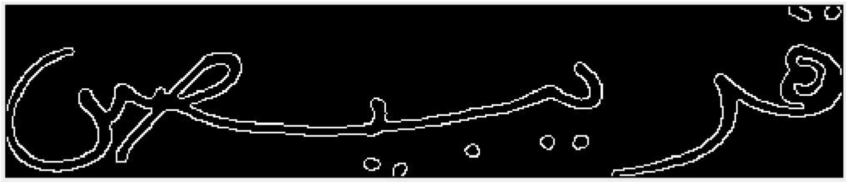

Fig. 3 Preprocessing output image

The finding gradients step is the methodology of skimming large-magnitude gradients to mark the required edges of the image. Non-maximum suppression is the step of marking only local maxima as edges. Double thresholding will be applied to determine potential edges by the mechanism of thresholding. Finally, in Edge Tracking in the Hysteresis step, final edges are determined by suppressing all unconnected edges to a very certain strong edge, as shown in Fig. 3.

In addition, morphological erosion methodology is applied in the preprocessing stage to remove islands and small objects. As a result, only substantive objects will remain. In morphological erosion, the value of output pixels is the minimum value of all the pixels in the neighborhood. In a binary image, a pixel is set to zero if any of the neighboring pixels have a value of zero.

The last methodology of the preprocessing stage is using the connected components method. Pixels are connected if their edges touch, and two adjoining pixels are a part of the same object if both are on and connected along the horizontal or vertical direction.

4.3.Stage three: CHT stage

The CHT has long been recognized as a robust technique for circle detection. Eq. (4) and Eq. (5) show the parametric equations for any circle. In such parametric equations, three parameters are used to represent a circle; a, b, and r. where “a” and “b” are the center of the circle in the X and Y directions, respectively. “r” is the radius of the circle.

$$x = a + r \cos(\theta), \quad (4)$$
$$y = b + r \sin(\theta), \quad (5)$$

Therefore, with the CHT, we expect to find the triplets (X, Y, R) that highly appear in circles in the image. The main aim of using CHT is to detect the close or semi-closed Arabic characters and numbers from the image words. The accumulator of CHT will contain numbers corresponding to the numerals of circles for each type of closed characters, such as (غ ـغغ ع ـعع ة ف ق و م ه هه صـ صـض ط ظ ـه ـهـ). Arabic closed characters can be divided into four categories according to their circle shape. Some of them have an approximate distorted circle shape, such as (ط , ظ , ص , ض). Another group has a blurred circle shape, such as (ق , ف , ـه , و , م , ـغغ , ـعع). Another group has an incomplete circle, such as (ه , ة , ط , ظ ,ض , ص). The last group can be written in an unformed circle shape (i.e., all closed Arabic characters), as shown in Fig. 4.

Fig. 4 (a)blurred circle, (b)uncompleted circle, (c)distorted circle, (d) unformed circle.

4.4.Stage Four: Voting stage

The voting stage depends on the accumulator results of CHT. We apply the radius parameter with ranges from 2 to 30 pixels within images of the IFN/ENIT database. This voting stage is the process of studying the recognition results to find the best radius range through trial and error. This best radius range will help recognize most of the closed Arabic characters in IFN/ENIT images. In addition, these ranges help to determine the wide of the circles in the closed Arabic characters.

5. Experimental Results

The proposed system, with its four stages, is applied to the images of the IFN/ENIT-Tunisian villages and towns. An array-based accumulator is used as an essential step within CHT. In addition, the voting process will revolve around picking out the candidate pixels that can form a pattern of a full circle with a fixed radius. Many iterations are performed to pick out the best and finest votes of prospective circles. The number of iterations is 13, and the consumed time/iteration is 11 seconds. Radius is ranged from minimum value = 2 pixels to maximum value = 30 pixels. The system is completely built using MATLAB using a laptop with the following specification: Windows 10 (64-bit), i5 with 2.2 GHZ microprocessor (5200U), and 4.00 GB memory RAM.

Fig. 5 shows an example of accumulator array outputs for the Arabic word (السعيدة) with different radii r = 13 and r = 5. Fig. 6 shows an example of detected circles in the Arabic word (الشرايع). In Figure 6, three circles are detected, and two of them are detected in error for unclosed characters (س , ع). In addition, Fig. 7 shows the detected circles in the Arabic word (بئر الطيب). In Figure 7, the closed character (ط) is detected correctly, and five circles in error are detected with the Arabic characters (ى , ب , ل).

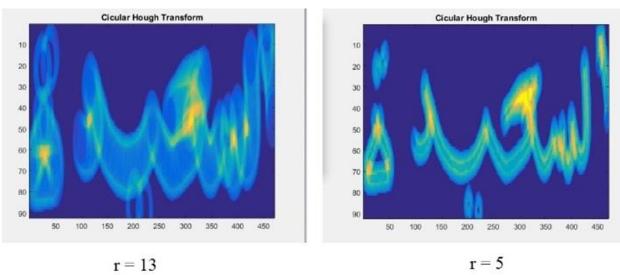

Fig. 5 Output CHT Accumulator with different radius.

