1

On Offline Arabic Character Recognition

Muhammad Sarfraz
Department of Information and Computer Science, King Fahd University of Petroleum and Minerals, Dhahran 31261, Kingdom of Saudi Arabia

Abdulmalek Zidouri
Department of Electrical Engineering, King Fahd University of Petroleum and Minerals, Dhahran 31261, Kingdom of Saudi Arabia

Syed Nazim Nawaz
Information Technology Department, College of Business Administration, Jeddah 21361, Kingdom of Saudi Arabia

Machine recognition of characters has received considerable research interest in the area of pattern recognition in the past few decades. This chapter presents the design and implementation of a system that recognizes machine printed Arabic characters. The proposed system consists of four stages: preprocessing of the text, segmentation of the text into individual characters, feature extraction using the moment invariant technique and recognition of characters based on two approaches. In the preprocessing of the text we deal with two problems: isolated pixel removal and drift detection and correction. Next, the given text is segmented into individual characters using horizontal and vertical projection profiles. Moment invariants are used for feature extraction for each character. Finally, the system is trained and the characters are recognized. Recognition of characters is attempted using two approaches, a syntactic approach and a neural network approach.

1. Introduction

In the past two decades, valuable work has been carried out in the area of character recognition and a large number of technical papers and reports have been devoted to this topic. Character recognition systems offer potential advantages by providing an interface that facilitates interaction between man

Computer-Aided Intelligent Recognition Techniques and Applications Edited by M. Sarfraz

and machine. Some of the applications of OCR include automatic processing of data, check verification and a large variety of banking, business and scientific applications. OCR provides the advantage of little human intervention and higher speed in both data entry and text processing, especially when the data already exists in machine-readable forms.

With all the above advantages considered, Arabic character recognition proves to be an interesting area for research. Many papers concerned with Optical Character Recognition (OCR) have been reported in the literature [1]. Several recognition techniques have been used over the past few decades by many researchers. These techniques were applied for the automatic recognition of both machine and hand-printed characters. An immense amount of research has been carried out on the recognition of Latin and Chinese characters. Against this background, only few papers have addressed the problem of Arabic character recognition. One of the main reasons behind this is the difficulty involved in processing printed Arabic text. The connectivity and variant shape of characters in different word positions present significant difficulty in processing of the text. Some of the features of Arabic script that have limited the research in this area are the following:

- Arabic script constitutes 28 characters, in addition to ten numerals.
- Each character can appear in four different shapes/forms depending on the position of the word (Beginning form, BF; Middle form, MF; Isolated form, IF; and End form, EF). Table 1.1 shows some Arabic characters in their different forms.
- The Arabic characters of a word are connected along a baseline. A baseline is the line with the highest density of black pixels. This calls for different segmentation methods from those used in other unconnected scripts.
- Characters in Arabic script are connected even when typed or printed.
- In addition to connectivity, vowel diacritic signs are an essential part of written Arabic. Vowels are represented in the form of overscores or underscores (see Figure 1.1).

Table 1.1 Different forms of Arabic characters.

IF	BF	MF	EF	IF	BF	MF	EF
أ	أ	ـا	ـا	ظ	ظـ	ـظـ	ـظ
ب	بـ	ـبـ	ـب	ك	كـ	ـكـ	ـك
ت	تـ	ـتـ	ـت	ل	لـ	ـلـ	ـل
ث	ثـ	ـثـ	ـث	م	مـ	ـمـ	ـم
ج	جـ	ـجـ	ـج	ن	نـ	ـنـ	ـن
ح	حـ	ـحـ	ـح	ع	عـ	ـعـ	ـع
خ	خـ	ـخـ	ـخ	غ	غـ	ـغـ	ـغ
د	د	ـد	ـد	ف	فـ	ـفـ	ـف
ذ	ذ	ـذ	ـذ	ق	قـ	ـقـ	ـق
ر	ر	ـر	ـر	س	سـ	ـسـ	ـس
ز	ز	ـز	ـز	ش	شـ	ـشـ	ـش
ص	صـ	ـصـ	ـص	ه	هـ	ـهـ	ـه
ض	ضـ	ـضـ	ـض	و	و	ـو	ـو
ط	طـ	ـطـ	ـط	ي	يـ	ـيـ	ـي

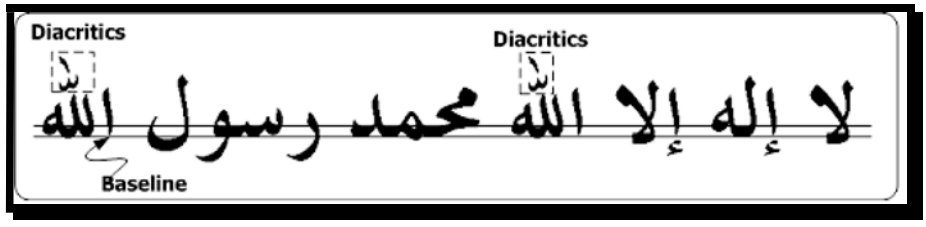


Figure 1.1 The baseline and diacritics of a given Arabic text.

Figure 1.2 Example of different characters with the same body.

- Many Arabic characters have dots, which are positioned above or below the letter body. Dots can be single, double or triple. Different Arabic letters can have the same body and differ in the number of dots identifying them, as shown in Figure 1.2.
- The problem of overlapping makes the problem of determination of spacing between characters and words difficult.
- Characters in an Arabic script usually overlap with their neighboring letters depending on their position in the word. The degree of overlapping varies depending on the size and style of the character.

Figure 1.3 depicts some of the features of Arabic script. As shown in the figure, the ligature in the beginning and the middle of the figure consists of two vertically stacked characters. The text baseline is between the two horizontal lines. The text shown in Figure 1.3 is a text with diacritics.

These and other special characteristics make it impossible to directly adapt English or other text recognition systems to Arabic text recognition.

