# المجلة العربية للعلوم والهندسة

# THE ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING

VOLUME 16, NUMBER 4

INCLUDES PAPERS ON THEME:
COMPUTERS AND ARABIZATION

OCTOBER 1991

*Published by* KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS, DHAHRAN 31261, SAUDI ARABIA

# ARABIC CHARACTER RECOGNITION: MANY APPROACHES AND ONE DECADE

**Kamal M. Jambi***

*Department of Computer Science*
*King Abdul Aziz University*
*Jeddah, Saudi Arabia*

الخلاصـــة :

يتم في هذا البحث دراسة العديد من الطرق المختلفة بغرض تمييز الأحرف العربية على مدى العشر سنوات الماضية ، وقد تضـمّـن دراسة العمليات التي تتم في مرحلة ماقبل المعالجة ومرحلة المعالجة نفسها ومرحلة مابعد المعالجة .

إن دراسة عمليات مرحلة ماقبل المعالجة تتضمن بعضاً من هذه العمليات ؛ ومنها إزالة التشويش القائم في خيال الحرف العربي وتحويل الخيال من متعدد القيم إلى ثنائي القيمة وكذلك التنحيف بالإضافة إلى فصل الأحرف المتشابكة . أما في مرحلة المعالجة فيتم فيها دراسة تجزئة الكلمة إلى الأحرف المكونة لها بالإضافة إلى اختيار الصفات المميزة لكل حرف كما يتم دراسة الوسائل المتّبعة من قبل الباحثين والخاصة بتمييز الأحرف العربية .

*Address for correspondence:
P.O. Box No. 17182
Jeddah 21484
Saudi Arabia

October 1991

## ABSTRACT

This paper gives a survey on Arabic character recognition through the last ten years. It covers different approaches taken by researchers regarding this topic. This includes different operations taken as preprocessing, the process of recognition itself, and postprocessing if applicable. The preprocessing includes, but is not limited to, the following operations: noise elimination, digitization, thinning, and separation of overlapped characters. The section on the processing stage covers the procedures for segmenting the whole word into characters, feature selection, and the different approaches adopted for recognition, which are discussed in detail.








# ARABIC CHARACTER RECOGNITION: MANY APPROACHES AND ONE DECADE

## 1. INTRODUCTION

The process of character recognition is an old topic, and is still a current issue. This has been true for Arabic character recognition since the first known system [1], called Interactive Recognition of Arabic Character (IRAC), was implemented in the late seventies. Nowadays, researchers strive to achieve more speed for recognition with higher accuracy. This becomes available by means of sophisticated and much more powerful PC's.

The ultimate goal for character recognition in general is to develop a communication interface between the computer and its users. This implies the direct storing of handwriting of the users into computer memory without going through a keyboard.

There are several properties of Arabic characters which give handwriting style uniqueness and cause difficulties for recognition. Being cursive (*i.e.* characters of a single word within any text are connected) implies that the boundaries of these characters can overlap very easily. It is interesting to know also that some groups of characters have the same main body with a slight change, for example, the number of dots and their relative positions (Figure 1). Moreover, Arabic characters have different shapes depending on their location within a word (*i.e.* isolated, or occurring at the beginning, middle, or end of the word).

Harmon [2] is considered to be the first one to deal with Latin cursive scripts. Although he gave 128

