Pergamon

*Pattern Recognition*, Vol. 31, No. 5, pp. 517–530, 1998
© 1998 Pattern Recognition Society. Published by Elsevier Science Ltd
All rights reserved. Printed in Great Britain
0031–3203/98 $19.00 + .00

**PII: S0031-3203(97)00084-8**

# OFF-LINE ARABIC CHARACTER RECOGNITION: THE STATE OF THE ART

ADNAN AMIN*

School of Computer Science and Engineering, University of New South Wales, Sydney, New South Wales 2052, Australia

(*Received* 13 *December* 1996; *accepted* 22 *July* 1997)

**Abstract**—Machine simulation of human reading has been the subject of intensive research for almost three decades. A large number of research papers and reports have already been published on Latin, Chinese and Japanese characters. However, little work has been conducted on the automatic recognition of Arabic characters because of the complexity of printed and handwritten text, and this problem is still an open research field. The main objective of this paper is to present the state of Arabic character recognition research throughout the last two decades. © 1998 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved

Arabic characters  Off-line recognition  Handwriting recognition  Segmentation
Feature extraction  Neural Network classifiers  Hidden Markov Models
Optical character recognition

## 1. INTRODUCTION

Character recognition systems can contribute tremendously to the advancement of the automation process and can improve the interaction between man and machine in many applications, including office automation, check verification and a large variety of banking, business and data entry applications.

The different approaches covered under the general term character recognition fall into either the on-line or off-line category, each having its own hardware and recognition algorithms.

In on-line character recognition systems, the computer recognizes the symbols as they are drawn.[1–4] The most common writing surface is the digitizing tablet, which operates through a special pen in contact with the surface of the tablet and emits the coordinates of the plotted points at a constant frequency. Breaking contact prompts the transmission of a special character. Thus, recording on the tablet produces strings of coordinates separated by signs indicating when the pen has ceased to touch the tablet surface.

On-line recognition has several interesting characteristics. First, recognition is performed on one-dimensional data rather than two-dimensional images as in the case of off-line recognition. The writing line is represented by a sequence of dots whose location is a function of time. This has several important consequences:

- The writing order is available and can be used by the recognition process.
- The writing line has no width.
- Temporal information, like velocity can also be taken into consideration.
- Additionally, penlifts can be useful in the recognition process.

Among the on-line systems that recognize isolated Arabic characters, several methods can be found in references (5)–(11). Amin[12] introduced three methods for recognizing on-line handwritten Arabic cursive words. The first is a structural method[13] based on segmenting the word into characters. Characters are then recognized using a method similar to that for isolated characters.[15] Word recognition works by constructing all possible words by following every path in the equivalence graph of the lattice. Binary diagrams[14] are also used to discard ineligible combination of letters. The second is syntactical method[15] based on segmentation of words into primitives such as curves and strokes. An automaton transforms the primitives into a list of the characters constituting the word. Finally, the third method uses a global approach:[16,17] each word is identified according to a vector of some pre-determined parameters. Furthermore, to enhance the recognition rate a syntactic and semantic analyzer that verifies the grammatical structure and the meaning of Arabic sentence is used.[18]

Al-Emmami and Usher[19] presented a system for on-line recognition of Handwritten Arabic words. Words are segmented into strokes based on the method proposed by Belaid.[20] In the preliminary learning process, specifications of the strokes of each characters are fed to the system, while in the recognition process, the parameters of each stroke are found

* E-mail: amin@cse.unsw.edu.au.

and special rules are applied to select the collection of strokes that best match the features of one of the stored characters. However, few words were used in the learning and testing process, which make the performance of the system questionable.

Off-line recognition is performed after the writing or printing is completed. Optical Character Recognition, OCR,[21-29] deals with the recognition of optically processed characters rather than magnetically processed ones. In a typical OCR system, input characters are read and digitized by an optical scanner. Each character is then located and segmented and the resulting matrix is fed into a preprocessor for smoothing, noise reduction, and size normalization. Off-line recognition can be considered the most general case: no special device is required for writing and signal interpretation is independent of signal generation, as in human recognition.

Over the past three decades, many different methods have been explored by a large number of scientists to recognize characters. A variety of approaches have been proposed and tested by researchers in different parts of the world, including statistical methods,[30-32] structural and syntactical methods,[33-35] neural networks,[36-38] expert systems[39-41] and machine learning.[42-44]

Many papers have been concerned with the recognition of Latin, Chinese and Japanese characters. However, although almost a third of a billion people worldwide, in several different languages, use Arabic characters for writing, little research progress, in both on-line and off-line, has been achieved towards the automatic recognition of Arabic characters. This is a result of the lack of adequate support in terms of funding, and other utilities such as Arabic text databases, dictionaries, etc. and of course because of the cursive nature of its writing rules.

Although other surveys have dealt with both online and off-line Arabic characters,[45-47] this paper tries to summarize all the work accomplished in the past two decades in only off-line systems in an attempt to pin-point the different areas that need to be tackled. The remainder of this paper is organized as follows: Section 2 reviews some of the basic characteristics of Arabic writing. Section 3 covers different approaches for segmentation and feature extraction, and presents various methods adopted for the recognition. Finally, concluding remarks are given in Section 4.

## 2. GENERAL CHARACTERISTICS OF ARABIC WRITING

Comparison of the various characteristics of Arabic, Latin, Hebrew and Hindi scripts are outlined in Table 1. Arabic is written from right to left. Arabic text (machine printed or handwritten) is cursive in general and Arabic letters are normally connected on the base line. This feature of connectivity will be shown to be important in the segmentation process. Some machine printed and handwritten texts are not cursive, but most Arabic texts are, and thus it is not surprising that the recognition rate of Arabic characters is lower than that of disconnected characters such as printed English.

Table 1. Comparison of various scripts

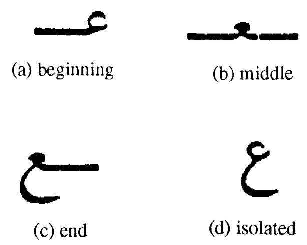
| Characteristics | Arabic | Latin | Hebrew | Hindi |
|---|---|---|---|---|
| Justification | R-to-L | L-to-R | R-to-L | L-to-R |
| Cursive | Yes | No | No | Yes |
| Diacritics | Yes | No | No | Yes |
| Number of vowels | 2 | 5 | 11 | — |
| Letters shapes | 1–4 | 2 | 1 | 1 |
| Number of letters | 28 | 26 | 22 | 40 |
| Complementary characters | 3 | — | — | — |

Arabic writing is similar to English in that it uses letters (which consist of 29 basic letters), numerals, punctuation marks, as well as spaces and special symbols. It differs from English, however, in its representation of vowels since Arabic utilizes various diacritical markings. The presence and absence of vowel diacritics indicates different meanings in what would otherwise be the same word. For example, مدرسة is the Arabic word for both "school" and "teacher". If the word is isolated, diacritics are essential to distinguish between the two possible meanings. If it occurs in a sentence, contextual information inherent in the sentence can be used to infer the appropriate meaning. In this paper, the issue of vowel diacritics is not treated, since it is more common for Arabic writing not to employ these diacritics. Diacritics are only found in old manuscripts or in very confined areas.

The Arabic alphabet is represented numerically by a standard communication interchange code approved by the Arab Standard and Metrology Organization (ASMO). Similar to the American Standard Code for Information Interchange (ASCII), each character in the ASMO code is represented by one byte. An English letter has two possible shapes, capital and small. The ASCII code provides separate representations for both of these shapes, whereas an Arabic letter has only one representation in the ASMO table. This is not to say, however, that the Arabic letter has only one shape. On the contrary, an Arabic letter might have up to four different shapes, depending on its relative position in the text. For instance, the letter (عA'in) has four different shapes: at the beginning of the word (preceded by a space), in the middle of the word (no space around it), at the end of the word (followed by a space), and in isolation (preceded by an unconnected letter and followed by a space). These four possibilities are exemplified in Fig. 1.

Table 2, shows the different shapes of the Arabic characters in the different positions of the word.