The rest of the chapter is organized as follows: Section 2 describes the structure of the proposed OCR system and the work that has been carried out in this area. The preprocessing stage is explained in Section 3. Section 4 discusses the proposed method for segmentation of text into individual characters. Section 5 explains feature extraction based on the moment invariant technique by Hu [2]. Section 6

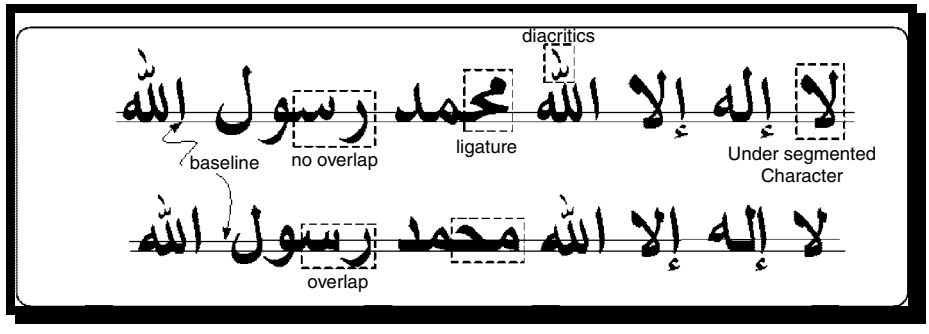


Figure 1.3 The characteristics of an Arabic text.

discusses the two approaches that have been used for recognition of characters. Finally, Section 7 discusses experimental analysis and the chapter is concluded in Section 8.

2. Structure of the Proposed OCR System

Figure 1.4 shows the block diagram of the proposed character recognition system. The system involves four image-processing stages: Preprocessing, Segmentation, Feature Extraction and Classification. The recognition of any script starts by acquiring a digitized image of the text using a suitable scanning system. Next, the preprocessing of the image takes place. There are two processes for handling the acquired image in the proposed system: drift correction and removal of isolated pixels. In the second stage, the segmentation of the text into individual characters takes place. Segmentation of connected script text into individual characters is the most important and difficult stage in an Arabic optical character recognition system. To achieve good recognition rate and performance, a character recognition system should have a good segmentation algorithm.

Since the early eighties there have been reports about successful research projects in the field of printed Arabic character recognition. Connectivity of characters being an inherent property of Arabic writing mean that it is of primary importance to tackle the problem of segmentation in any potentially practical Arabic OCR system. A state of the art on offline Arabic character recognition can be found in [1].

Segmentation of the text into individual characters can be achieved using a variety of techniques. Different approaches for the segmentation of the text in to individual characters are presented in [3]. Segmentation of text in to characters can be achieved based on the geometrical and topological features of the characters [4,5,6], closed contours [7] and horizontal and vertical projection of pixel rows and columns [8–11].

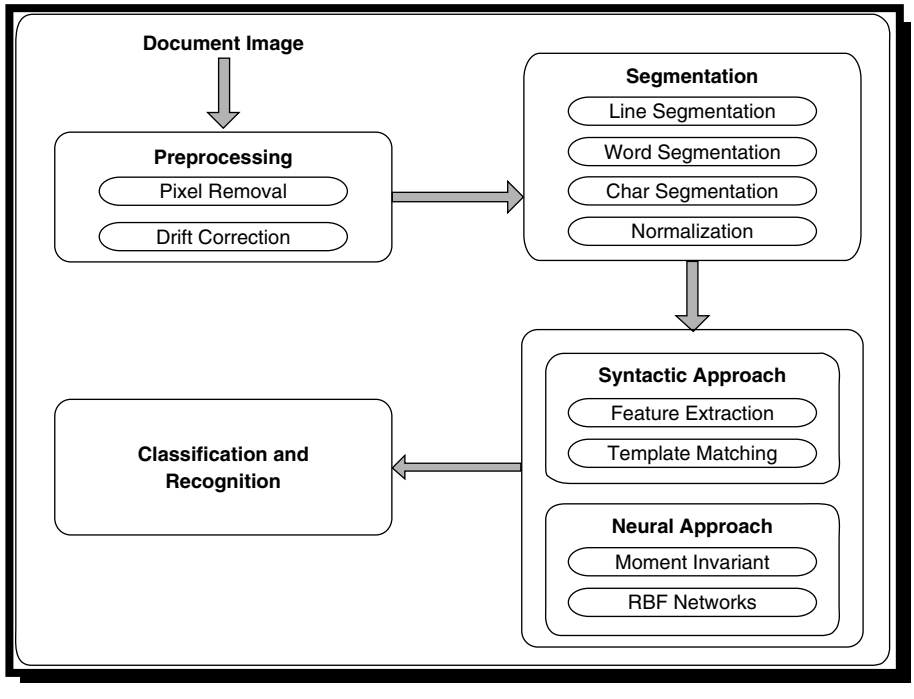


Figure 1.4 Structure of the proposed OCR system.

Almuallim and Yamaguchi [4] proposed a structural recognition technique for Arabic handwritten words. In their approach, an Arabic word is segmented into strokes which are classified based on their geometrical and topological properties. 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.

Segmentation of text can also be achieved using closed contours. The SARAT system [7] used outer contours to segment Arabic words into characters. The word is divided into a series of curves by determining the start and end points of words. Whenever the outer contour changes sign from positive to negative, a character is segmented.

Hamami and Berkani [8] employed a simple segmentation method. Their method is based on the observation of projections of pixel rows and columns, and uses these projections to guide the segmentation process. In their method, undersegmentation of characters, which is a common problem, is treated by considering the entire undersegmented set of characters as a single character. On the other hand, the other common problem of oversegmentation is solved by taking care of the situations in which it may occur and resolving them accordingly.

After segmentation, numerical features of the character are extracted. Features of the characters are extracted and matched with the existing ones to identify the character under investigation. Trier *et al.* [12] present a survey of the different feature extraction methods available in the literature. Features can be extracted using template matching [13], closed contours [7], zoning [14] and moment invariants [2,4,15–18].

Fakir and Hassani [19,20] utilized moment invariant descriptors developed by Hu [2] to recognize the segmented Arabic printed characters. A look-up table is used for the recognition of isolated handwritten Arabic characters [21]. 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 a Freeman Code. Jambi [22] adopted a look-up table for the recognition of isolated Arabic characters.

Using nonlinear combinations of geometric moments, Hu [2] derived a set of seven invariant moments, which has the desirable property of being invariant under image translation, scaling and rotation. Abdul Wahab [15] has shown, based on the work of Hu [2], that there exists a set of invariant momentum features that can be used for classification purposes.

The step following the segmentation and extraction of appropriate features is the classification and recognition of characters. Many types of classifier are applicable to the OCR problem. Recognition of a pattern can be done using a statistical approach (decision theoretic), a syntactic approach or a neural approach. Among the three, syntactic and neural approaches are showing promising results compared to the statistical approach.