| name | single | last | middle | first | name | single | last | middle | first |
|---|---|---|---|---|---|---|---|---|---|
| alif | ا | ـا | ـا | ا | dhad | ض | ـض | ـضـ | ضـ |
| ba | ب | ـب | ـبـ | بـ | tta | ط | ـط | ـطـ | طـ |
| ta | ت | ـت | ـتـ | تـ | zha | ظ | ـظ | ـظـ | ظـ |
| tha | ث | ـث | ـثـ | ثـ | ain | ع | ـع | ـعـ | عـ |
| jeem | ج | ـج | ـجـ | جـ | ghain | غ | ـغ | ـغـ | غـ |
| hha | ح | ـح | ـحـ | حـ | fa | ف | ـف | ـفـ | فـ |
| kha | خ | ـخ | ـخـ | خـ | qaf | ق | ـق | ـقـ | قـ |
| dal | د | ـد | ـدـ | دـ | kaf | ك | ـك | ـكـ | كـ |
| thal | ذ | ـذ | ـذـ | ذـ | lam | ل | ـل | ـلـ | لـ |
| ra | ر | ـر | ـرـ | رـ | meem | م | ـم | ـمـ | مـ |
| za | ز | ـز | ـزـ | زـ | noon | ن | ـن | ـنـ | نـ |
| seen | س | ـس | ـسـ | سـ | ha | ه | ـه | ـهـ | هـ |
| sheen | ش | ـش | ـشـ | شـ | waw | و | ـو | ـوـ | وـ |
| sad | ص | ـص | ـصـ | صـ | ya | ي | ـي | ـيـ | يـ |

*Figure 1. Shapes of Arabic Characters.*

references on Latin character recognition, his survey still could not be comprehensive as he stated. This was due to a rapid and prolific evolution which automatic recognition of print and script had experienced. Nagy [3], a pioneer in Chinese character recognition, has given a history of Chinese character recognition over a period of twenty-five years. In this survey, he claimed that technology made Chinese recognition commercially practical, but the keys to a successful system were good feature extraction and classification algorithms. He also stated that for estimating a defined error rate, the size of sample space should be ten times the inverse of that error rate (*i.e.* 10 000 characters should be tested in order to proclaim an overall error of 0.1 percent). Accordingly, this present survey covers only the resources available to the author.

The process of character recognition goes through different stages, and accordingly this paper is organized in the same manner. It goes through different methods used in order to capture images. This is followed by a discussion on different operations used in the preprocessing stage. At the end a detailed investigation on the process of recognition will be given.

## 2. OBTAINING IMAGES

Obtaining the images of texts, words, or characters is considered to be the first step in the process of character recognition. On-line systems use graphic tablets which give those systems more information related to the direction of plotted lines and their order similar to those used in [1, 4]. However, this technique is somehow not practical since it requires the physical existence of writers for obtaining the images. Yet, some applications such as signature verification require this kind of device. The other kind of systems are called off-line where direction of lines and their order are not known. One of the devices used in this kind of systems are scanners as those used in [5, 6]. Scanners are used to capture the image of a full word. These devices have the ability of automatic digitization and the binary image is obtained easily. Moreover, the noise associated with using these devices is considered to be very low. Video cameras are also used in off-line systems such as [7–9]. In this case, images are obtained with noise and multi-grey-level which implies the necessity of careful determination of a threshold value in order to obtain the binary image. Another device called a telecopier was used in [10].

## 3. PREPROCESSING

The main point behind preprocessing is to remove the writer's personal style and leave normalized writer independent script. The authors of [11] claimed that preprocessing for recognition purposes has been given much less attention than it deserves. Therefore, it is strange enough to know that no pretreatment is used in [12]. The preprocessing operations include, but are not limited to, the following.

### 3.1. Noise Elimination

The objective of this operation is to remove the noise associated with obtaining images. Usually, this is done by scanning the image with a low-pass filter where scanning is carried out in a raster fashion. The concept of removing the noise is implemented by distributing or smoothing out the noise into the surrounding pixels. For instance, in [10] this is done by recalculating values of the binary image by using two equations. The first one is to restore missing points and the second to delete noise and loosely connected edge points. The effect of the second equation is called smoothing in [9].

### 3.2. Digitization

The multi-grey-level images obtained by using a video camera need to be converted to a binary image. This is done through digitization, where a histogram is made and the threshold value is determined. The histogram itself is obtained by counting the number of pixels having a specific value, which is one of the multi-grey-level values, and the threshold is assigned the value of the valley between the two peaks in the histogram. Therefore, pixels with greater value than the threshold become 1's, otherwise they are converted to 0's. However, recent research [13] claimed that images could be converted into binary form more accurately if we took into consideration the grey values of surrounding pixels. This implies having two histograms and the threshold value as a vector with two elements.

