A Proposed OCR Algorithm for the Recognition of Handwritten Arabic Characters

Ahmed T. Sahlol[*1], Cheng Y. Suen[2], Mohammed R. Elbasyouni[3], Abdelhay A. Sallam[4]

[1, 3]Department of Computer Teacher Preparation, Damietta University, Damietta, Egypt
[2]Department of Computer Science, Concordia University, Montreal, Quebec, Canada
[4]Department of Electrical Engineering, Port-Said University, Port Said, Egypt

[*1]asahlol@encs.concordia.ca; [2]parmidir@encs.concordia.ca; [3]mmrefaat@hotmail.com; [4]aasallam@ucalgary.ca

***Abstract*-Recognition of handwritten Arabic text awaits accurate recognition solutions. There are many difficulties facing a good handwritten Arabic recognition system such as unlimited variation in human handwriting, similarities of distinct character shapes, character overlaps, and interconnections of neighbouring characters and their position in the word. Arabic characters are drawn in four forms: Isolated, Initial, Medial, and Final. The typical Optical Character Recognition (OCR) systems are based mainly on three stages, pre-processing, features extraction and recognition. Each stage has its own problems and effects on the system efficiency which may be time consuming, resource using and may contribute to the possibility of recognition errors.**

There are many feature extraction methods for handwritten letters. In this paper, an efficient approach for the recognition of off-line Arabic handwritten characters is presented. The approach is based on novel preprocessing operations (including different kinds of noise removal and dilation), structural, statistical and topological features from the main body of the character and also from the secondary components. Evaluation of the importance and accuracy of the selected features is made.

An off-line recognition system based on the selected features was built. The system was trained and tested with CENPRMI dataset. We used the popular Feed Forward Neural Network for classification to enhance recognition accuracy. The proposed algorithm obtained has promising results in terms of accuracy (success rate of 100% for some letters with an average rate of 88%). Compared to other related works, we find that our success outcomes are higher.

Keywords- Handwritten Arabic Characters; Noise Removal; Secondary Components

I. INTRODUCTION

Arabic handwriting recognition systems enable the automatic reading or searching of historic Arabic manuscripts. The estimated number of these manuscripts exceed three million [1]. Now we are able to translate images of typewritten or handwritten text into machine-editable text encoded in a standard encoding scheme (e.g. ASCII or Unicode). Recognition of handwritten cursive text such as Arabic text is a current research problem [2-3].

Currently OCR systems have expanded to recognize Latin alphabets, Japanese Katakana syllabic characters, Kanji (Japanese version of Chinese) characters, Chinese characters, Hangul characters, etc.

Work on Arabic OCR started in the 1970s [4]. The first published work on Arabic OCR dates back to 1975 [5]. The first Arabic OCR system was made available in 1990s [6]. The recognition of Arabic handwriting presents some unique challenges and benefits to researchers [7]. Although more than three decades have passed, there has been a lack of effort in the recognition of Arabic handwritten texts compared to the recognition of texts in other scripts [4, 6].

The main problem encountered when dealing with handwritten Arabic characters is that characters have unlimited variation in human handwriting styles, similarities of distinct character shapes, character overlaps, and interconnections of neighboring characters.

Users are still waiting for reliable and accurate solutions for recognizing handwritten cursive scripts such as Arabic texts [8].

A. Arabic Writing System

Arabic is written from right to left and is always cursive. It has 28 basic characters. Thus, roughly the alphabet set can expand to 84 different shapes according to the position of the letter (beginning, middle, end or isolated) as well as according to the style of writing (Nasekh, Roqa'a, Farsi and others).

A character is drawn in an isolated form when it is written alone and is drawn in three other forms when it is connected to other characters in the word. For example, the character Ain has four forms: isolated, initial, medial and final. See Fig. 1.

a b c d

Fig. 1 Different forms of Ain character, (a) isolated; (b) initial; (c) medial; (d) final

The secondary components are character components that are disconnected from the main body. Sixteen Arabic letters have from one to three secondary components (dots).

Additionally, some characters like Alif (أ) and Kaaf (ك) can have a zigzag-like stroke called Hamza (ء). Dots and Hamzas are called secondaries and they are located above the primary part of the character as in Alif (أ), or below like Baa (ب), or in the middle like Jeem (ج).