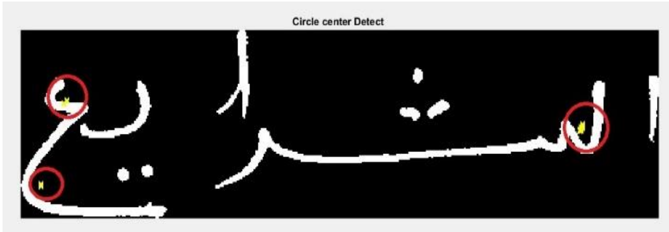

Fig. 6 Example (1) Detect the closed characters.

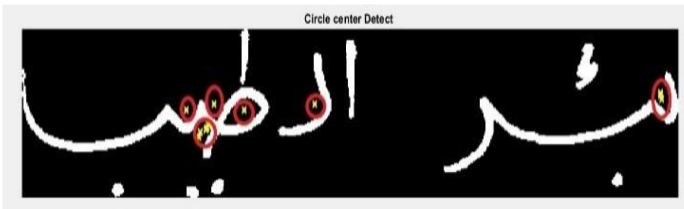

Fig. 7 Example (2) Detect the closed characters.

To avoid an error in circle detection, we apply many runs with different ranges of radius to find the best results of applying our system within handwritten-based images of the IFN/ENIT database. According to the results in Table 1, a radius range of 2–7 pixels is the best range because it achieves correct detection for 551 images out of the 570 images in the IFN/ENIT database. As the radius range increases, the correct detection is reduced. For example, a range of 8–12 pixels causes the correct detection of 437 images. When ranges 23–27 and 28–30 are applied, images of correct detection become 400 and 406, respectively. The results become worst after the final range (28–30).

Table 1. Performance of the system

Radius range	**Successful recognition**	**Error (%)**
2-7	551	3.33
8-12	437	23.3
13-17	432	24.2
18-22	431	24.4
23-27	400	29.9
28-30	406	28.8

Fig. 8 shows an example of the recognition of different closed Arabic characters in sample 1 of the IFN/ENIT database. Sample 1 includes the closed characters (, و , ط , عع , ف ق , ه). Our system can detect the closed characters correctly with a recognition performance rate of 96.67%. Fig.9 shows the recognition of sample 2 for the remaining closed characters with the same performance rate. Sample 2 includes the closed characters , ض , ة , ص , م , ظ (ع).

Fig. 8 Sample 1 of the recognition result

fig. 9 Sample 2 of the recognition result

Table 2 shows a comparative study between the results of different related works in Arabic handwritten recognition within the IFNENIT database. Our proposed system is included in such a comparative study, which achieves a considerable performance rate (96.67%) in the detection of closed Arabic and handwritten characters.

Table 2. Comparative results.

Paper	Algorithm	Problem	Database	Accuracy (%)
(Abdelhak et al, 8)	Linear Hough Transform	Skew angles, Skew	IFN/ENIT	95
(El Moubtahij et al, 12)	Hidden Markov Model with Viterbi algorithm	Skew and Slant characters	IFN/ENIT	78.95
(Mouhcine Rabi et al, 14)	Hidden Markov Model	Baseline and skew detection	IFN/ENIT	87.93
(Zerdoumi et al, 16)	Projection profile and Histogram	Ligatures and open characters	IFN/ENIT and IAM	98
(A.Tahir et al, 17)	Otsu's model and Histogram projection	Unoccupied space between characters	IFN/ENIT	-
(Nada Essa et al, 18)	Morphological algorithm	Ligatures and segmentation	IFN/ENIT	94.99
Proposed	Circular Hough Transform	Closed Characters	IFN/ENIT	96.67

6. Conclusion

The CHT was successfully initiated for the first time with Arabic characters in the IFN/ENIT database. This study has led us to explore the specific properties of the CHT in recognizing closed and circular handwritten Arabic characters. Most of the past literature has only addressed LHT in skew and slant characters in Arabic handwriting. Notably, this study has achieved a high accuracy of 96.67% for all four classes related to the 11 closed Arabic characters (out of the 28 characters in the Arabic alphabet). By using MATLAB–IFN/ENIT based on the results of recognition terms and using an algorithm, the expected results give clarity to the formed character and the appearance of true circles. Our proposal is to further explore the CHT method for offline closed Arabic handwriting characters in more future studies.

Finally, this study can be considered an important step forward toward the use of CHT in recognizing Arabic handwriting characters.

The authors, whose names are listed on this paper, certify that they have no conflicts of interest to disclose.

References

[1] Mezghani A, Kanoun S, Khemakhem M, Abed H, "A database for arabic handwritten text image recognition and writer identification", *In: International conference on frontiers in handwriting recognition,* (ICFHR), 2012, p. 339–402.2

[2] Ali A, Suresha M., "Survey on Segmentation and Recognition of Handwritten Arabic Published online", 6 June 2020, p.1.

[3] Zaidan A, Zaidan B, Jalab H, Alanazi H, Alnaqeib R., "Offline Arabic Handwriting Recognition Using Artificial Neural Network",*(JCSE) ,journal of computer science and engineering,* P47-67, 2010.