### 3.3. Thinning

This operation allows the representation of images in a simpler form without losing any information. Smith [14] investigated a wide range of thinning algorithms. However, new techniques are still being produced, seeking less time, fewer iterations, and better skeletons. The thinning algorithm of [15] is claimed to preserve important topology of Arabic characters such as slashes and dots (*Nuqtah*). This is done by reducing *Nuqtah* to a single point instead of a short vertical line (to be distinguished from a slash or stroke). However, each iteration of this algorithm requires scanning the image in four directions which might slow down the thinning process.







### 3.4. Separation of Overlapped Characters

Althogh overlapping of adjacent Arabic characters occurs naturally, yet it is ignored in most of the research work. This happens because recognition is implemented either on printed text or on handwritten with no overlapping, which is unnatural. However, [6] and [10] consider this issue explicitly by inserting at least one blank column between those characters. On the other hand [7] dealt with this issue implicitly by tracing a continuous stroke, erasing it from the image, then considering other strokes. It is claimed also in [16] that separation between different overlapping subwords can be done by contour follower.

Other operations during preprocessing include finding the frame of the word which is used in [6] to reduce the scanning area by eliminating the process of scanning the white area around the frame of the isolated word. Preprocessing includes removing the dots from characters with the same basic shape in order to have better correlation measurement between character shapes as implemented in [17]. Determining the base line [6, 7] is also considered to be a preprocessing operation. The baseline is the line where interconnection of the characters of a word takes place. Therefore finding the baseline gives an indication where the location of the starting point of each character might be located. Preprocessing operations also include removing any stray boundary strokes [1] and using some routine to remove accidental breaks in strokes due to removing the pencil.

## 4. PROCESSING

This section deals with the core of the topic. It covers different approaches taken for recognition. However, the scope of each work is different and varies from recognizing a printed isolated character to a handwritten cursive word.

### 4.1. Segmentation

This operation identifies the character decomposition of a word. The importance of this operation comes from the fact that wrong segmentation produces misrecognition or rejection. Segmentation by histogram (Figure 2) is a strong candidate since it has

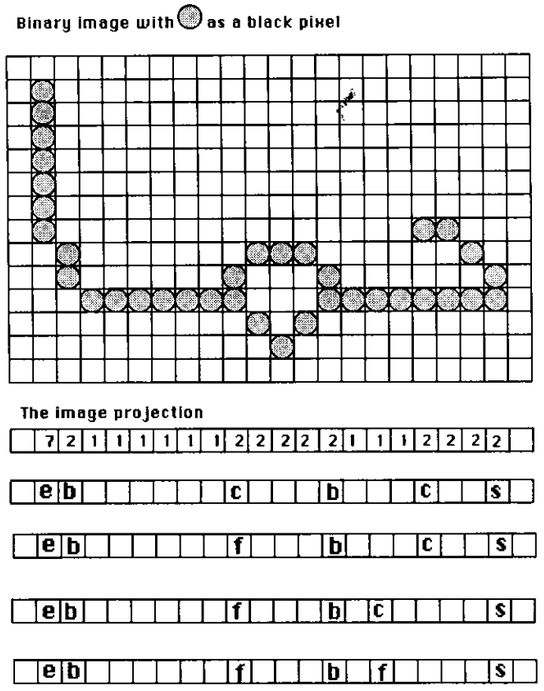



*Figure 2. Steps Taken in Function 'Histo' to Identify the Start and End of Each Character [6].*

been used in different papers [4–6, 12, 18, 19]. The reason behind selecting segmentation through histogram is its simplicity. Nevertheless, this approach depends heavily on a predefined threshold value related to the character width. Moreover, skewed images will not work effectively with this approach. The concept of a histogram is used in [12] with three stages to allocate the text, to separate words in the line, and to separate characters of a word. However, in [18] the results of segmentation are primitives rather than characters, in [4] and [19] few rules need to be satisfied, and in [12] the morphology of Arabic characters is taken into consideration. This last approach worked very well on printed text where morphology can be detected very easily.