In addition, different Arabic characters may have exactly the same shape, and are distinguished from



(a) beginning (b) middle

(c) end (d) isolated

Fig. 1. Different shapes of the Arabic letter 'A' in '.

each other only by the addition of a complementary character. (complementary characters: a portion of a character that is needed to complement an Arabic character). These are normally a dot, a group of dots or a zigzag (hamza). These may appear on, above, or below the base line and are positioned differently, for instance, above, below or within the confines of the character. Figure 2 depicts two sets of characters, the first set having five characters and the other set three characters. Clearly, each set contains characters which differ only by the position and/or the number of dots associated with it. It is worth noting that any erosion or deletion of these complementary characters results in a misrepresentation of the character. Hence, any thinning algorithm needs to efficiently deal with these dots so as not to change the identity of the character.

Arabic writing is cursive and is such that words are separated by spaces. However, a word can be divided into smaller units called subwords (a portion of a word including one or more connected characters). Some Arabic characters are not connectable with the succeeding character. Therefore, if one of these characters exists in a word, it divides that word into two subwords. These characters appear only at the tail of a subword, and the succeeding character forms the head of the next subword. Figure 3 shows three Arabic words with one, two, and three subwords. The first word consists of one subword which has nine letters; the second has two subwords with three and one letter, respectively. The last word contains five subwords, each consisting of only one letter.

Arabic writing can be, in general, classified into typewritten (Naskh), handwritten (Ruq'a) and artistic (or decorative Calligraphy, Kufi, Diwani, Royal, and Thuluth) styles as shown in Fig. 4. Handwritten and decorative styles usually include vertical combinations of characters called ligatures. This feature makes it difficult to determine the boundaries of the characters. Furthermore, characters of the same font have different sizes (i.e. characters may have different widths even though the two characters have the same font and point size). Hence, word segmentation based on a fixed size width cannot be applied to Arabic.

## 3. RECOGNITION OF ARABIC CHARACTERS

There are two strategies which have been applied to printed and handwritten Arabic character recognition. These can be categorized as follows:

(1) Holistic strategies in which the recognition is globally performed on the whole representation of words and where there is no attempt to identify characters individually. These strategies were originally introduced for speech recognition and can fall into two categories:
  (1.1) Methods based on distance measurements using Dynamic Programming.[48,49]
  (1.2) Methods based on a probabilistic framework (Hidden Markov Models).[50-55]

(2) Analytical strategies in which words are not considered as a whole, but as sequences of small size units and the recognition is not directly performed at word level but at an intermediate level dealing with these units, which can be graphemes, segments, pseudo-letters, etc.[47,56,57]

### 3.1. *Word segmentation*

The segmentation phase is a necessary step in recognizing printed Arabic text. Any error in segmenting the basic shape of Arabic characters will produce a different representation of the character component.

Two techniques have been applied for segmenting machine printed and handwritten Arabic words into individual characters: implicit and explicit segmentations.

(1) Implicit segmentation (straight segmentation): in this technique, words are segmented directly into letters. This type of segmentation is usually designed with rules that attempt to identify all the character's segmentation points.

(2) Explicit segmentation: in this case, words are externally segmented into pseudo-letters which are then recognized individually. This approach is usually more expensive due to the increased complexity of finding optimum word hypotheses.

In all printed Arabic characters, the width at a connection point is much less than the width of the beginning character. This property is essential in applying the baseline segmentation technique.[56,57,59] The baseline is a medium line in the Arabic word in which all the connections between the successive characters take place. If a vertical projection of bi-level pixels is performed on the word equation (1)],

$$v(j) = \sum_i w(i,j) \qquad (1)$$

where $w(i,j)$ is either zero or one and $i, j$ index the rows and columns, respectively, the connectivity point will have a sum less than the average value $(AV)$ [equation (2)]

$$AV = (1/Nc) \sum_{j=1}^{Nc} Xj \qquad (2)$$

and where $Nc$ is the number of columns and $Xj$ is the number of black pixels of the $j$th column.

Table 2. The basic alphabets of Arabic characters and their forms at different positions in the word

| | isolated (i) | end (e) | middle (m) | beginning (b) |
|---|---|---|---|---|
| alif | أ | ـا | ـا | ا |
| ba | ب | ـب | ـبـ | بـ |
| ta | ت | ـت | ـتـ | تـ |
| tha | ث | ـث | ـثـ | ثـ |
| jim | ج | ـج | ـجـ | جـ |
| ha | ح | ـح | ـحـ | حـ |
| kha | خ | ـخ | ـخـ | خـ |
| dal | د | ـد | ـد | د |
| dhal | ذ | ـذ | ـذ | ذ |
| ra | ر | ـر | ـر | ز |
| zan | ز | ـز | ـز | ز |
| siin | س | ـس | ـسـ | سـ |
| shiin | ش | ـش | ـشـ | شـ |
| sadd | ص | ـص | ـصـ | صـ |
| dad | ض | ـض | ـضـ | ضـ |
| tahn | ط | ـط | ـطـ | طـ |
| zah | ظ | ـظ | ـظـ | ظـ |
| ayn | ع | ـع | ـعـ | عـ |
| ghayn | غ | ـغ | ـغـ | غـ |
| fa | ف | ـف | ـفـ | فـ |
| qaf | ق | ـق | ـقـ | قـ |
| kaf | ك | ـك | ـکـ | کـ |
| lam | ل | ـل | ـلـ | لـ |
| miim | م | ـم | ـمـ | مـ |
| noon | ن | ـن | ـنـ | نـ |
| ha | ه | ـه | ـهـ | هـ |
| waw | و | ـو | ـو | و |
| ya | ى | ـي | ـيـ | يـ |
| lamalif | لا | ـلا | ـلا | لا |
| tamarbot | ة | ـة | | |

Hence, each part with a sum value much less than $AV$ should be a boundary between different characters. However if the histogram produced from the vertical projection does not follow the condition of equation (3), the character remains unsegmented, as illustrated in Fig. 5.

By examining Arabic characters, it is found that the distance between successive peaks does not exceed one third the width of the Arabic character. That is

$$|d_k| < d_l/3 \tag{3}$$

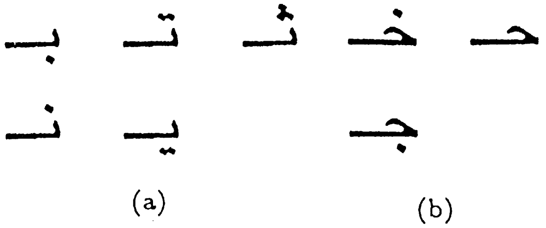


Fig. 2. Arabic characters differing only with regard to the position and number of associated dots.

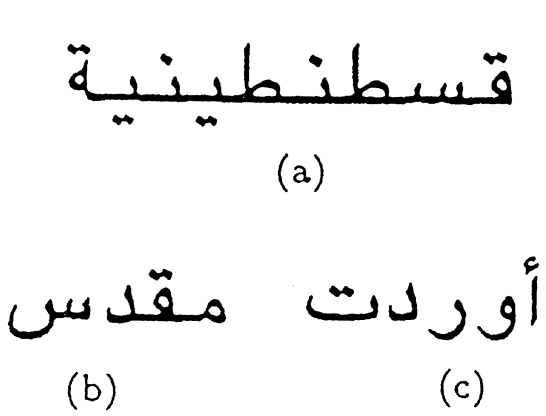


Fig. 3. Arabic words with constituent subwords.

where $d_k$ is the distance between $k$th peak and peak $k+1$, and $d_l$ is the total width of the character.

Moreover, at the end of a word or a subword, equation (4) is also to hold.

$$L_{k+1} > 1.5L_k \qquad (4)$$

where $L_k$ is the $k$th peak in the histogram. This rule is brought to bear because of the inter-connectivity of Arabic characters and their shapes at the end of a word.

This approach depends heavily on a predefined threshold value related to the character width. Moreover, this approach will not work effectively for skewed images.

Almuallim and Yamaguchi[57] proposed a structural recognition technique for Arabic handwritten words. Their system consists of four phases. The first is preprocessing, in which the word is thinned and the midline of the word is detected. Since it is difficult to segment a cursive word into letters, words are segmented into separate strokes and classified as complementary characters, strokes with a loop and strokes without a loop. These strokes are then further classified using their geometrical and topological properties. Finally, the relative positions of the classified strokes are examined, and the strokes are combined in several steps into the string of characters that represents the recognized word. System failures in most cases were due to incorrect segmentation of words.