The type and position of the secondary components are very important features of Arabic letters.

In Table 1 we see letters can be distinguished only by their secondary components. For example, Tah (ط) A1 and Thah (ظ) A2 differ only by the number of dots above the main body, also Seen (س) B1 and Sheen (ش) B2.

Another important kind of variation in drawing the secondary components appears in drawing two or three dots. As shown in Table 1 Sheen (ش) B2, B3 and B4, the three dots come in three variations: isolated, connected and linked to the character, respectively. Also Taa (ت) C1, C2 can be drawn in two variations: two isolated dots or one short horizontal dashed line.

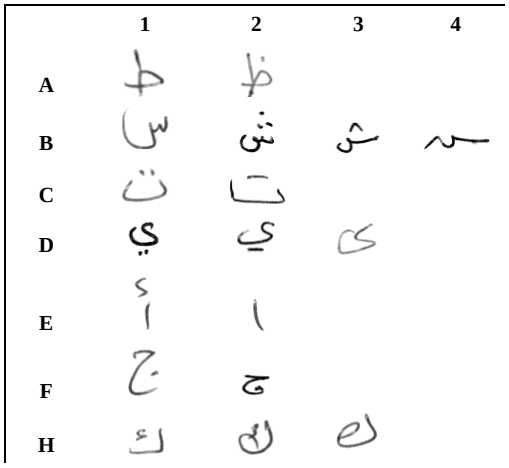
Another kind of difference between characters depends only on the position of the two dots; see Taa (ت) C1, Yaa (ي) D1 and Taa (ت) C2, Yaa (ي) D2 respectively.

There is also another classification challenge; some characters which contain secondary components can also be written without those secondary components if written in "Roqa'a" writing style. For example Yaa (ي) can be drawn with two isolated dots D1 or with short dashed line D2 or without any dots D3. Another example is Alif (أ) which can be drawn with hamza E1(default) or without it E2.

Another recognition difficulty is due to some writers' styles which can join the secondary components of isolated and final forms with main body curves. Table 1 shows some examples: Samples F1 and F2 show how the one dot of isolated Geem (ج) is joined to the main character body.

Another difficulty in recognizing the secondary components is due to the writer writing quickly, drawing them connected to the main body. For example, Samples H2, H3 show the Hamza connected to Kaf's (ك) body. Any secondary components classification process should take all the above variations into consideration [9].

TABLE 1 VARIATIONS IN DRAWING THE SECONDARY COMPONENTS



Considerable work has been undertaken in the area of Arabic character recognition, targeted in various ways to improve accuracy and efficiency but with limited success. This is due to the nature of Arabic characters and to the problems mentioned above.

El-Dabi et al. [10] presented a recognition system for typed Arabic text, which involves a statistical approach for character recognition. El-Sheikh et al [11] proposed algorithms to recognize Arabic handwritten characters; this system assumes that characters result from a reliable segmentation stage, thus, the position of the character is known a priori. Four different sets of character shapes have been independently considered (initial, medial, final, and isolated). Each set is further divided into four subsets depending on the number of strokes in the character. El-Khaly *et al*. [12] discussed an algorithm for the machine recognition of optically captured Arabic characters and their isolation from the printed text. Moment-invariant descriptors are investigated for the purpose of recognition of individual characters. Sabri [13] has used Fourier and contour analysis for the recognition of Arabic characters with acceptable recognition rates. The features of an input character are compared to the models' features using a distance measure. The model with the minimum distance is taken as the class representing the character.

If we look at a character as an image from which we can extract much useful information, such information can be structural features such as loops, branch-points, endpoints, and dots or statistical which include pixel densities, histograms of

chain code directions, moments, and Fourier descriptors. Many approaches and techniques have been proposed [14-18] using loops, dots, curves, relative locations, height, sizes of parts of characters, loop positions and types, line positions and directions and turning points. Others like Abuhaiba, Al-Yousefi and Dehghan [19-21] used statistics from moments of horizontal and vertical projection. Histogram of slopes along contour is used by Abdelazeem [22].