Segmentation is also achieved by tracing the outer contour of a given word and calculating the distance between the extreme points of intersection of the contour with a vertical line [16]. In [7] the segmentation into primitives is done by tracing continuous strokes. Another method is used in [10]: finding what is called an actual connection column (ACC).

No segmentation is a different approach taken by authors of [20] for recognizing typewritten Arabic words. They treated the word instead of the character as a unit of processing. Since no segmentation is needed, dynamic programming techniques are used to perform connected character recognition. Those techniques are taken from the theory of optimal control by using non-linear transformation [20]. However, much time is needed for implementing dynamic programming.

### 4.2. Feature Extraction and Recognition

It is known that features represent the smallest set that can be used for discrimination purposes and unique identification for each character. Among the earliest work is [21]. Nouh and others proposed a standard character set by having *Nuqtah* (dot) as a unit, *Alif* with six units as a standard length, and an imaginary circle that all characters can be written within. In this work, 13 features or radicals are selected by inspection which represent parts of characters. The recognition is based on a decision tree and a strong correlation measurement. Another early work [22] tried to find the best features to represent Arabic characters. It started by the binary image to select 13 features which were reduced later by eliminating the extra features. In [17] different algorithms dealing with feature extraction are discussed. The emphasis was done toward the mathematical features where three different measurements of correlation were tested. The recognition is based on the results of correlation measurement calculated using geometrical features of character. Probability correlation is used also by [5]. In this work, probability correlation is proceeded by a *K*-means clustering and a decision-tree is used for recognition.

#### *4.2.1. Hierarchical Classification*

Going over the techniques used for the process of recognition, it seems that the hierarchical classification, (similar to a decision-tree (Figures 3, 4)), is the dominant one. The following are the approaches that

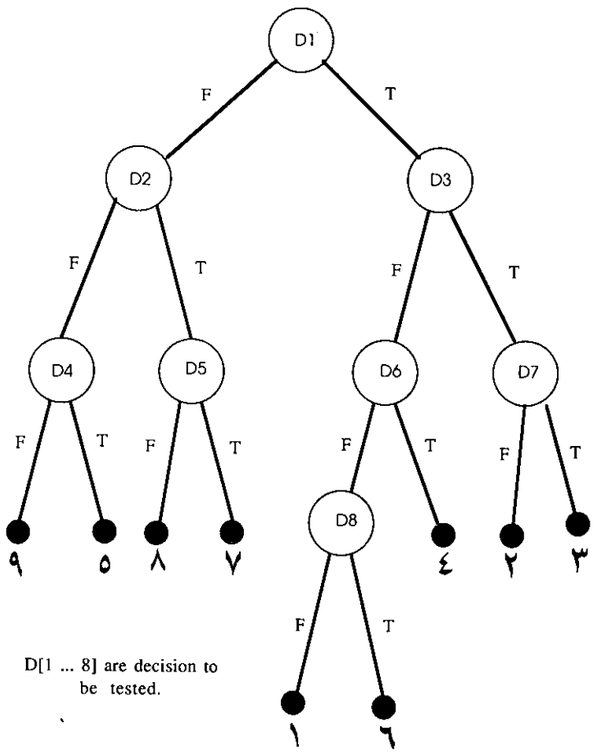



*Figure 3. Tree Structure for Recognition of Hindi Numerals Used in [24].*










*Figure 4. Tree Classifier for Arabic Characters Used in [26].*

adopted this type of classification. For instance, the implementation of IRAC [1] was done by using a decision tree classifier. The decision was made depending on the number of strokes, characteristics of the dots (their number and position), and the presence of zigzag. There was also a formula to be evaluated which depended on number of segments of the main stroke and the direction of each segment.