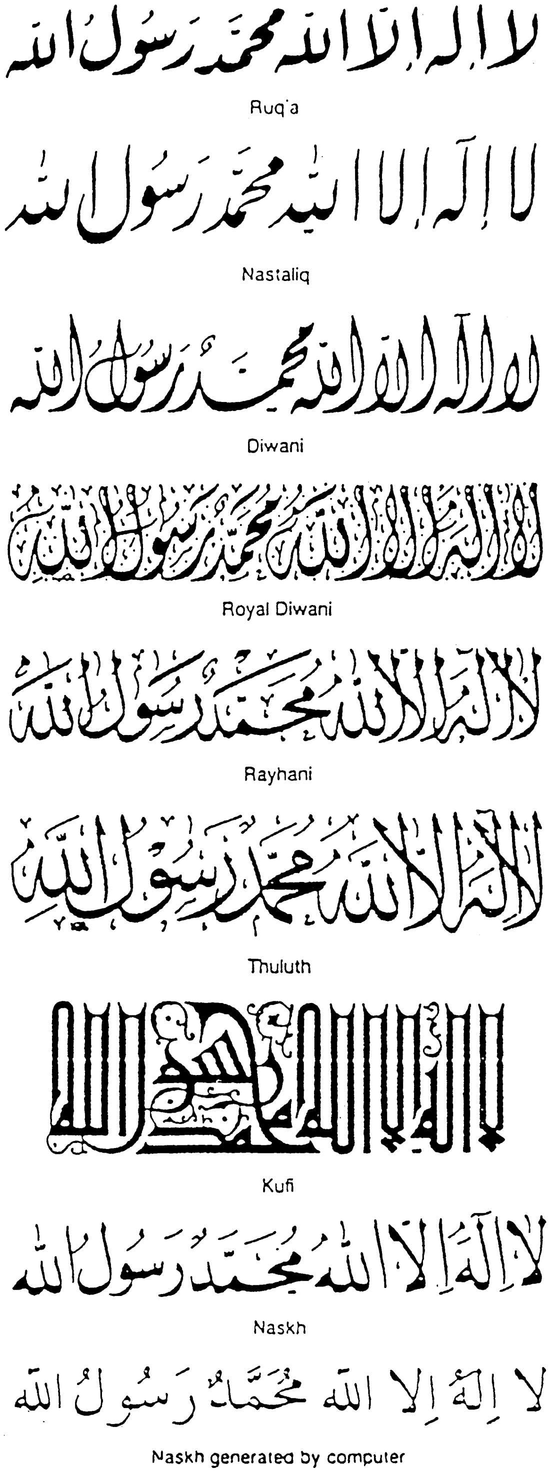


Fig. 4. Different styles and fonts for the writing of Arabic text.

Segmentation is also achieved by tracing the outer contour[60] of a given word and calculating the distance between the extreme points of intersection of the contour with a vertical line. The segmentation is based on a horizontal scan from right to left of the

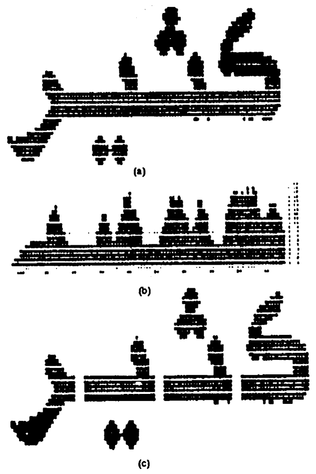


Fig. 5. An example of segmentation of the Arabic word كثير into characters: (a) Arabic word, (b) histogram, (c) word segmented into characters.

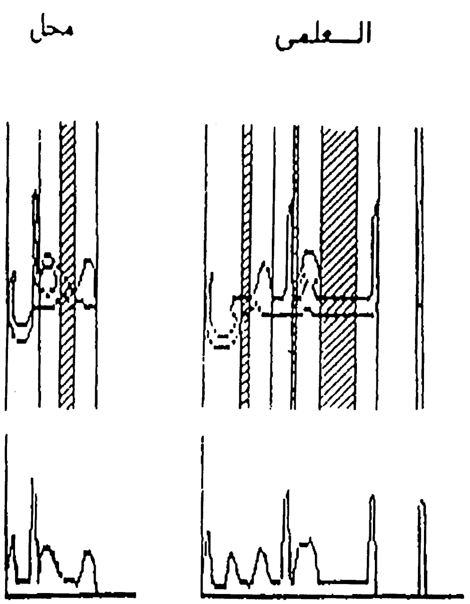


Fig. 6. Segmented Arabic words and the corresponding contour heights.

closed contour using a window of adjustable width *w*. For each position of the window, the average vertical distance $h_{av}$ is calculated across the window. At the boundary between two characters, the following conditions should be met:

(1) $h_{av} < T$. In this case, a silence region is detected, which means that the average vertical distance over the window should be less than a certain preset threshold *T*.
(2) Detected boundaries should lie on the same horizontal line (the base line).
(3) No complementary characters should be located (above or below the base line) at a silence region.

Readjustment of parameters *w* and *T* as well as backtracking may occur if segmentation leads to a rejected character shape. Figure 6 illustrates some examples of this method.

El-Khaly and Sid-Ahmed(61) segment a thinned word into characters by following the average baseline of the word and detecting when the pixels start to go higher or lower than it.

Abdelazim and Hashish(62) use the technique of traversing an energy curve (similar to that used in speech recognition, to discriminate the spoken utterance from the silence background), which shows the number of black pixels in each column of the digitized word, to segment the word into characters. This curve is traversed and a threshold value is used to select significant primitives leaving out silent zones.

Shoukry(63) used a sequential algorithm based on the input-time tracing principle which depends on the connectivity properties of the acquired text in the binary image domain. This algorithm bears some resemblance to an algorithm devised by Wakayama(64) for the skeletonization of binary pictures.

The SARAT system(65) used outer contours to segment an Arabic word into characters. The word is divided into a series of curves by determining the start and end points of the word. Whenever the outer contour changes sign (from a positive to a negative curvature) a character is segmented.

Kurdy and Joukhadar(66) use the upper distance function of the subword, which is the set of the highest points in each column. They assign to each point of the function a token name by comparing the point's height to the height and token name of the point on its right. Using a grammar, they then parse the sequence of tokens of a subword to find the connection points.

Finally, Amin and Al-Sadoun(67, 68) adopted a new technique for segmenting Arabic text. The algorithm can be applied to any font and it permits the overlay of characters. There are two major problems with the traditional segmentation method which depends on the baseline:

(1) Overlapping of adjacent Arabic characters occurs naturally, see Fig. 7a. Hence, no baseline exists. (his phenomenon is common in both typed and handwritten Arabic text.
(2) The connection between two characters is often short. Therefore, placing the segmentation points is a difficult task. In many cases, the potential segmentation points will be placed within a character rather than between characters.

The word in Fig. 7a was segmented utilizing a baseline technique. Figure 7b shows the proper segmentation and the result of the new segmentation method is shown in Fig. 7c.

The new technique can be divided into four major steps. First is the digitization step in which the

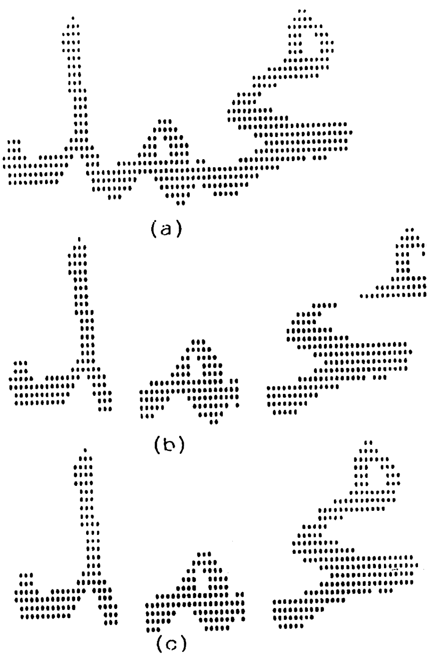


Fig. 7. Example of an Arabic word محمد and different techniques of the segmentation.

original image is transformed into a binary image utilizing a scanner (300 dpi). Second, there is a preprocessing step in which the Arabic word is thinned using a parallel thinning algorithm. Third, the skeleton of the image is traced from right to left using a $3 \times 3$ window and a binary tree is constructed. The Freeman code[(69)] is used to describe the skeleton shape. Finally, the binary tree is segmented into subtrees such that each subtree describes a character in the image.