Artificial Neural Networks (ANNs) are the common element in most, if not all, classifiers recognition. In this paper we use neural networks as a classifier like Abandah [23] who proposed a system for the recognition of handwritten Arabic characters. Also Sherif and Mostafa [24-25] presented a parallel design for a back propagation neural networks approach in order to accelerate the computation process. But another kind of neural network called Learning Vector Quantization (LVQ) was used in [26] for handwritten Arabic character recognition. Recurrent neural networks which is another kind of neural network was used in [27] for learning features directly from raw word images. Bluche and Ney [28] made a combination of a convolutional neural network with a HMM that gave better results compared with recurrent neural networks, instead of using only HMM as in [29]. Another combination between neural networks and HMM was done in [30].

While others [31-32] used SVM (support vector machine) as a classifier for Arabic numerals and for texts [33]. A more recent survey on Arabic handwritten text recognition was presented in [34].

In this paper, we propose a novel approach for preprocessing procedures including different kinds of noise removal and dilation, then extracting statistical, morphological and topological features of handwritten Arabic characters. We apply this technique in extracting moment features and show that this technique provides better feature sets that give higher recognition accuracies. The block diagram of the proposed system is shown in Fig. 2.

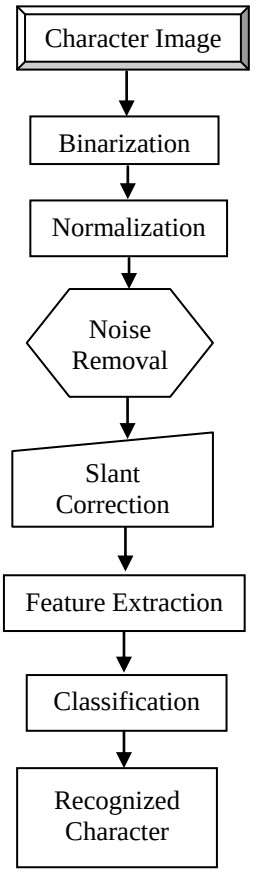


Fig. 2 Block diagram of the proposed system

This paper is organized into three sections. Section II describes the methodology of our work including Binarization, normalization, noise removal algorithms, feature extraction techniques, classification stage including its architecture, training and testing phases. Section III provides experimental results including classification accuracy, comparison between our work and others and our contributions. Finally, Section IV describes the main conclusions and future work.

II. METHOD

A. Binarization

Our purpose in this step is to convert the input image to a binary image based on threshold. Binary images may contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise and texture.

Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image.

Our purpose in this step is to convert the input image to a binary image based on threshold. Binary images may contain numerous imperfections. In particular, the binary regions produced by simple thresholding are distorted by noise and texture. Morphological image processing pursues the goals of removing these imperfections by accounting for the form and structure of the image.

Replace all pixels in the input image with luminance greater than level with the value 1 (white) and replace all other pixels with the value 0 (black).

Compute a global threshold (level) that can be used to convert an intensity image to a binary image with a normalized intensity value that lies in the range 0, 1.

We use Otsu's method [35], which chooses the threshold to minimize the intraclass variance of the black and white pixels. Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels on each side of the threshold, i.e. the pixels that either fall in the foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum. The algorithm assumes that the image to be thresholded contains two classes of pixels (e.g. foreground and background), then calculates the optimum threshold separating those two classes so that their combined spread (intra-class variance) is minimal.

We use Otsu method not only because it is a global binarization technique but also because of its short running time; the mean running time for the Otsu's binarization method was 2.0 secs and this is one of the lowest running times (Original Sauvola algorithm takes 12.6 secs [36]).

Slant Correction: we make slant correction for every character image to eliminate any slant in each character. The basic idea is to locate near-vertical strokes in the character and to estimate the average slant of the character from these strokes. Then, the slant in a character is corrected by applying a shear transformation to the character.

B. Normalization

Normalization is to regulate the size, position, and shape of character images, so as to reduce the shape variation between the images of same class. Denote the input image and the normalized image by $f(x, y)$ and g $(\bar{x},\bar{y})$, respectively, normalization is implemented by coordinate mapping.

$$\begin{cases} x' = x'(x,y), \\ y' = y'(x,y). \end{cases} \tag{1}$$

We denote the width and height of the original character by $W1$ and $H1$, the width and height of the normalized character by $W2$ and $H2$, and the size of the standard (normalized) plane by L as seen in Fig. 3. The standard plane is usually considered as a square and its size is typically 32 × 32 or 64 × 64, among others. We define the aspect ratios of the original character ($R1$) and the normalized character ($R2$) as

$$R_1 = \frac{\min(W_1, H_1)}{\max(W_1, H_1)}$$

and

$$R_2 = \frac{\min(W_2, H_2)}{\max(W_2, H_2)} \tag{2}$$

which is always considered in the range of [0, 1].