The recognition in [19] is done for multifont Arabic text, including character recognition and word recognition. In this work, no thinning is performed and segmentation is by vertical histogram, with certain rules that have to be satisfied to insure the correctness of segmentation. This is followed by scanning the image through a $2\times2$ window for contour detection; Freeman code is used for character description [19]. At the end a decision tree is used for recognition. The decision tree is also used in [4], where eight primitives are considered and used to determine the path through the tree. The later work had the ability to learn, since if a new character was considered, a new path associated with this character would be added and the character name would be used as a label for the leaf of that path. This ability to learn allows adding new characters, which implies an enhancement in the adequacy of recognition. It should be noted that [4] covered the recognition of isolated characters as well as cursive words, and different techniques were presented.

A direct hierarchical classification is used in [23] for recognizing segmented handwritten characters. In this case, four supergroups are identified depending on the position of the character with respect to a word. These positions are the beginning, within, and end of the word. The fourth corresponds to the case when the character is isolated. Moreover, each group is classified into four subgroups depending on the number of strokes (one, two, three, or four). The classification of characters within each subgroup is done by means of some features such as having a maximum or a minimum on either horizontal or vertical direction, ratio between length and width, the type of secondary stroke, *etc.*

Hidden Markov Model (HMM) is used by [24] for the recognition of Hindi numerals. The features in this technique are the number of crossings in the two

orthogonal directions when sliding along the centroidal axis of the numeral. These features are not sensitive to variation in style and size of numeral. These features are desirable and make the process of recognition independent of the writer's writing style. A tree classifier on geometrical features is used for recognition (Figure 3). The same technique (HMM) is also used by [25] although the technique in this case is used for recognizing a spoken Arabic word.

The authors of [26] took a hierarchical organization of characters to reduce recognition time (Figure 4). In their work, the main stroke is represented by a chain code and the features used are related to the secondary stroke. These include the number of dots, their relative position, number of secondary strokes, and their slopes with respect to the main stroke. The classification uses a tree structure with three levels where the first level is obtained by *K*-nearest neighbor. The exact match is obtained by finding a minimum distance.

In order to simplify the process of recognition, a new and clear shape of Arabic letters is proposed in [27]. This proposition enforced that each letter has at most three shapes and no overriding is allowed. Moreover, some modification of letters' shape is made to obtain different code. Each character is represented by the number of horizontal and vertical crossing and in both cases only the variation is counted. The decision tree is used for recognition by means of two steps. The first finds out the cluster, and the second identifies the character itself.

The decision in [28] is based on finding the optimum path in the decision tree by selecting an array with a maximum information. The fuzzy approach adopted in this work is based on a given fuzzy recognition model. In this case, each character is written inside a square frame with sixteen radial axes drawn from its center. The number of intersections made by any character with the radial axes will be considered as the elements of its feature vector. These elements are no-crossing, single-crossing, double-crossing, and triple-crossing.

*4.2.2. Non-Hierarchical Classification*

The following approaches do not use a decision tree as a classifier. For instance, table lookup is adopted in [6], and this approach begins by decomposition of the word into separate characters. This is followed by determining the frame window of the character and dividing it into small windows where features can be searched for (Figure 5). The features

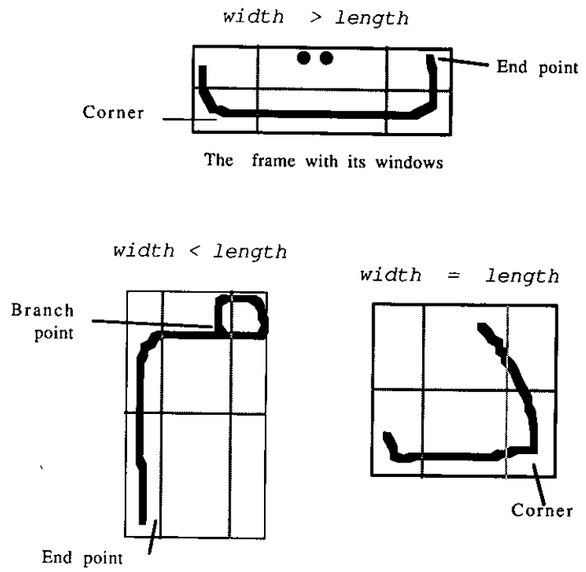



*Figure 5. The Three Different Classes of Characters [6].*

include end point, intersection point, corner, and the relationship between the length and width of window frame. Characters are identified by an association between feature points and their locations within the window frame [6]. At the end, recognition is achieved by finding a match between an unknown character and entries in a lookup table.