### 3.2. *Feature extraction and recognition*

It is known that features represent the smallest set that can be used for discrimination purposes and for a unique identification for each character. Features can be classified into two categories:

(1) Local features which are usually *geometric* (e.g. concave/convex parts, type of junctions: intersections/T-junctions/endpoints, etc.).
(2) Global features which are usually *topological* (connectivity, number of connected components, number of holes, etc.) or *statistical* (Fourier transform, invariant moments, etc.).

Nouh *et al.*[(70)] suggested a standard Arabic character set to facilitate computer processing of Arabic characters. In this work, thirteen features, or radicals, which represent parts of characters are selected by inspection. The recognition is based on a decision tree and a strong correlation measurement. The disadvantage of the proposed system is the assumption that the incoming characters are generated according to specified standard rules.

Parhami and Taraghi[(71)] presented a technique for the automatic recognition of machine printed Farsi text (which is similar to Arabic text). The authors first segment the subword into characters by identifying a series of potential connection points on the baseline at which line thickness changes from or to the thickness of the baseline. Although they also have some rules to keep characters at the end of a subword intact, they segment some of the wider characters (e.g. س) into up to three segments. Then they select twenty features based on certain geometric properties of the Farsi symbols to construct a 24 bit vector that is compared with entries of a table where an exact match is checked first. The system is heavily font dependent, and the segmentation process is expected to give incorrect results in some cases.

Table lookup is used for the recognition of isolated handwritten Arabic characters.[(72)] 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 by using a Freeman code. However, this information is not sufficient to determine Arabic characters, therefore extra information related to the number of dots and their position is added. If there is no match, the system will add the feature vector to the table and consider that character as a new entry.

Amin and Masini[(56)] adopted a structural approach for recognizing printed Arabic text. Words and subwords are segmented into characters using the baseline technique. Features such as vertical and horizontal bars are then extracted from the character using horizontal and vertical projections. Four decision trees, chosen according to the position of the character within the word which was computed by the segmentation process, have been used. The structure of four decision trees allows a rapid search for the appropriate character. Furthermore, trees are utilized in distinguishing characters that have the same shape but appear in different positions within a word.

Amin and Mari[(57)] proposed a new technique for multifont Arabic text which includes character and word recognition. A character is divided into many segments by a horizontal scan process (Fig. 8). In this way, segments are connected to form the basic shape of the character. Segments not connected with any other segment are considered to be complementary characters. By using the Freeman code,[(69)] the contour detection process is applied to these segments to trace the basic shape of the character and generate a directional vector through a $2 \times 2$ window. A decision tree is then used for the recognition of the characters. Finally, a Viterbi algorithm[(73)] is used for Arabic word recognition to enhance the recognition rate. The main advantage of this technique is to allow an automatic learning process to be used.

The study reported in refertences (61), (74) and (75) utilizes moment invariant descriptors to recognize the

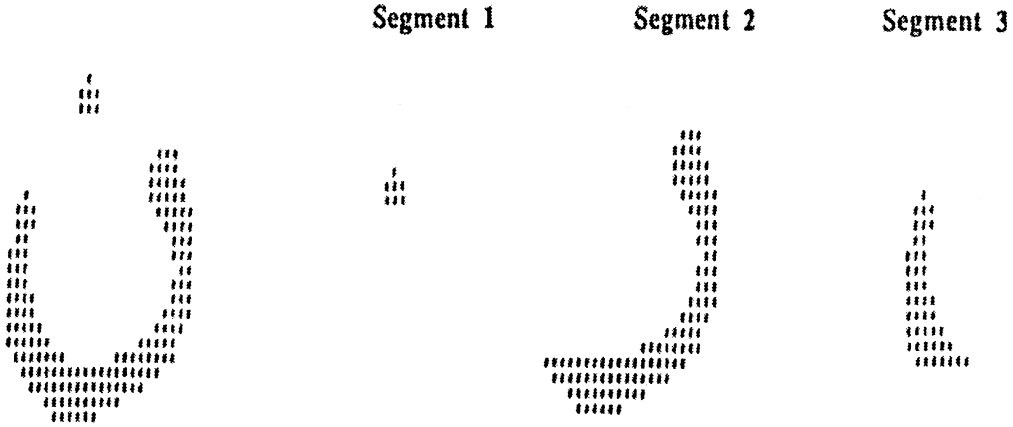


Fig. 8. Major segments of character ن.

characters. Other techniques include a set of Fourier descriptors from the coordinate sequences of the outer contour which is used for the recognition.[60] Also, in reference (76) each character is assigned a logical function where characters are pre-classified into four groups depending on the existence of certain pixels in a specified location of the image.

In reference (45) table lookup is adopted for the recognition of isolated Arabic characters. In this approach, the character is placed in the frame window and divided into small windows to extract some features. These features include end points, intersection points, corners, and the relationship between the length and width of the window frame. Characters are identified by an association between feature points and their locations within the window frame. The recognition is achieved by finding a match between unknown characters and entries in a lookup table.

To enhance the recognition rate of an OCR system, some characteristic morphological properties of the Arabic language can be used. Amin and Al-Fedaghi[58,77] describe a method for spell correction of Arabic words. They correct spelling errors and complete words that have some unrecognized characters using an algorithm that depends on the frequencies of roots and patterns in Arabic.

Haj Hassan[78,79] introduced a syntactic-structural method for recognizing printed Arabic text. Words are segmented into characters using a method similar to that proposed in reference (71). Primitives such as horizontal, vertical, and oblique with positive and negative slopes are then extracted from the character. These primitives are detected in a pre-defined regions inside the characters. Finally, descriptive languages (binary word) are used to describe the characters.

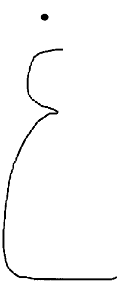
Al-Badr and Haralick[80] proposed a system to recognize machine printed Arabic words without prior segmentation by applying mathematical morphology operations on the whole page to find the locations where shape primitives are present. They then combine those primitives into characters and print out the character identities and their location on the page.

Sano *et al.*[81] introduced a structural approach using fuzzy relations for recognizing handwritten isolated Arabic characters. Each input pattern is divided into sub-patterns (strokes) by feature points; end points, branch points, intersections and maximum curvatures point, etc. The number of sub-patterns varies from one to six depending on the input character. The sub-pattern are then represented in terms of similarity to primitive elements (straight line, circle and diacritical point). The algorithm has been tested on a small number of handwritten samples.

Finally, Bouslama[82] adopted an algorithm based on structural technique and fuzzy logic for recognizing isolated printed Arabic characters. The structural technique is used to extract features from the input character such as number of strokes before and after segmentation, the position of the center of gravity of each sub-segment, the black pixel ratio of the sub-segment with respect to the total number of black pixels in the skeleton, the chain code, the length ratio of the distance between end points and the total length of each sub-segment, etc. Fuzzy logic concepts are used to model any variations or uncertainties in the feature values to allow a better and more realistic representation of these features. Moreover, Fuzzy rules is also used for characters classification.