Chinveerphan *et al.* [23] and Zidouri *et al.* [11,24,25] presented a structural approach for the recognition of Arabic characters and numerals. Their approach is based on the modified Minimum Covering Run (MCR) expression. The given Arabic text is represented in its MCR form and its feature values are extracted. From the feature values, the different character shapes are described and the reference prototypes are built. Finally, the character is identified by matching the data of the document image with the reference prototypes.

Sadoun *et al.* [26] proposes a new structural technique for the recognition of printed Arabic text. Here, the recognition is done by parsing the sentence in which the given image occurs. The advantage of this technique is that it is font independent.

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:

- it requires less input of knowledge about the problem than other approaches;
- it is amenable to high-performance parallel-processing implementation;
- it provides good fault tolerance compared to statistical and structural approaches [1].

Altuwaijri *et al.* [16] used a multilayer perceptron network with one hidden layer and back-propagation learning to classify a character. Their input layer consisted of 270 neurons. Smagt [27] tested the performance of three general purpose neural networks: the feed-forward network, the Hopfield network and the competitive learning network for OCR applications. The classification capabilities of the network are compared to the nearest neighbor algorithm applied to the same feature vectors. The feed-forward and the competitive learning networks showed better recognition rates compared to the Hopfield network. Each of the three techniques is based on learning approaches where an adjustment to the weight is made following each iteration.

One problem with the neural networks approach is that it is difficult to analyze and fully understand the decision making process [28]. It is often quite difficult to analyze which kind of classifier is most suitable for a recognition process. An unbiased comparison between each of the recognition processes is quite difficult and many conditions need to be fulfilled. To have an efficient comparison between the three approaches, the best statistical or structural classifiers have to be compared with the same quality neural network classifier [29]. In this work we try to compare the results of the syntactic approach and the neural network approach for the Arabic OCR problem. The first two stages of preprocessing and segmentation are common to both approaches. The features used and recognition stage has been attempted using the two approaches. Results and discussion about the two methods are illustrated. The syntactic approach gave a higher recognition rate than the neural network approach. Recommendations on how to improve the recognition rate and performance of the system are given at the end of this chapter.

3. Preprocessing

In the preprocessing stage for this work we concentrate on removal of non-useful information that can be considered as noise and skew detection and correction. To remove the noise as isolated pixels for any given pixel, we check for the existence of a neighboring pixel in all the possible eight directions (Figure 1.5). If a pixel exists in any of the possible directions, then the pixel is not an isolated pixel. However, if there is no pixel in any of these directions, then the pixel under investigation is considered to be an isolated pixel and is removed.

The text to be recognized may be transferred to the system slanted. This affects the accuracy of segmentation and recognition. To tackle the problem of skew detection and correction we have employed the following drift correction procedure. First we determine the rotation angle of the text by computing the tangents of all the line segments that can be constructed between any pair of black pixels in the image. Then, the corresponding angles are computed. To reduce the computation cost, one could apply this process just to a selected window of text instead of the whole image, assuming that the whole page is skewed in the same way everywhere.

The angle that has the highest number of occurrences is assumed to be the angle of rotation of the image. After determining the angle of rotation, the baseline drift is corrected by rotating the image by

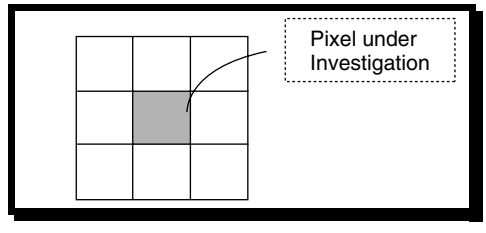
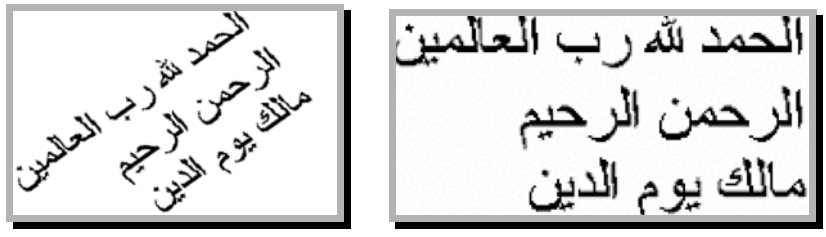


Figure 1.5 Eight-connected components.



(a) (b)

Figure 1.6 (a) Original image skewed by 36°; (b) image after drift correction.

the same angle in the opposite direction. Figure 1.6(a) and Figure 1.6(b) show the original image and the result obtained by applying the preprocessing on an image that is severely skewed by 36 degrees with respect to the horizontal.

4. Segmentation

Segmentation is the process of isolating the individual characters to be passed to the recognition phase. Character segmentation is the most crucial and the most difficult step in Arabic OCR. A poor segmentation process produces misrecognition or rejection. The segmentation process for the character recognition problem can be divided into three levels: line segmentation, word segmentation and character segmentation. Figure 1.7 illustrates the entire segmentation process for the proposed system.

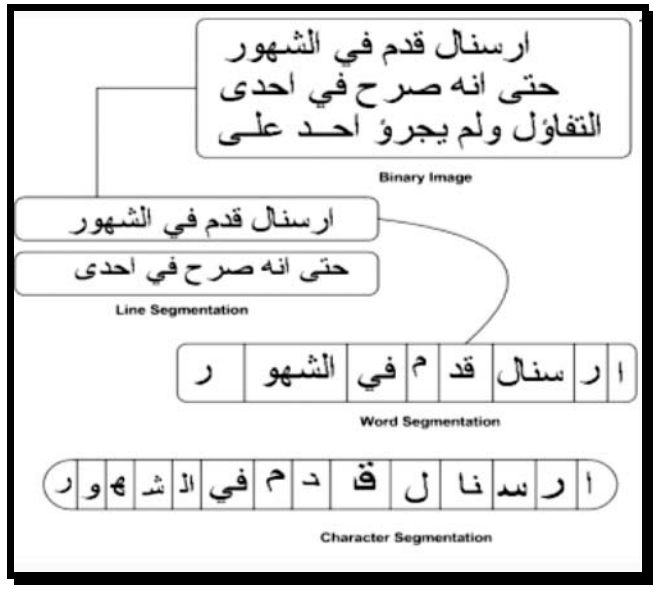


Figure 1.7 The overall segmentation process.

4.1 Line Segmentation and Zoning

After preprocessing, the given Arabic text is first segmented into individual lines of text. Segmentation of text into lines is done using the horizontal projection profile of an image. This is popular because it is efficient and easily implemented. Next, the segmented line of text is divided into four *zones*; namely the baseline zone, the middle zone, the upper zone and the lower zone (Figure 1.8).