Table lookup is used in [9] for recognizing an isolated handwritten Arabic character. In this approach, the character is placed in a frame which is divided into six rectangles and a contour tracing algorithm is used for coding the contour as a set of directional vectors using Freeman code. Unfortunately, it turns out that contour information is not sufficient to determine Arabic characters, therefore the authors added the information related to number of dots and their position. In the case of no match, the system will add the feature vector to the table and consider that character as a new entry.

The work of [7] was presented in the most comprehensive and detailed way. So, after classifying strokes using their geometrical and topological properties, those strokes are combined in several steps into a string of characters that represent the recognized word. The classification of strokes is done according to their calculated values such as start and end points, length, frame, and the connection points. The features are extracted according to the classifi-

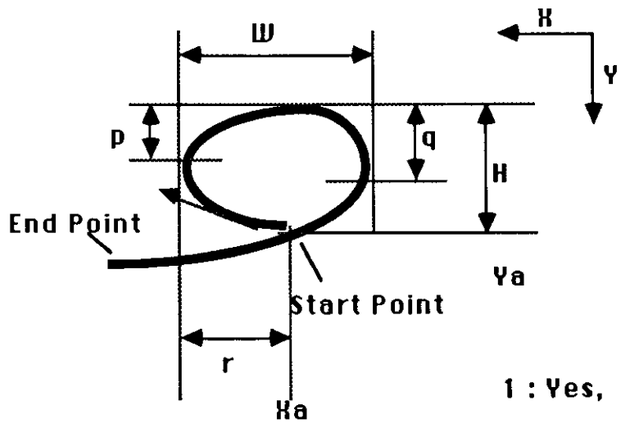






cation of strokes by answering questions dedicated to each class of strokes (Figure 6). The recognition itself is done by triggering a rule corresponding to each character when a match is observed. Therefore, the handwritten word is recognized when all its characters are identified.

Other techniques used include obtaining a set of Fourier descriptors from the coordinate sequences of the outer contour which is used for recognition in [16]. Also, in [8] each character is assigned a logic function where characters are preclassified into four groups depending on the existence of certain pixels




| Strokes | f1 | f2 | f3 | f4 | f5 | f6 | f7 | code |
|---|---|---|---|---|---|---|---|---|
| | 1 | 0 | 1 | 1 | * | * | 0 | 1 |
| | 1 | 0 | * | 0 | * | 0 | 0 | 2 |
| | 1 | 0 | 0 | * | * | 1 | 0 | 3 |
| | 1 | 0 | 0 | 1 | * | 0 | 0 | 3 |
| | 1 | 1 | 0 | * | 0 | * | 1 | 4 |
| | 1 | 1 | * | * | 0 | * | 0 | 5 |
| | 0 | * | * | * | 0 | * | 0 | 6 |
| | * | 1 | 0 | * | 1 | * | * | 7 |

f1 : (Ya - Ylpmin) > (Ylpmax-Ya) ?

f2 : Is the end point a line end ?

f3 : $0 < {}^{a} <= 90$ ?

f4 : H <= 0.75 W ?

f5 : (Ya - Yb) >= abs( Xa-Xb) ?

f6 : (p<0.3W) and (q<0.3w) and (r>0.4w)?

f7 : $-40 < {}^{b} < 40$ ?