### 3.3. *Neural network classifiers*

Among the many applications that have been proposed for neural networks, character recognition has been one of the most successful. Compared to other methods used in pattern recognition, the advantages most often stated in favor of a neural network approach to pattern recognition are that (1) it requires



**Segment 1 (1 Dot Above)**
**10000 10010XX**
**Segment 2 ( East Curve Small Upper)**
**00010 1000001**
**Segment 3 ( East Curve Large)**
**00010 1000100**

**Segment inter-relationship matrix**
**(2,1) Relationship 100000000**
**(3,1) Relationship 100000000**
**(3,2) Relationship 100000001**

Fig. 9. Complete representation of an Arabic character for the neural network input layer.

less input of knowledge about the problem than other approaches, (2) it is capable of implementing more complex partitioning of feature space, and (3) it is amenable to high-performance parallel-processing implementations. However, the disadvantages of neural network solutions, compared with statistical approaches, include (1) the extensive amount of training required, (2) slower operation when implemented as a simulation on a conventional computer, and (3) the unavailability of a detailed understanding of the decision-making process that is being used (i.e. the decision surfaces in feature spaces).[83]

Amin and Al-Sadoun[84-86] proposed a structural approach for recognizing handwritten Arabic characters. The binary image of the character is first thinned using a parallel thinning algorithm and then the skeleton of the image is traced from right to left using 3 × 3 window in order to build a graph to represent the character. Features like straight lines, curves and loops are then extracted from the graph. Finally, a five layer artificial neural network is used for the character classification. Each character is classified in term of the segments used in the system such as dot, hamza, line, curve and loop. The relationships between the segments are encoded in the object inter-relationship matrix. The overall design of the input layer uses 150 neurons. Figure 9 illustrates an example of the character representation using this input layer design.

Altuwaijri and Bayoumi[87] introduced a system for recognizing printed Arabic words using artificial Neural Networks (NN). The system can be described into three different steps: first the Arabic input word is segmented into characters using an approach similar to that of reference (56). Next, six moments are used for extracting features from the segmented characters feeding it to the neural network. Finally, a multi-layer perceiptron network with back-propagation learning with one hidden layer is used to classify the character.

Finally, Amin and Mansoor[88] used artificial neural networks for recognizing Arabic printed text. The technique can be summarized into three major steps: The first step is pre-processing in which the original image is transformed into a binary image utilizing a 300 dpi scanner and then forming the connected component. Second, global features of the input Arabic word are then extracted such as number subwords, number of peaks within the subword, number and position of the complementary character, etc. Finally, a three layer artificial neural network is used for the word classification. The overall design of the input layer uses a total of 270 neurons.

### *3.4. Stochastic methods*

Hidden Markov models (HMM) have now become the prevalent paradigm in automatic speech recognition.[89-91] Recently, several researchers in handwriting recognition have tried to transpose the HMM technology to their field after realizing that word images could be assimilated to sequences of observations.[92-101] HMM's form a family of tools for modeling sequential processes in a statistical and generative manner. Their reputation is due to the results attained in speech recognition which derive mostly from the existence of automatic training techniques and the advantages of the probabilistic framework.

An HMM can be defined by: (1) a set of states $\{S\}$, with an initial state $S_I$ and a final sate $S_F$; (2) The transition probability matrix, $\mathrm{A} = \{a_{ij}\}$, where $a_{ij}$ is the transition probability of taking the transition from state i to state j; (3) The output probability matrix $B$. For a discrete HMM, $B = \{\mathrm{b}_j(O_k)\}$, where $O_k$ represents a discrete observation symbol. For a continuous HMM, $\mathrm{B} = \{b_j(x)\}$, where $x$ represents continuous observations of $k$-dimensional random vectors. If the initial state distribution $\pi = \{\pi_i\}$, the complete parameter set of the HMM can be expressed compactly as:

$$\lambda = (A, B, \pi) \tag{5}$$

An HMM can be based either on discrete observation probability distributions or continuous mixture probability density function. In the discrete HMM, the discrete probability distributions are sufficiently powerful to characterize any random event with a reasonable number of parameters. The principal advantage of continuous HMM is the ability to directly model the parameters of a continuous signal. A semi-continuous HMM provides a framework for unifying the discrete and continuous HMM's.

Given the definition of HMM, there are three basic problems of interest that must be solved for real world applications: the evaluation problem, the decoding problem, and the learning problem. The solutions to these three problems are forward–backward algorithm, The Viterbi algorithm, and the Baum–Walch algorithm. For more detailed references on the theory and computation of HMM, the readers may consult reference (102).

The major challenge in the Arabic writing recognition systems come from the cursive nature of the data. Therefore, none of the existing HMM systems, if any, were able to accurately recognize an unconstrained Arabic handwritten cursive script. However, few researchers applied HMM on Arabic printed text.

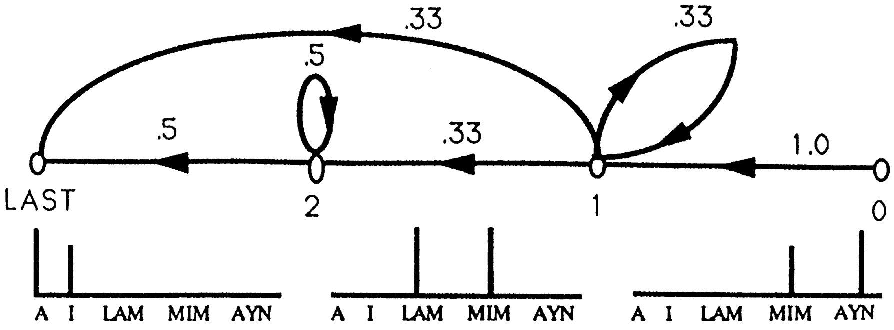


Fig. 10. HMM associated to the word: ala / ayn lam a/.

For example, Amin and Mari[57] used Viterbi Algorithm to enhance the recognition of multi-font Arabic printed text. Upon failure to recognize a given word, a lattice of character hypotheses is generated with it's probabilities using different models. Figure 10 shows the HMM used for the word "ala". A discrete probability density function (PDF) is shown at every state of the word model. Each bin of the PDF represents the probability of observing a given character. It is determined by combining the frequencies of trigrams extracted from the dictionary and the OCR confidence score for each character. For each sequence of the lattice, the system determines the maximum likelihood state sequence using the Viterbi algorithm which maximizes the joint probability of the observation and the state sequence.

Schwartz *et al.*[53] presented a method for reusing existing continuous speech recognition package (the BBN BYBLOS) on printed Arabic text. The idea is that HMM algorithms for training and recognition are language independent; only the lexicon and the training data can make the difference in their application. Similar to speech where data is a single continuous utterance, text line image is used as an entry to the system. The observation sequence is composed of feature vectors computed as a function of horizontal position within the line (see Fig. 11). A frame, defined as a narrow vertical strip, with a width that is a small fraction (typically about 1/15) of the height of the line is divided into 20 equal overlapping cells. Vector components are language independent and highlight simple properties as local intensity, vertical and horizontal derivative of intensity, local slope, etc.

The system uses a right-to-left HMM for each character. A model for a word is obtained by the concatenation of its character models. The forward–backward training algorithm is used for deriving the maximum likelihood estimates of the model parameters. The algorithm is reinforced by ensuring distributions to cluster similar states with insufficient training and retraining the weights of the cluster states. The authors also used two models: the lexicon (obtained by using a large text corpus) and the language model. The language model could be a bigram or trigram which contains probabilities of words in the lexicon. In the recognition phase, a multi-pass search algorithm[103] is used instead of the Viterbi algorithm because of the largeness of the state space. The system has been tested for multi-font texts with a large Arabic database resulting in an average character error rate of 1.9%.

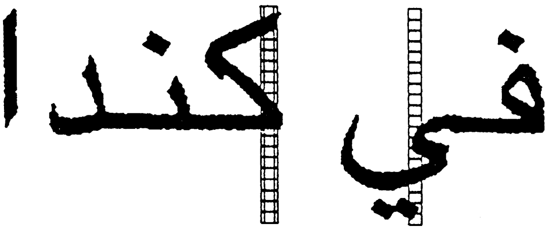


Fig. 11. Feature extraction for a line of text.

Mahjoub[53] followed a more conventional scheme for the recognition of isolated on-line Arabic characters. Each character is represented by an observation sequence, determined from a list of normalized radial distances. During training, the forward–backward algorithm is applied to derive the maximum likelihood density estimate.