4.1.1 Line Segmentation Using Horizontal Projection

First we determine the size of the image. For an image of a particular size we project horizontally along each row of the image. If the number of black pixels encountered during horizontal projection is zero, then we just increment the row number, i.e. we project horizontally (determine the number of black pixels) in the next row. This process is repeated until the number of black pixels along a row is not equal to zero. When this happens we initialize the row number to a variable. The moment it happens it means that a line of text has just started. We again repeat the above process but this time we check for a row number with zero pixels. The occurrence of a row number with zero pixels indicates the end of the line of text. The values of the row numbers indicate the starting and ending positions of a line in the text, and the line is extracted.

4.1.2 Zone Classification

The baseline zone is the zone with the highest density of black pixels. In Figure 1.8 this zone can be identified as the area within the horizontal lines of the histogram. The region just above the baseline zone is the middle zone and then the upper zone. Anything below the baseline is classified to belong to the lower zone. Any zone that is just above the baseline and twice the thickness of the baseline is the middle zone. This middle zone is useful for segmenting the words into individual characters. Figure 1.8 gives the horizontal projection profile of a sample Arabic text.

4.2 Word Segmentation

Next, the line of text is segmented into words or subwords. This is done using the vertical projection of the image. Segmentation of a line of text into words is simple and is similar to the horizontal projection. The only difference between the horizontal and vertical projections is that in the latter, we count the number of black pixels on each column of the image line by line of text. If the number of black pixels is not equal to zero, it indicates a connected part and consequently the text is not segmented. On the other hand, if the number of black pixels in a particular column is equal to zero, the word is segmented. Figure 1.9 shows the vertical projection profile for a given image.

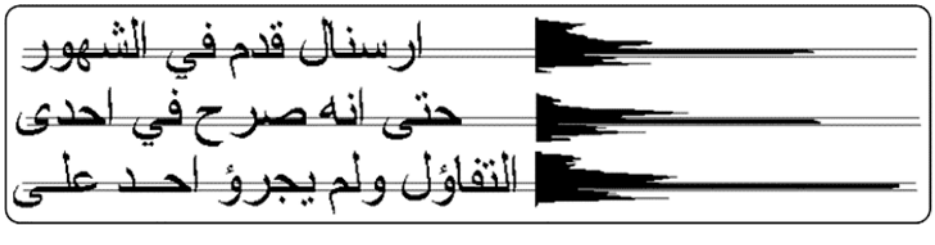


Figure 1.8 Horizontal projection profile for a given Arabic text.

Figure 1.9 Vertical projection profile for a given Arabic text.

4.3 Segmentation of Words into Individual Characters

Finally, the word or subword is segmented into characters. First, the vertical projection of the middle zone is created. The middle zone is the zone right above the baseline. Next, the word is scanned from right to left. A fixed threshold is used for segmenting the word into characters. Whenever the value of the vertical profile of the middle zone is less than two thirds of the baseline thickness, the area is considered a connection area between two characters. Then, any area that has a larger value is considered as the start of a new character, as long as the profile is greater than one third of the baseline. This process is repeated until the full length of the line is exhausted.

The overall segmentation procedure is as follows:

Procedure 1: Line segmentation

```
1. Identify the height and width of the image.
2. Traverse from top to bottom of the entire image.
3. For each row, identify the number of black pixels.
   While the number of black pixels is zero:
   (a) Increment row number;
   (b) Move until the number of black pixels is not equal to zero;
   (c) Initialize the row number to startx.
   While the number of black pixels is not zero:
   (a) Increment startx.
   (b) If the number of black pixels is zero:
   (a) Assign startx to endx.
   (b) Row numbers between startx and endx indicate a line of text.
4. End for.
```

Procedure 2: Word and character segmentation

```
1. Project vertically and repeat the above process to segment lines into
   individual parts or words.
2. For each word:
   (a) Project horizontally and determine the rows with the highest value.
       This zone of large row values is the baseline zone.
   (b) Middle zone = 2*baseline zone thickness.
   (c) If vertical projection(middle zone area) > 1/2 (baseline zone) or
       < 2/3 (baseline zone) The area is a connection area.
       Else isolate the character.
       End if.
3. End for.
```

5. Feature Extraction

The next stage in our Arabic recognition system is the feature extraction stage. This is a very important stage in any recognition system. The performance of the classifier depends directly on the feature selection and extraction. The main advantage of feature extraction is that it removes redundancy from the data and represents the character image by a set of numerical features. These features are used by the classifier to classify the data.

In our implementation, moment invariants used by Hu [2] have been utilized to build the feature space. Using nonlinear combinations of geometric moments, we derived a set of invariant moments which has the desirable property of being invariant under image translation, scaling and rotation.

The central moments, which are invariant under any translation, are defined as

$$M_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\overline{x})^p (y-\overline{y})^q f(x,y)\,\mathrm{d}x\,\mathrm{d}y \tag{1.1}$$

where

$$\left.\begin{aligned} \overline{x} &= \frac{\overline{M}_{10}}{\overline{M}_{00}}, \overline{y} = \frac{\overline{M}_{01}}{\overline{M}_{00}} \quad \text{and} \\ \overline{M}_{pq} &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,\mathrm{d}x\mathrm{d}y \end{aligned}\right\} \tag{1.2}$$

However, for images, the continuous image intensity function $f(x, y)$ is replaced by a matrix, where x and y are the discrete locations of the image pixels. The integrals in Equations (1.1) and (1.2) are approximated by the summations:

$$M_{pq} = \sum_{x=0}^{m}\sum_{y=0}^{n} (x-\overline{x})^p (y-\overline{y})^q f(x,y)\,\mathrm{d}x\mathrm{d}y \tag{1.3}$$

$$\overline{M}_{pq} = \sum_{x=0}^{m}\sum_{y=0}^{n} x^p y^q f(x,y)\,\mathrm{d}x\mathrm{d}y \tag{1.4}$$

where m and n are dimensions of the image. The set of moment invariants that has been used by Hu are given by:

$$\phi_1 = M_{20} + M_{02} \tag{1.5a}$$

$$\phi_2 = (M_{20} - M_{02})^2 + 4M_{11}^2 \tag{1.5b}$$

$$\phi_3 = (M_{30} - 3M_{12})^2 + (3M_{21} - M_{03})^2 \tag{1.5c}$$

$$\phi_4 = (M_{30} + M_{12})^2 + (M_{21} + M_{03})^2 \tag{1.5d}$$

$$\begin{aligned} \phi_5 = {} & (M_{30} - 3M_{12})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2] \\ & + (3M_{12} - M_{03})(M_{21} + M_{03}) * [3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] \end{aligned} \tag{1.5e}$$

$$\begin{aligned} \phi_6 = {} & (M_{20} - M_{02})[(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] \\ & + 4M_{11}(M_{30} + M_{12})(M_{21} + M_{03}) \end{aligned} \tag{1.5f}$$

$$\begin{aligned} \phi_7 = {} & (3M_{21} - M_{03})(M_{30} + M_{12})[(M_{30} + M_{12})^2 - 3(M_{21} + M_{03})^2] \\ & + 3(M_{21} - M_{03})(M_{21} + M_{03}) * [3(M_{30} + M_{12})^2 - (M_{21} + M_{03})^2] \end{aligned} \tag{1.5g}$$

Table 1.2 Moment invariants represented by a 4×4 array.

M	0	1	2	3
0	X	X	√	√
1	X	√	√	X
2	√	√	X	X
3	√	X	X	X

Table 1.3 Moment invariant values for some characters.

	ة	ي	ف	ظ	ا
ϕ_1	−1.180	−1.422	−1.595	−1.536	0.375
ϕ_2	−4.027	−5.823	−4.790	−6.244	0.636
ϕ_3	−7.293	−9.088	−7.642	−8.935	−2.668
ϕ_4	−6.459	−8.621	−10.190	−13.024	−2.804
ϕ_5	−13.338	−17.739	−19.181	−24.368	−5.541
ϕ_6	−8.576	−11.578	−12.584	−16.325	−2.582
ϕ_7	−13.175	−17.404	−20.016	−23.981	−9.058

These functions can be normalized to make them invariant under a scale change by using the normalized central moments instead of the central moments. The normalized central moments are defined by

$$m_{pq} = \frac{M_{pq}}{M_{00}^{a}} \text{ where } a = \frac{(p+q)}{2} + 1 \tag{1.6}$$

These, when substituted into the above equations, will give seven moments which are invariant to translation, scale change and rotation.

The ϕs have large dynamic values. Thus, it was found that it was more practical to deal with the logarithms of the magnitudes of the ϕs [15] Thus, the seven moment invariants used in the proposed system are replaced by their logarithmic values.

In the final implementation to consider all the four possible positions of the characters, the four shapes of the letter are represented in the feature space. From Table 1.2 it can be noted that for seven moment invariants if a 4×4 matrix is constructed, only moments $M_{02}, M_{03}, M_{11}, M_{12}, M_{20}, M_{21}, M_{30}$ (upper diagonal elements) are used. However, if the number of moments is increased, other cells are also expected to be filled. For each character, the above moment invariant descriptors are calculated and fed to the artificial neural network. Table 1.3 shows the rounded values of ϕ obtained for some of the characters in the training set. Each of the characters is a 100×100 binary image.

6. Recognition Strategy

The process of recognizing a pattern is basically dealt with by three possible approaches: statistical, syntactic and neural approaches [30]. The statistical approach is based on decision making (the probabilistic model), the syntactic approach deals with the structural description of pattern (formal grammar) and the neural approach is based on training the system with a large input data set and

storing the weights (stable state), which are used later for recognition of trained patterns. A survey on character recognition shows that the syntactic and neural approaches have gained widespread attention compared to the decision theoretic approach. This is due to the simplicity and efficiency of the first two approaches compared to the latter. Not surprisingly, in our system we also discuss the recognition of Arabic characters using the syntactic and neural approaches.

6.1 Recognition Using the Syntactic Approach

In the structural approach, recognition is based on the stored structural pattern information. The information stored is the pixel value for the investigated character and the prototyped character. For the proposed system, the pattern of the character is stored as a matrix. Each cell stores the information related to each pixel of a particular character. The pixel value can be either 0 (indicating a black pixel), or 1 (indicating a white pixel). Recognition is done by matching the investigated character with the prototype character. The segmented character is passed through the following two stages to achieve recognition.

1. Normalization.
2. Recognition using template matching.

6.1.1 Normalization

In this phase, first the extracted characters are refined to fit the characters into a window without white spaces on all four sides. Figure 1.10(a) shows the template of the extracted character 'ح'. Now, each character image is normalized to a size of 30×30. Figure 1.10(b) shows the character 'ح' normalized to 30×30. The normalization is done using a window to view port transformation. This mapping is used to map every pixel of the original image to the corresponding pixel in the normalized image. The image obtained is invariant to scaling because the characters were refined (i.e. all the white spaces on the top, bottom, left and right were removed, so as to fit the character exactly into the window).

6.1.2 Template Matching using the Hamming Distance

Template matching for character recognition is straightforward and is reliable. This method is more tolerant to noise. In this approach, the templates are normalized to 30×30 pixels and stored in the

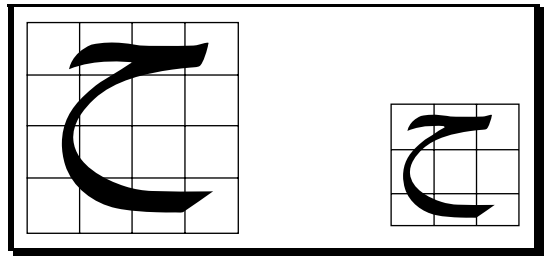


Figure 1.10 (a) Original template for character 'ح' (b) *normalized* 30×30 template for character 'ح'.

database. The extracted character, after normalization, is matched with all the characters in the database using a Hamming distance approach. This approach is shown in Equations (1.7) and (1.8).

$$\sum_{i=1}^{nrows} \sum_{j=1}^{ncols} \text{mismatch}_{i,j} \tag{1.7}$$

where

$$\text{mismatch}_{i,j} = \begin{cases} 1, & \text{if original}_{i,j} \neq \text{extracted}_{i,j} \\ 0, & \text{if original}_{i,j} = \text{extracted}_{i,j} \end{cases} \tag{1.8}$$

where *nrows* and *ncols* are the number of rows and columns in the original and extracted images. In our case, $nrows = ncols = 30$, as the image is normalized to 30×30. The mismatches for each of the extracted characters are found by comparison with the original characters in the database, and the character with the least mismatch value is taken as the recognized character. The database consists of a total of 33 classes of character. There are 149 characters in all the 33 classes of character. In addition to the usual 28 classes of character, oversegmented ones (such as س) are also included in the database.