*Figure 6. Features, Codes, and Identification Vector of Group 3 Strokes [7].*

in a specified location of the image matrix. The approach in [10] is done by selecting 20 features to construct a 24 bit vector, then comparing with entries of a table where exact match is checked first. The best match is selected if the exact match failed.

## 5. POSTPROCESSING

This aspect is not touched by most researches. This might be the case because the associated works claimed a satisfactorily high precentage recognition of characters. In fact, postprocessing is used frequently in recognizing words rather than characters which might also give some explanation for not implementing postprocessing techniques.

One common technique of postprocessing is done by investigating the probabilities of each character with respect to its location within the word as well as its order (*i.e.* comes after or before) against the other characters. These probabilities can be obtained by studying the words of a dictionary. Therefore, after finding out the characters of the unknown word, these probabilities should be taken into consideration in order to determine the best candidate word.

However, postprocessing is carried out in [5] where lexicon rules are used to identify the character after observing the primitive sequence. The same authors [18] generated a lexicon matrix to find out the character and primitive component of a word.

## 6. CONCLUSION

The process of Arabic character recognition is still an open field for more research. This survey represents some of the work done in this field. As mentioned before, the scope of the above work varies with respect to complexity from printed isolated characters to handwritten words. This implies that evaluation of the work based only on the recognition rate is not be fair and the reader might be mislead. So, it is left to reader, after tasting the flavor of each technique to refer to each work individually for more details.

As stated before, new research should hunt for more accuracy and less time. In order to attain this goal, researchers introduce a kind of parallelism [29], or use a kind of associative matching [30] to speed up the recognition time. Moreover, the characteristics of Arabic characters should be taken into consideration when a new approach is derived. The rule of "quick and dirty" can also be used as a preliminary step to identify what might be called a superclass or cluster to narrow the scope of search. As a last word, we would like to see one of these approaches implemented and used at a commercial level.

## REFERENCES


[1] A. Amin, A. Kaced, J. Haton, and R. Mohr, "Handwritten Arabic Character Recognition by The I.R.A.C. System", *International Conference on Pattern Recognition,* 1980, p. 729.

[2] L. D. Harmon, "Automatic Recognition of Print and Script", *Proceedings of the IEEE,* **60** (1972), p. 1165.

[3] G. Nagy, "Chinese Character Recognition: A Twenty-Five-Year Retrospective", *The 9th International Conference on Pattern Recognition,* 1988, p. 163.

[4] A. Amin, "Arabic Handwritten Recognition and Understanding", *Proceedings of Computer Processing of The Arabic Language, Kuwait,* 1985.

[5] H. Y. Abdelazim and M. A. Hashish, "Arabic Reading Machine", *Proceedings of the 10th National Computer Conference, Riyadh,* March 1988, p. 733.

[6] K. Jambi and T .Grace, "A New Topological Structural Approach for the Recognition of an Isolated Arabic Word", *Information Technology in Support of Economic Development Conference, Khartoum, Sudan,* December 9–12, 1990.

[7] H. Almuallim and S. Yamguchi, "A Method of Recognition of Arabic Cursive Handwritten", *IEEE Trans. on Pattern Analysis and Machine Intelligence,* **9(5)** (1987), p. 715.

[8] A. Nouh, A. Ula, and A.Sharaf Eldin, "Boolean Recognition Technique for Typewritten Arabic Character Set", *Proceedings of the First King Saud University Symposium on Computer Arabization, Riyadh,* 1987, p. 563.

[9] Suhail Saadallah and Saad Yacu, "Design of an Arabic Character Reading Machine", *Proceedings of Computer Processing of The Arabic Language, Kuwait,* 1985.

[10] B. Parhami and M. Taraghi, "Automatic Recognition of Printed Farsi Texts", *Pattern Recognition,* **14(16)** (1981), p. 395.

[11] M. Brown and S. Ganapathy, "Preprocessing Techniques for Cursive Script Word Recognition", *Pattern Recognition,* **16(5)** (1983), p. 447.