Similar to Latin character recognition systems, the application of 1D–HMM, limited to linear sequence observation, to Arabic words is not the best approach. An alternative approach is to enlarge the HMM by defining "bidimensional" models.[104–106] However, this approach resulted in exponential complexity for the recognition process. Kuo and Aggazi[104] suggested a solution for this problem by splitting the image into bands (horizontally or vertically) and associating a 1D-HMM for each band. Moreover, left-to-right models (called secondary) are proposed in the horizontal direction because of Latin characters stability in this direction. In addition, another model in the vertical direction (called principal with super-states) correlates the observation generated by the horizontal model. This architecture is attractive, however, it poses a line independence problem of the super-state. Several solutions have been proposed in references [104] and [105]. The first solution is resolved by adding a post–processing module taking into account

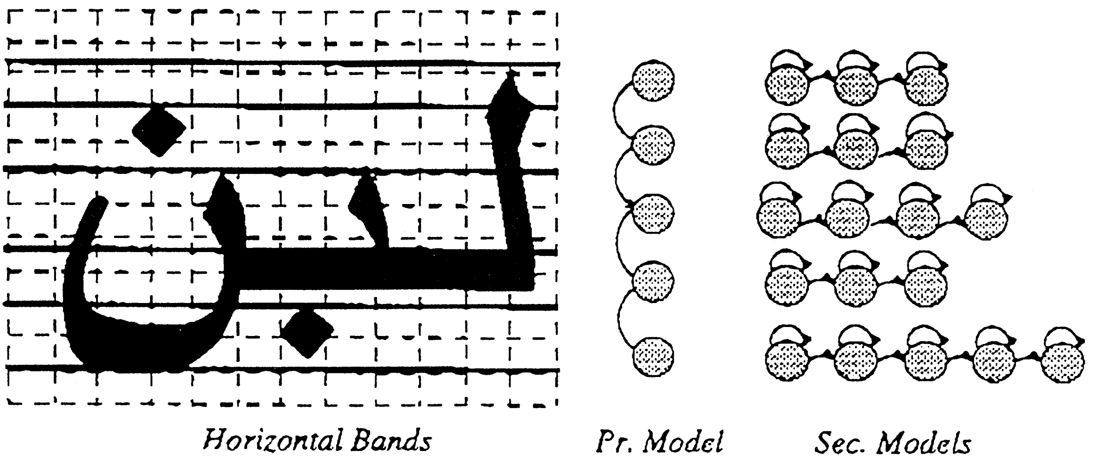


Fig. 12. PAW separated into six super-states.

the duration, while the second solution is to compute the equivalent classes of super-state.

Finally, Ben Amara and Belaid[55] used an efficient super-state duration distribution modeling. The application deals with connected chains of Printed Arabic Words (PAW). Bands are determined according to the topology of each PAW. The system uses several features such as: stem, upper dots, loop, lower dots and descender, etc.. (Fig. 12). The observation for secondary models is composed of successive segments found in lines, coding their duration (length) and location. During training, the image is segmented into vertical bands given duration and lines belonging to super-states. The probability distribution of the duration is calculated by the estimation of the frequency of a band height for a given super-state.

## 4. CONCLUSIONS

This paper presented the problems related to printed and handwritten Arabic characters, and much of the important research work was briefly described in an attempt to present the current status of Arabic character recognition research. This is still an open research area and there is no commercial Arabic OCR system available yet. This is because of the segmentation problem, which is in fact similar to the segmentation of cursive script in many languages, and because of the complexity of Arabic characters. Moreover, all the algorithms presented in this paper deal with unvocalized text and the recognition of vowel diacritics is an extremely important research area in the Arabic language.

However, it is very difficult to give comparative results for the methods proposed so far. Most of the methods used for Arabic handwriting were tested on small and different Databases created by only a few people as there is not any common Arabic Database available. This is also true for all the methods used for printed Arabic characters. This is the reason why the results in the previous sections, which are dedicated to the description of the techniques, were not included.

As stated previously, no vital computational techniques in this area have yet been fully explored. As such, this field is of importance for future research.

## REFERENCES

1. J. Kim and C. C. Tappert, Handwriting recognition accuracy versus tablet resolution and sampling rate, *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal, pp. 917–918 (1984).
2. J. R. Ward and T. Kuklinski, A model for variability effects in handprinted with implication for the design of handwritten character recognition system, *IEEE Trans. Man Cybernet.* **18,** 438–451 (1988).
3. F. Nouboud and R. Plamondon, On-line recognition of handprinted characters: Survey and beta tests, *Pattern Recognition* **25,** 1031–1044 (1990).
4. C. Tappert, C. Y. Suen and T. Wakahara, The state of the art in one-line handwriting recognition, *IEEE Trans. Pattern Anal Mach. Intell.* **PAMI-12,** 787–808 (1990).
5. A. Amin and A. Kaced, Reconnaisances des caracters Arabes Manuscrits, Actes Congres AFCET Informatique Logiciel et materiel applications et implications, pp. 35–44 (1979).
6. A. Amin, A. Kaced, J. P. Haton and R. Mohr, Handwritten Arabic characters recognition by the IRAC system, *Proc. 5th Int. Conf. on Pattern Recognition*, Miami, pp. 729–731 (1980).
7. A. Amin and A. Shoukry, Topological and statistical analysis of line drawing, *Pattern Recognition, Lett.* **1,** 365–374 (1983).
8. M. S. El-Wakil, On-line recognition of handwritten isolated arabic characters, *Pattern Recognition*, **22**(2), 97–105 (1989).
9. T. S. El-Sheikh and S. G. El-Taweel, Real-time Arabic handwritten character recognition, *Pattern Recognition* **23,** 1323–1332 (1990).
10. A. Amin, Issue on Arabic character recognition, *Arabian J. Sci. Engng* **18,** 319–341 (1993).
11. A. Alimi and O. Ghorbe, The analysis in an on-line recognition system of Arabic handwritten characters, *Proc. 3rd Int. Conf. on Document Analysis and Recognition*, Canada, pp. 890–893 (1995).