6.2 Recognition Using the Neural Network Approach

Characters are classified according to their computed modified moment invariants by means of artificial neural networks. Among the many applications that have been proposed for neural networks, character recognition has been one of the most successful.

Many neural network architectures have been used in OCR implementation. MLP is usually a common choice. Unfortunately, as the number of inputs and outputs grow, the MLP grows quickly and its training becomes very expensive. In addition, it is not easy to come up with a suitable network design and many trial-and-error cycles are required. Radial Basis Function (RBF) networks, on the other hand, offer better features for recognition applications. In addition to highly efficient training, no design details are required since the network automatically adjusts its single hidden layer.

However, before implementing the neural network, numerical features of the character are extracted and the system is trained to associate the segmented character with the desired character. In our implementation, we used the invariant moment technique by Hu [2] to build the feature space. This measure is invariant with respect to size, translation and scaling.

Characters are classified according to their computed moment invariants by means of artificial neural networks. In our method, we used a Radial Basis Function (RBF) network with a back-propagation error learning algorithm. In implementing the RBF network architecture, the Brain Construction Kit (BCK) has been a very helpful tool. BCK is a Java package developed in the public domain that enables users to create, train, modify and use Artificial Neural Networks (ANNs).

RBF networks have been successfully applied to solve some difficult problems by training them in a supervised manner with an error back-propagation algorithm [31]. This algorithm is based on the error correction learning rule. Basically, error back-propagation learning consists of two passes through the different layers of the network: a forward pass and a backward pass [31]. In the forward pass, an input vector is applied to the nodes in the input layer and its effect propagated through the network layer by layer. Finally, a set of output is produced as the actual response of the network. During the forward pass, the weights of the network are all fixed. During the backward pass, the weights of the network are all adjusted in accordance with the error correction rule, i.e. the actual response of the network is subtracted from the desired response to produce an error signal. This error is then back propagated through the network and the weights are adjusted to make the actual response of the network move closer to the desired response. Figure 1.11 shows the block diagram of the implemented RBF network.

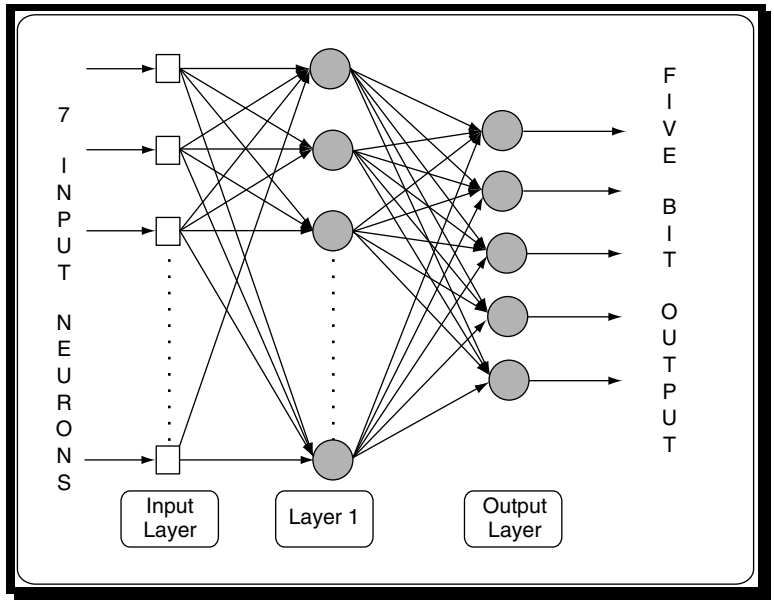


Figure 1.11 Three-layered radial basis function network.

In an RBF network there is an important difference between a normal synapse and a BCKNetConnection. During a forward pass through a network, in which an input vector is fed into the input layer and each non-input neuron evaluates a response to the supplied input, synapses are used to transfer the output of one neuron to the input of another. BCKNetConnections cause the output of the source neuron to provide the output of the target neuron, that is, during a forward pass through cascaded networks, the output vector of the source network is used as the input vector for the target network, not as inputs to the neurons in the target network input layer. The activation is calculated as the Euclidean distance of the weight vector from the input vector, output is calculated using a Gaussian transfer function. This neuron contains an extra parameter, the standard deviation value, for use in the transfer function.

The RBF network consists of three layers, an input layer of dummy neurons, a hidden layer of radial neurons and an output layer of linear neurons. Unlike the other architectures, which contain a fixed number of nodes, the RBF architecture is dynamic, in that it adds neurons to its hidden layer as it is trained. There are a total of 20 nodes in the hidden layer in the implemented systems, determined heuristically. The input layer is composed of seven neurons. These seven input neurons are the seven moment invariant features extracted from the feature extraction phase. Based on the input features that are passed to the neural network, the output layer gives the output character belonging to a particular class. The number of the output depends on the number of the characters in the character set. The output layer consists of five neurons, each neuron represented by a bit '1' or '0'. For example, for the character 'ت', the output layer gives the output '00011'. This output is then converted to a decimal value and this decimal value represents a particular character of that class. For example, '00011' when converted to a decimal value gives a value 3, which is the index for the class of character 'ت'.

The learning time is reduced by partitioning the classes of characters into four sets, one for each character form (Arabic characters have four forms: beginning, middle, end and isolated). The training set is composed of a total of 33 classes of character. There are 149 characters in all the 33 classes of character. In addition to the usual 28 classes of character, oversegmented ones (such as س) are also

included in the training set. The training document that is passed to the neural network is a 100×100 character image.

7. Experimental Results and Analysis

This section presents the implementation details and the results obtained using the proposed system.

7.1 System Training

In the syntactic approach, the training set is sampled data. The Hamming distance of the investigated character is matched with each character present in the sample data. The sampled data set consists of a total of 149 characters. To make the reference prototypes cover a wide range of possible variations without causing misrecognition, the sample data may be increased as new character samples are encountered. In the neural approach, the training of the system is based on the features extracted using the moment invariant technique. First, the RBF network computes an output for a given input. Next, the training of the system takes place using a weight adjustment strategy until the actual output converges with the desired output.