[4] Ali A, Suresha M., "Survey on Segmentation and Recognition of Handwritten Arabic Published online", 2020, P1-2.

[5] Zaidan A, Zaidan B, Jalab H, Alanazi H, Alnaqeib R., "Offline Arabic Handwriting Recognition Using Artificial Neural Network", ",*(JCSE) ,journal of computer science and engineering*, , P47-67, 2010.

[6] Edgard Chammas, Chafic Mokbel and Laurence Likforman-Sulem, "Arabic handwritten document preprocessing and recognition", *International Conference on Document Analysis and Recognition* (ICDAR), 2015, 10.1109.

[7] Bhavesh Kumar Shukla, Gautam Kumar, and Ashish Kumar, "An approach for Skew Detection using Hough Transform", *International Journal of Computer Applications* (0975 – 8887) Volume 136 – No.9, February, 2016,

[8] Abdelhak Boukharouba, "A new algorithm for skew correction and baseline detection based on the randomized Hough Transform*", King Saud University Journal of King Saud University, Science direct, production and hosting by Elsevier* B.V, P1319-1578, 2017.

[9] Bafjaish S, Azmi M , Al-Mhiqani M, et al., "Skew detection and correction of Mushaf Al-Quran script using Hough transform", *International Journal of Advanced Computer Science and Applications*, P402-409, 2018.

[10] Salem Saleh Bafjaish, Mohd Sanusi Azmi, Mohammed Nasser Al-Mhiqani, Ahmed Abdalla Sheikh, "Skew correction for Mushaf Al-Quran: a review", *Indonesian Journal of Electrical Engineering and Computer Science* Vol. 17, No. 1, P516-523, January 2020.

[11] Aid Ahmed Radaideh, and Mohd Shafry Mohd Rahim, "Existing techniques in Arabic characters recognition (ACR) ", *Journal of Informatics and Mathematical Sciences*, VOL.8, No.5, 2016.

[12] El Moubtahij Hichama, Halli Akramb, Satori Khalid, "Using features of local densities, statistics and HMM toolkit (HTK) for offline Arabic handwriting text recognition", *Journal of Electrical systems and information Technology science direct*, JESIT, Vol.4, P387-396, 2017.

[13] Mouhcine RABI, Mustapha AMROUCH, and Zouhir MAHANI, "Recognition of cursive Arabic Handwritten text using Embedded Training based on HMMs", *Journal of Electrical Systems and Information Technology* (JESIT), 2017.

[14] Mouhcine Rabi, Mustapha Amrouch and Zouhir Mahani, "Recognition of Cursive Arabic Handwritten Text Using Embedded Training Based on Hidden Markov Models", *International journal of pattern recognition and Artificial Intelligence*,Vol.32, No.01., 2018.

[15] Ansar Hani, Mohamed Elleuch, and Monji Kherallah "Arabic Handwritten Script Recognition System Based on HOG and Gabor Features", *International Arab Conference on Information Technology* (ACIT), 2016.

[16] Zerdoumi Saber, Aznul Qalid Md Sabri, Amirrudin Kamsin, Saqib Hakak, "Efficient Approach to Segment Ligatures and Open Characters in Offline Arabic text", *International Journal of Computing, Communication and Instrumentation Engineering* (IJCCIE) Vol. 4, Issue 1, Malaysia, 2017.

[17] A. Tahir, M. S. Azmi, N. Ahmad, N. A. Arbain, A. R. Radzid, "Document Feature Extraction Based on Unoccupied Space Using Triangle Model: A Preliminary Work", *Journal of Telecommunication, Electronic and Computer Engineering*, JTEC, Malaysia, Vol.10, P2-7, 2018.

[18] Nada Essa , Eman El-Daydamony, Ahmed Atwan Mohamed, "Enhanced technique for Arabic handwriting recognition using deep belief network and a morphological algorithm for solving ligature segmentation", *ETRI Journal*, 2018.

[19] Mazen Bahashwan, Syed Abu-Bakar, and Usman Sheikh, "Efficient Segmentation of Arabic Handwritten Characters Using Structural Features", *The International Arab Journal of Information Technology*, Vol. 14, No. 6, November, 2017.

[20] A. Lawgali, M. Angelova and A. Bouridane, "A Framework for Arabic Handwritten Recognition Based on Segmentation", *International Journal of Hybrid Information Technology*, Vol.7, No.5, 2014.

[21] Atallah M. AL-Shatnawi, and Khairuddin Omar, " The Thinning Problem in Arabic Text Recognition - A Comprehensive Review", *International Journal of Computer Applications* (0975 – 8887) Volume 103 – No.3, October, 2014.

[22] Chaouki Boufenar, Mohamed Batouche, and Marc Schoenaue, "An artificial immune system for offline isolated handwritten arabic character recognition", *Springer Journal, Evolving Systems*, 9(1), 2018.

[23] Khaoula Jayech, Mohamed Ali Mahjoub, and Najoua Ben Amara, "Arabic Handwritten Word Recognition based on Dynamic Bayesian Network", *The International Arab Journal of Information Technology*, VOL. 13, NO. 3, 2016.