12. A. Amin, IRAC: Recognition and understanding systems, in *Applied Arabic Linguistic and Signal and Information Processing*, ed., pp. 159–170. R. Descout, Hemisphere, New York (1987).
13. A. Amin, Machine recognition of handwritten Arabic word by the IRAC II system, *Proc. 6th Int. Conf. on Pattern Recognition*, Munich, Germany, pp. 34–36 (1982).
14. E. M. Riseman and R. W. Ehrich, Contextual word recognition using binary diagrams, *IEEE Trans. Comput.* **c-20,** 397–403 (1971).
15. A. Amin, G. Masini and J. P. Haton, Recognition of handwritten Arabic words and sentences, *Proc. 7th Int. Conf. on Pattern Recognition*, Montreal, pp. 1055–1057 (1984).
16. A. Amin and G. Masini, Machine recognition of cursive Arabic words, Application of Digital image Processing Vol. IV G. Tescher, ed., pp. 1127–1135 (1982).
17. A. Amin and G. Masini, Deux Methodes de Reconnaissance de Mots pour l'Ecriture Arabe Manuscrite, Act. 5'eme Congre's AFCET, Reconnaissance des Formes et Intelligence Artificielle, pp. 837–848, Grenoble 1985).
18. A. Amin, Arabic handwritten recognition and understanding, *Proc. Computer processing of the Arabic Language*, Kuwait, pp. 1–40 (1985).
19. S. Al-Emami and M. Usher, On-line recognition of handwritten Arabic characters, *IEEE Trans. Pattern Anal Mach. Intell.* **PAMI-12,** 704–710 (1990).
20. A. Belaid and G. Masini, Segmentation of line drawings for recognition and interpretation, *Technol. Sci. Informatics* **1**(2), 121–134 (1983).
21. C. Y. Suen, M. Berthod and S. Mori, Automatic recognition of handprinted characters, the state of the art, *Proc. IEEE* **68,** 469–483 (1980).
22. J. Schuermann, Reading machines, *Proc. 6th Int. Conf on Pattern Recognition*, Munich, Germany, pp. 1031–1044 (1982).
23. J. R. Ullmann, Advance in character recognition, in *Application of Pattern Recognition*, K. S. Fu, ed., pp. 197–236 (1982).
24. M. K. Brown and S. Ganapathy, Preprocessing technique for cursive script word recognition, *Pattern Recognition* **19**(1), 1–12 (1983).
25. R. H. Davis and J. Lyall, Recognition of handwritten characters a review, *Image Vision Comput.* **4,** 208–218 (1986).
26. V. K. Govindan and A. P. Shivaprasad, Character recognition—A review, *Pattern Recogn.* **23,** 671–683 (1990).
27. S. Srihari, From pixel to paragraphs: the use of models in text recognition, *Proc. 2nd Ann. Symp. on Document Analysis and Information Retrieval*, Las Vegas, USA, pp. 47–64 (1993).
28. E. Lecolinet and O. Baret, *Cursive Word Recognition: Methods and Strategies*, *Fundamentals in Handwriting Recognition*, S. Impedovo,ed., pp. 235–263 (1994).
29. O. D. Trier, A. Jain and T. Taxt, Feature extraction methods for character recognition, *Pattern Recognition* **29,** 641–662 (1996).
30. J. W. The and R. T. Chin, On image analysis by the methods of moments, *IEEE Trans. Pattern Anal Mach. Intell.* **PAMI-10,** 496–508 (1988).
31. S.J. Raudys and A. Jain, Small sample size effect in statistical pattern recognition, *IEEE Trans. Pattern Anal Mach. Intell.* **PAMI-1,** 252–264 (1991).
32. T. Matsunage and H. Kida, An experimental study of learning curves for statistical pattern classifiers, *Proc. 3rd Int. Conf. on Document Analysis and Recognition*, Canada, pp. 1103–1106 (1995).
33. M. Berthod and J. P. Maroy, Learning in syntactic recognition of symbols drawn on a graphic tablet, *Compu. Graphiics Image Process.* **9,** 166–182 (1979).
34. A. Belaid and J. P. Haton, A syntactic approach for handwritten mathematical formula recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6,** 105–111 (1984).
35. R. Freund, Syntatic analysis of handwritten characters by quasi-regular programmed array grammars, in *Advances in Structural and Syntactic Pattern Recognition*, H. Bunke, ed., pp. 310–319 (1992).
36. Y. Lecun, Backpropagation applied to handwritten zip code, *Neural Comput.* **1,** 541–551 (1989).
37. I. Guyon, Application of neural network to character recognition, in *Character and Handwriting Recognition in Expanding Frontiers*, P. S. P. Wang, ed., pp. 353–382 (1991).
38. S-W. Lee and Y-J. Kim, A new type of recurrent neural network for handwritten character recognition, *Proc. 3rd Int. Conf. On Document Ananlysis and Recognition,* Canada, pp. 38–41 (1995).
39. C. Y. Suen and C. L. Yu, Performance Accessmant of a character recognition Expert System, Int. Expert System application EXPERSYS 90, pp. 195–200 (1990).
40. L. Likfooman_Solem, H. Maiutre and C. Sirait, An expert and vision system for analysis of Hebrew characters and autheutication of manuscript, *Pattern Recognition* **24,** 121–137 (1991).
41. A. Amin, M. Bemford and A. Hoffman, A knowledge acquisition technique for recognizing Hand-printed Chinese characters, *Proc. 13th Int. Conf. on Pattern Recognition*, Austria, pp. 254–258 (1996).
42. A. Amin, D. Ziino and C. Sammut, Recognition of hand printed Latin characters using machine learning, *Proc. 3rd Int. Conf. on Document analysis and Recognition*, Canada, pp. 1098–1102 (1995).
43. A. Amin and P. Compton, Hand-Printed characters recognition using machine learning, *Proc. 5th Int. workshop on Frontiers in Handwriting Recognition*, Essex, England, pp. 247–250 (1996).
44. A. Amin, C. Sammut and K. C. Sum, Learning to recognize Hand-printed Chinese characters using Inductive Logic Programming, *Int. J. Pattern Recognition Artificial Intell.* **10,** 829–847 (1996).
45. K. Jambi, Arabic charcter recognition: Many approaches and one decade, *Arab J. Sc. Engng* **16,** 499–509 (1991).
46. B. Al-Badr, and S. Mahmoud, Survey and bibliography of Arabic optical text recognition, *Signal Process.* **41,** 49–77 (1995).
47. A. Amin, Arabic character recognition, in *Handbook of Character Recognition and Document Image Analysis*, H. Bunke and P. S. P. Wang, eds, pp. 349–420, world scientific (May 1997).
48. M. Khemakhem, Reconnaissance de caracters imprimes par comparaison dynamique, These de Doctorate de 3 e'me cycle, University of Paris XI (1987).
49. M. Khemakhem and M. C. Fehri, Recognition of Printed Arabic charcters by comparaison dynamique, *Proc. 1st Kuwait Comput. Conf.* pp. 448–462 (1989).
50. H. Y. Abdelazim and M. A. Hashish, Interactive font learning for Arabic OCR, *Proc 1st. Kuwait Comput. Conf., pp.* 464–482 (1989).
51. H. Y. Abdelazim and M. A. Hashish, Automatic recognition of handwritten Hindi numerals, *Proc. 11th* National Comput. Conf., Dhahran, pp. 287–299 (1989).
52. Z. Emam and M. A. Hashish, Application of Hidden Markov Model to the recognition of isolated Arabic word, *Proc.* 11*th National Comput. Conf.*, Dhahran, pp. 761–774 (1989).
53. R. Schwartz, C. LaPre, J. Makhoul, C. Raphael and Y. Zhao, Language independent ocr using a continuous speech recognition system. *Proc. 13th Int. Conf., on Pattern Recognition*, Vol. C, pp 99–103, Vienna, Austria (1996).