7.2 Experimental Set-up

Experiments have been performed to test the above system. The developed Arabic text recognition system has been tested using randomly selected text. The system is designed in JDK 1.4.1 for the recognition of a popular font called Naskh font. The system developed is a single font, multisize system. The image to be tested is captured using a scanner and is passed to the system as a bitmap file. The system has been tested with many different sizes of Naskh font. The experiments for the above system were implemented under different situations, i.e. the experiments were carried out for both document images in normal shapes and images that are skewed with some angle of slant.

As stated earlier, classification and recognition of characters is done using both the structural approach and the neural approach. In the structural approach, after the text is segmented into characters, each segmented character is normalized to a 30×30 image. The normalized character is matched pixel by pixel with the characters present in the training set using the Hamming distance approach. The character with the least distance is considered to be the output character. Even though this technique is costly in terms of time, the accuracy of the system is greatly improved.

In the neural approach, first the features of the character are extracted and the system is trained for the extracted features. The RBF network adopted consists of three layers, an input layer of dummy neurons, a hidden layer of radial neurons and an output layer of linear neurons. The input to the system is the seven moment invariant features. The single hidden layer consists of 20 nodes. Based on the input features that are passed to the neural network, the output layer gives the output character belonging to a particular class. The training set is composed of a total of 33 classes of character. There are 149 characters in all the 33 classes of character. The training document that is passed to the neural network is a 30×30 character image.

7.3 Results Achieved

The implemented system shows a recognition rate of 89 %–94 % using the structural approach, and an 83 % recognition rate using the neural network approach. When recognition is performed on small words or text with isolated characters ‘الحمد,يؤب , يؤم’, the recognition rate achieved using the structural approach is 98–100 %, and the recognition rate achieved using neural networks is 88 %.

The experimental results show that the shortcoming of the system using neural networks is mainly due to the closeness among the features of the characters. The features obtained using the moment invariant technique are not unique themselves, but closely related to the features of other characters. This prevents the neural network from clear classification. However, one possible solution is to improve the number of moments for better recognition. Dabi *et al.* [18] have shown that if the number of moments used in the feature extraction technique is increased to 11, the recognition rate can be improved considerably.

Figure 1.12(a) shows a test document used for verification of the system accuracy. The recognition rate achieved for this image using the syntactic approach was 94 %, and using the neural approach was 83 %. Most of the failures in the recognition of the text using the syntactic approach are due to improper segmentation of the text into individual characters. To improve the accuracy using the syntactic approach, one solution is to increase the number of classes in the training set. The test

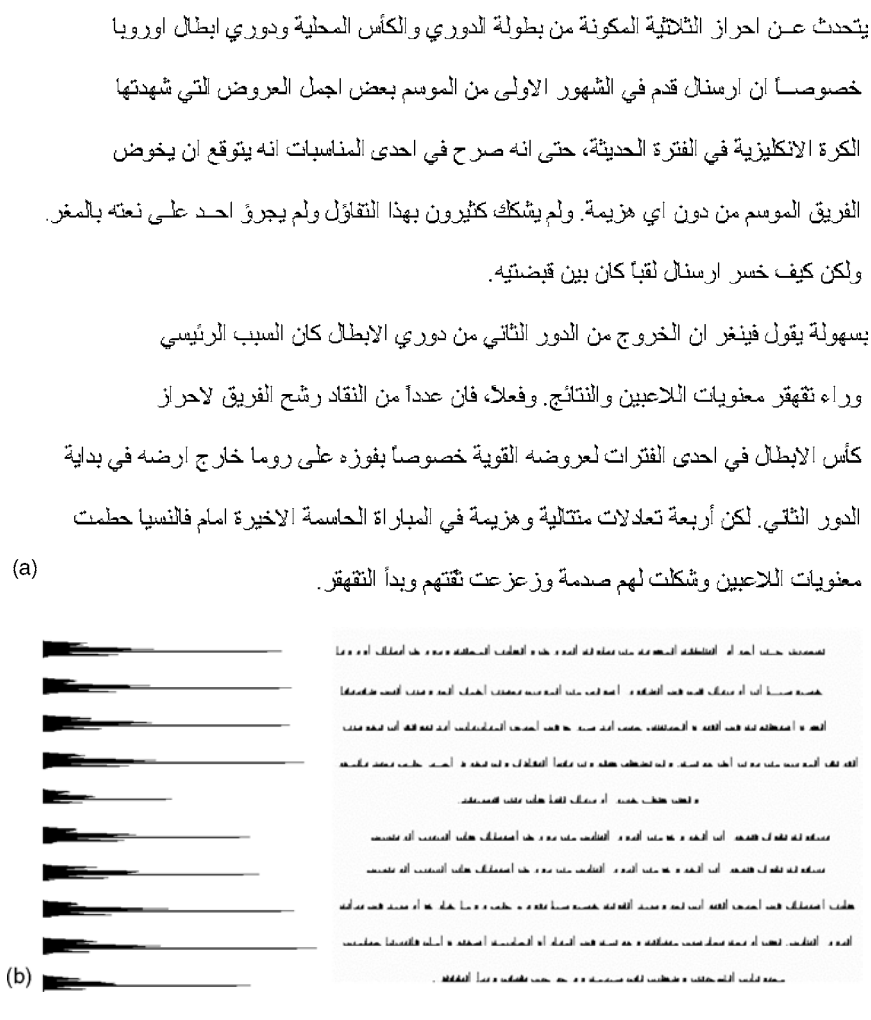


Figure 1.12 (a) Image used to test the system after preprocessing; (b) horizontal and vertical projections of the test document image.

document consisted of approximately 580 characters. Figure 1.12(a) shows an image that has been used to test the system. The original image was skewed by 18 degrees with respect to the horizontal. Figure 1.12(a) shows the resultant image when the image is passed through the preprocessing stage of the proposed system. The resultant image is inclined to zero degrees with respect to the horizontal. This shows that the proposed system is highly invariant to the rotation transformation. Figure 1.12(b) shows the horizontal and vertical projections of this test document image.

8. Conclusion

A method for offline recognition of Arabic text has been presented. Two approaches have been investigated: the syntactic approach and the neural network approach. In the preprocessing stage, we solved the problem of skew when scanning the documents. Even though the system developed is being tested for Naskh font only, it provides an ample scope for research towards the development of a system for multifont recognition. Extending the system to other most used fonts by extending the existing technique can be of great interest. We have shown the algorithms in detail for both recognition approaches and have suggested ways of improving the system. On the general improvement of the recognition rate using neural networks, an increase in the number of features extracted from seven to nine, or some higher number, may help in increasing the recognition rate. Other improvements on increasing the accuracy of the system include: computing the variance between the different momentum values obtained for a character; and trying to increase the variance among the momentum values. This will be important, especially for multifont recognition. The system is implemented in Java and is still in progress.