[12] K. Bouhilali, M. K. Hamrouni, and N. Ellouze, "Method of Segmentation of Arabic Text Image into Characters", *Proceedings of The First Kuwait Computer Conference,* March 1989, p. 442.

[13] A. Abutaleb, "Automatic Thresholding of Gray-Level Pictures Using Two-Dimensional Entropy", *Computer Vision, Graphics, and Image Processing,* **47(1)** (1989), p. 22.

[14] R. Smith, "Computer Processing of Line Images: A Survey", *Pattern Recognition,* **20(1)** (1987), p. 7.

[15] A. Nouh, N. Ula, and A. Sharaf Eldin, "A Proposed Algorithm for the Thinning Binary Arabic Character Pattern", *Proceedings of The First Kuwait Computer Conference,* March 1989, p. 426.

[16] Talaat El-Sheikh and Ramez Guindi, "Computer Recognition of Arabic Cursive Scripts", *Pattern Recognition,* **21(4)** (1988), p. 293.

[17] A. Nouh, N. Ula, and A. Sharaf Eldin, "Algorithms for Feature Extraction: A Case Study for Arabic Character Recognition", *Proceedings of the 10th National Computer Conference, Riyadh,* March 1988, p. 563.

[18] H. Y. Abdelazim and M. A. Hashish, "Interactive Font Learning for Arabic OCR", *Proceedings of The First Kuwait Computer Conference,* March 1989, p. 464.

[19] A. Amin and Mari, "Machine Recognition and Correction of Printed Arabic Text", *IEEE System, Man, and Cybernetics,* **19(5)** (1989), p. 1300.

[20] M. Khemakhem and M. Fehri, "Arabic Typewritten Character Recognition Using Dynamic Comparison", *Proceedings of The First Kuwait Computer Conference,* March 1989, p. 448.

[21] A. Nouh, A. Sultan, and R. Tulba, "An Approach for Arabic Character Recognition", *J. Eng. Sci., Univ. Riyadh,* **6(2)** (1980), p. 185.

[22] A. Nouh, A. Sultan, and R. Tulba, "On Feature Extraction and Selection for Arabic Character Recognition", *Arab Gulf Journal of Scientific Research,* **2(1)** (1984), p. 329.

[23] T. S. El-Sheikh and S. G. El-Taweel, "Real-Time Arabic Handwritten Character Recognition", *IEEE 89 3rd Int. Conf. of Image Processing and Its Application,* 1989, p. 212.

[24] H. Y. Abdelazim and M. A. Hashish, "Automatic Recognition of Handwritten Hindi Numerals", *Proceedings of the 11th National Computer Conference, Dhahran,* March 1989, p. 287.

[25] Z. Emam and M. A. Hashish, "Application of Hidden Markov Model to the Recognition of Isolated Arabic Word", *Proceedings of the 10th National Computer Conference, Riyadh,* March 1988, p. 761.

[26] M. S. El-Wakeel and A. A. Shoukry, "On-Line Recognition of Handwritten Arabic Character", *Pattern Recognition,* **22(2)** (1989), p. 97.

[27] S. El-Ramly and M. El-Hamalawy, "A New Font Arabic Characters Simplifies Recognition Procedure", *Proceedings of The First Kuwait Computer Conference,* March 1989, p. 396.

[28] M. Al-Tikriti and S. Al-Ramahi, "A Fuzzy Approach for Some Arabic Handwritten Characters Computer Recognition", *Proceedings of Computer Processing of The Arabic Language, Kuwait,* 1985.

[29] H. F. Li, R. Jayakumar, and M. Youssef, "Parallel Algorithm for Recognizing Handwritten Characters Using Shape Features", *Pattern Recognition,* **22(6)** (1989), p. 641.

[30] E. Kimura, and others, "An Intelligent Character Recognition System with High Accuracy and High Speed by Integrity Image-Type and Logical-Type Information Processing", *9th International Conference on Pattern Recognition,* **1** (1988), p. 38.


**Paper Received 12 May 1990; Revised 17 December 1990; 14 May 1991.**