54. M. A. Mahjoub, Choix des parametres lies a l"apprentissage dans la reconnaissance en ligne des caractercs arabes par les chaines de markov cachees, in *Forum de la Recherche en Informatique*, Tunis, (Juillet 1996).
55. N. BenAmara and A. Belaid. Printed PAW recognition based on planar hidden Markov models. In *13th Int. Conf. on Pattern Recognition*, Vol. B, Vienna, Austria, (1996).
56. A. Amin and G. Masini, Machine recognition of mutifonts printed Arabic texts, *Proc. 8th Int. Conf. on Pattern Recognition*, Paris, 392–395 (1986).
57. H. Almuallim and S. Yamaguchi, A method of recognition of Arabic cursive handwriting, *IEEE*, *Trans. Pattern Anal. Mach. Intell.* **PAMI-9,** 715–722 (1987).
58. A. Amin and J. F. Mari, Machine recognition and correction of printed Arabic text, *IEEE Trans. Man Cybernet.* **9,** 1300–1306 (1989).
59. A. Amin and S. Al-Fedaghi, Machine recognition of printed Arabic text utilising a natural language morphology, *Int. J. Man–Machine Stud.* **35,** 769–788 (1991).
60. T. El-Sheikh and R. Guindi, Computer recognition of Arabic cursive script, *Pattern Recognition* **21,** 293–302 (1988).
61. F. El-Khaly and M. Sid-Ahmed, Machine recognition of optically captured machine printed Arabic text, *Pattern Recognition* **23,** 1207–1214 (1990).
62. H. Abdelazim and M. Hashish, Arabic reading machine, *Proc. 10th National Computer Conf.* Riyadh, Saudi Arabia, pp. 733–740 (1988).
63. A. Shoukry, A sequential algorithm for the segmentation of typewritten Arabic digitized text, *Arabian J. Sci. Engng* **16,** 543–556 (1991).
64. T. Wakayama, A core-line tracing algorithm based on maximal square moving, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-4,** 68–74 (1982).
65. V. Margner, SARAT—A system for the recognition of Arabic printed text, *Proc.* 11*th Int. Conf. on Pattern Recognition*, 561–564 (1992).
66. B. M. Kurdy and A. Joukhadar, Multifont recognition system for Arabic characters, *Proc. 3rd Int. Conf. Exhibition on Multi-lingual Computing* (*Arabic and Roman Script*), U.K. pp. 731–739 (1992).
67. A. Amin and H. Al-Sadoun, A segmentation technique of Arabic text, *Proc. 11th Int. Conf. on Pattern Recognition*, The Netherlands, pp. 441–445 (1992).
68. H. B. Al-Sadoun and A. Amin, A new structural technique for recognizing printed Arabic text, *Int. J. Pattern Recognition Artificial Intell.* **9,** 101–125, (1995).
69. H. Freeman, On the encoding of arbitrary geometric configuration, *IEEE. Trans. Electronic Comp.* **EC-10,** 260–268 (1968).
70. A. Nouh, A. Sultan and R. Tulba, An approach for Arabic character recognition, *J. Engng Sc.* **6**(2), 185–191 (1980).
71. B. Parhami and M. Taraghi, Automatic recognition of printed Farsi texts, *Pattern Recognition* **14,** 395–403 (1981).
72. S. Saadallah and S. Yacu, Design of an Arabic character reading machine, *Proc. Computer Process. Arabic Language*, Kuwait (1985).
73. D. Forney, The Viterbi algorithm, *Proc. IEEE* **61,** 268–278 (1973).
74. S. El-Dabi, R. Ramsis and A. Kamel, Arabic charcter recognition system: statistical approach for recognizing cursive typewritten text, *Pattern Recognition* **23,** 485–495 (1990).
75. H. Al-Yousefi and S. S. Udpa, Recognition of Arabic characters, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-14,** 853–857 (1992).
76. A. Nouh, A. Ula and A. Sharaf-Edin, Boolean recognition technique for typewritten Arabic character set, *Proc. 1st King Saud Univ. Symp. on Computer Arabization*, Riyadh, pp. 90–97 (1987).
77. S. Al-Fedagi and A. Amin, Automatic correction of spelling errors in Arabic, *J. Univ. Kuwait* **19,** 175–194 (1992).
78. F. Haj Hassan, Arabic character recognition, Arab School of Science and Technology. pp. 23–30 (1985).
79. F. Haj Hassan and W. Haj Ali, Printed Arabic text recognition, *Arab. J. Sci. Engng* **16,** 511–518 (1991).
80. B. Al-Badr and R. Haralick, Segmentation-free word recognition with application to Arabic, *Proc. 3rd Int. Conf. on Document Analysis and Recognition*, Montreal, pp. 355–359 (1995).
81. M. Sano, T. Kosaki and F. Bouslama, Fuzzy structural approach for recognition of handwritten Arabic characters, *Proc. Int. Conf. on Robotics Vision and Parallel Processing for Industrial Automation*, Ipon, Malaysia, pp. 252–257 (1996).
82. F. Bouslama, Arabic character recognition by Fuzzy techniques, *Proc. 5th European Congress on Intelligent Techniques and Soft Computing,* Aachen, Germany (1997).
83. R. P. W. Duin, Superlearning and neural network magic, *Pattern Recognition Lett.* **15,** 215–217 (1994).
84. A. Amin and H. Al-Sadoun, Arabic character recognition system using artificial neural network, Int. Workshop on Appl. of Neural Networks to Telecommunications, USA, pp. 99–105 (1993).
85. A. Amin and H. Al-Sadoun, Handprinted Arabic character recognition system, *Proc. 12th Int. Conf. on Pattern Recognition*, pp. 536–539 (1994).
86. A. Amin and H. Al-Sadoun, Handprinted Arabic character recognition system using an arficial neural network, *Pattern Recognition* **29,** 663–675 (1996).
87. M. Altuwaijri, and M. Bayoumi, Arabic text recognition using neural networks, Proc. Int. Symp. on Circuits and Systems — ISCAS'94. pp. 415–418 (1994).
88. A. Amin and W. Mansoon, Recognition of printed Arabic text using Neural networks, *Proc. 4th Int. Conf. on Document Analysis Recognition,* Ulm, Germany, (August 1997).
89. L. R. Rabiner and B. H. Juang, An introduction to hidden Markov models, *IEEE ASSP Mag.* **3**(1), 4–16 (1986).
90. A. B. Poritz, Hidden Markov models: a guided tour, *Proc. IEEE*, *Int. Conf. Acoust. Speech*, *Signal Process.* pp. 7–13 (1988).
91. L. R. Rabiner, A tutorial on Hidden Markov Models and selected application in speech recognition, *Proc. IEEE* **77**(2) 257–286 (1989).
92. A. Kundu, Y. He and P. Bahl, Recognition of handwritten word: first and second order Hidden Markov models based approach, *Pattern Recognition* **22,** 283–297 (1989).
93. M. Y. Chen, A. Kundu and S. N. Srihari, Unconstrained handwritten word recognition using continuous density variable duration hidden Markov models, *Proc. IEEE*, *Int. Conf. on Acoust. Speech Signal Process.* (ICASSP'93) (1993).
94. M. Y. Chen and A. Kundu, An alternative to variable duration HMM in handwritten word recognition, *Proc. 3rd Int. Workshop on Frontiers in Handwritten Recognition,* pp. 82–91 (1993).
95. M. Gilloux,J. M. Bertille and M. Leroux, Recognition of handwritten words in a limited dynamic vocabulary, *Proc. 3rd Int. Workshop on Frontiers in Handwritten Recognition,* pp. 417–422 (1993).
96. H. S. Park and S. W. Lee, Off-line recognition of large set handwritten Hangul (Korean script) with HMM, *Proc. 3rd Int. Workshop on Frontiers in Handwritten Recognition,* pp. 51–61 (1993).

97. M. Gilloux, Hidden Markov models in handwritten recognition, *Fundamentals in Handwriting Recognition*, S. Impedovo, ed., Springer-Verlag, pp. 264–288 (1994).
98. C. B. Bosc and S. Kuo. Connected and degraded text recognition using hidden Markov model, *Pattern Recognition*, **27,** 1345–1363 (1994).
99. A. El-Yacoubi, J-M Bertille and M. Gilloux, Conjoined location and recognition of street names within a postal address delivery line, *Proc. 3rd Int. Conf. on Document Analysis and Recognition, ICDAR'95,* pp. 1124–1127. Montreal, Canada, (1995).
100. M.Y. Chen, A. Kundu, J Zhou, and S. N. Srigari, Off-line handwritten word recognition using hidden Markov mode, in *USPS'92* (1992).
101. M. Chen, A. Kundu and J. Zhou, Off-line handwritten word recognition using HMM type stochastic network, *IEEE Trans. Pattern Analysis Mach. Intell.* **PAMI 16,** 481–496 (1994).
102. L. Rabiner and B-H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs NJ.
103. L. Nguyen et al., The 1994 bbn/byblos speech recognition system. *Proc. ARPA Spoken Language Technology Workshop*, pp. 77–81. Morgan Kaufmann, Austin, TX (1995).
104. S. Kuo and O. E. Agazzi, Keyword spotting in poorly printed documents using pseudo 2–D hidden Markov moddels, *IEEE Trans. Pattern Analysis and Machine Intell.* **PAMI 16,** 842–848 (1994).
105. M. Gilloux, Reconnaissance de chiffres manscrits par modele de Markov pseudo 2-D, in act du 3eme Colloque National sur l'Erit et le Document, 11–17, Rouen, France (1994).
106. G.Saon and A. Belaid, Recognition of unconstrained handwriten word using Markov random fields and HMM, *Proc. 5th Int. Workshop on Frontiers in Handwritten Recognition* (1996).

**About the Author**—ADNAN AMIN received the B.Sc. Degree in Mathematics and Diploma in Computer Science from Baghdad University in 1970 and 1973, respectively. He received D.E.A. in Electronics from the University of Paris XI (Orsay) in 1978 and presented his Doctorate D'Etat (D.Sc.) in Computer Science to the University of Nancy I (CRIN), France, in 1985. From 1981 to 1985, Dr Amin was Maitre Assistant at the University of Nancy II. Between 1985 and 1987 he worked in INTEGRO (Paris) as Head of Pattern Recognition Department. From 1987 to 1990, he was an Assistant Professor at Kuwait University and joined the School of Computer Science and Engineering at the University of New South Wales, Australia, in 1991 as a Senior Lecturer.

Dr Amin's research interests are in Pattern Recognition and Artificial Intelligence (Document Image understanding ranging from character/word recognition to integrating visual and linguistic information in document composition), neural networks, machine learning and knowledge acquisition. He is a member of the Editorial Boards of International Journal of Pattern Recognition and Artificial Intelligence, and Journal of Pattern Analysis and Applications, and has served on the Program and Technical committees of several International Conferences and Workshops. He is serving as cochairman of SPR and SSPR to be held in Sydney 1998. Dr Amin is a member of IEEE, the IEEE Computer Society and the ACM.