Acknowledgment

The authors also acknowledge the support of King Fahd University of Petroleum and Minerals for funding this work under the Project No. EE/AUTOTEXT/232.

References

[1] Amin, A. "Off-Line Arabic Character Recognition system: State of the Art," *Pattern Recognition*, **31**(5), pp. 517–530, 1998.

[2] Hu, M. K. "Visual Pattern Recognition by Moment Invariant," *IRE Transactions on Information Theory*, **IT – 8**, pp. 179–187, 1962.

[3] Amin A. and Al-Sadoun, H. B. "A new Segmentation Technique of Arabic Text," *11th IAPR*, The Hague **2**, pp. 441–445, 1992.

[4] Almuallim, H. and Yamaguchi, S. "A method of recognition of Arabic Cursive Handwriting," *IEEE Transaction Pattern Analysis and Machine Intelligence*, PAMI – 9, pp. 715–722, 1987.

[5] Cheung, A., Bennamoun, M. and Bergmann N. W. "An Arabic optical character recognition system using recognition-based segmentation," *Pattern Recognition*, **34**, pp. 215–233, 2001.

[6] Zidouri, A. "A Structural Description of Binary Document Images: Application for Arabic Character Recognition," *Proceedings of International Conference On Image Science, Systems, and Technology*, Las Vegas, **I**, pp. 458–464, June 25–28, 2001.

[7] Margner, V. "SARAT: A system for the Recognition of Arabic Printed Text," *Proceedings of the 11th International Conference on Pattern Recognition*, pp. 561–564, 1992.

[8] Hamami, H. and Berkani, D. "Recognition System for Printed Multi-Font and Multi-Size Arabic Characters," *Arabian Journal for Science and Engineering*, **27**(1B), pp. 57–72, 2002.

[9] Nazim Nawaz, S., Sarfraz, M., Zidouri, A. B. C. and Al-Khatib, W. "An Approach to Offline Arabic Character Recognition using Neural Networks," *10th International Conference on Electronics, Circuits and Systems, ICECS 2003*, Sharjah, UAE, 2003.

[10] Sarfraz, M., Nazim Nawaz, S. and Al-Khoraidly, A. "Offline Arabic Text Recognition System," *International Conference on Geometric Modelling and Graphics, GMAG 2003*, London, England, 2003.

[11] Zidouri, A. B. C., Sarfraz, M., Nazim Nawaz, S. and Ahmed, M. J. "PC Based Offline Character Recognition System," *Seventh International Symposium on Signal Processing and its Applications, ISSPA 2003*, Paris, France, 2003.

[12] Trier, O. D., Jain, A. K. and Taxt, T. "Feature Extraction methods for character recognition: A Survey," *Pattern Recognition*, **29**(4), pp. 641–662, 1996.

[13] Tubbs, J. D. "A Note on Binary Template Matching," *Pattern Recognition*, **22**(4), pp. 359–365, 1989.

[14] Mori, S. Suen, C. Y. and Yamamoto, K. "Historical review of OCR research and development," *Proceedings of the IEEE*, **80**, pp. 1029–1058, 1992.

[15] Abdul Wahab, O. A. "Application of Artificial Neural Networks to Optical Character Recognition," Thesis Dissertation, King Fahd University of Petroleum and Minerals, Dhahran, K. S. A, June 1994.

[16] Altuwaijri, M. and Bayoumi, M. "Arabic text recognition using Neural Networks," *Proceedings International Symposium on Circuits and Systems – ISCAS'94*, pp. 415–418, 1994.

[17] Boyce, J. F. and Hossack, W. J. "Moment Invariants for pattern recognition," *Pattern Recognition Letters*, **1**, pp. 451–456, 1983.

[18] El-Dabi, S. S., Ramsis, R. and Aladin Kamel, R. "Arabic Character Recognition System: A Statistical approach of recognizing cursive typewritten text," *Pattern Recognition*, **23**(5), pp. 485–495, 1990.

[19] Fakir, M. and Hassani, M. M. "Automatic Arabic Character recognition by moment invariants," *Colloque international de telecommunications*, Fes, Morocco, pp. 100–103, 1997.

[20] Fakir, M., Hassani, M. M. and Sodeyama, C. "Recognition of Arabic Characters using Karhunen–Loeve Transform and Dynamic Programming," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, **6**, pp. 868–873, 1999.

[21] Sadallah, S. and Yacu, S. "Design of an Arabic Character Reading Machine," *Proceedings of Computer Processing of Arabic Language*, Kuwait, 1985.

[22] Jambi, K. "Arabic Character Recognition: Many approaches and one Decade," *Arabian Journal for Science and Engineering*, **16**, pp. 499–509, 1991.

[23] Chinveerphan, S. and Zidouri, A. B. C. "Modified MCR Expression using Binary Document Images," *IEICE Transactions Information & Systems*, **E78 –D**(4), pp. 503–507, 1995.

[24] Zidouri, A. B. C. *Arabic Document Image Analysis and Recognition Based on Minimum Covering Run*, Thesis Dissertation, Tokyo Institute of Technology, Japan, 1995.

[25] Zidouri, A. B. C., Chinveerphan, S. and Sato, M. "Classification of Document Image Blocks using Stroke Index," *IEICE Transactions Information & Systems*, **E78 –D**(3), pp. 290–503, 1995.

[26] Al–Sadoun, H. B. and Amin, A. "A new structural technique for recognizing printed Arabic Text," *International Journal on Pattern Recognition and Artificial Intelligence*., **9**, pp. 101–125, 1995.

[27] Smagt, P. P. V. "A Comparative Study of Neural Network Algorithms Applied to Optical Character Recognition," *IEA/AIE*, **2**, pp. 1037–1044, 1990.

[28] Duin, R. P. W. "Superlearning and Neural Network Magic," *Pattern Recognition Letters*, **15**, pp. 215–217, 1994.

[29] Jain, A. K. and Mao, J. "Neural Networks and pattern recognition," In *Proceedings of the IEEE World congress on Computational Intelligence*, Orlando, Florida, 1994.

[30] Schalkoff, R. J. *Pattern Recognition: Statistical, Structural and Neural Approaches*, John Wiley and Sons Inclusive.

[31] Haykin, S. *Neural Networks: A Comprehensive Foundation*, second edition, Prentice Hall, 1999.