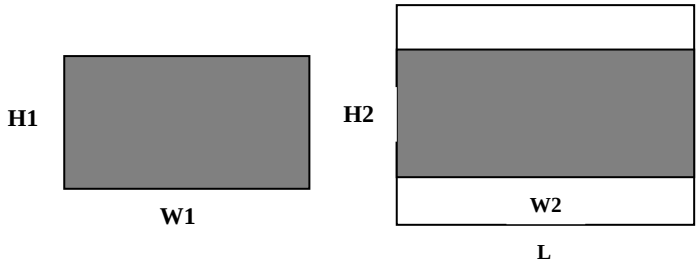


Fig. 3 Normalization method, (a) Original character; (b) normalized character filled in standard plane

We use the Linear Backward mapping method [37] where:

$$x = x'/\alpha$$
$$y = y'/\beta \quad (3)$$

Where α and β denote the ratios of scaling, given by:

$$\alpha = W_2/W_1$$
$$\beta = H_2/H_1 \quad (4)$$

where $W1$ and $H1$ are the horizontal span and vertical span of the strokes of the original character (size of minimum bounding box).

C. Noise Removal

1) Statistical Noise removal:

In addition to enhancement of character image by contrast and dynamic range modification, a character image can also be enhanced by reducing degradations that may be present. This area of image enhancement overlaps with image restoration.

Median filtering [38] is a nonlinear process useful in preserving edges in an image while reducing random noise. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value. (If the neighborhood under consideration contains an even number of pixels, the average of the two middle pixel values is used). Fig. 4 shows the process of Median filtering.

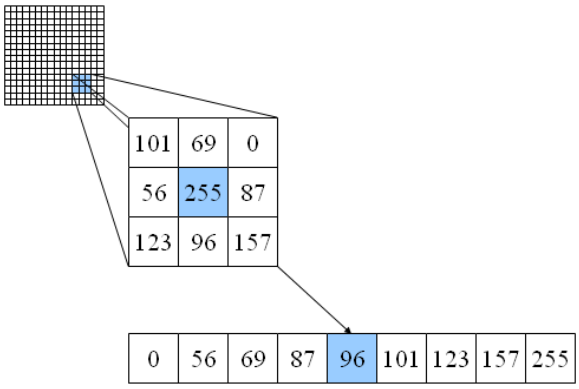


Fig. 4 Median filtering process

A template of size 3x3, 5x5, 7x7,… etc is applied to each pixel. The values within this template are sorted and the middle of the sorted list is used to replace the template central pixel. Median filtering can preserve discontinuities in a step function and can smooth a few pixels whose values differ significantly from their surroundings without affecting the other pixels [38].

In this paper, several filters were tested, however a 3 ×3 median filter is chosen because it gives us the best result. The median filter is used to reduce noise in an image by considering each pixel within its neighboring pixels to decide whether or not it is representative of its surroundings. Then, it replaces the pixel value with the median of the values of the neighboring pixels.

2) Morphological noise removal:

Filling: fill isolated interior pixels (individual 0s that are surrounded by 1s) such as the centre pixel in Fig. 5. For each pixel p in the binary image I, check the two and decide whether P is to be 0 or 1 if B(p) =4, where B(p) is the number of non-zero neighbors of p.

1	1	1		1	1	1
1	p	1	becomes	1	1	1
1	0	1		1	1	1

Fig. 5 Filling Process

Bridging: bridge unconnected pixels, that is, sets 0-valued pixels to 1 if they have two nonzero neighbors that are not connected. For each pixel p in the binary image I, check the two neighbors (as shown in the below figure) and decide whether **P** to be 0 or 1 if B (p1)>=2, where B (p) is the number of non-zero neighbors of p.

1	0	0		1	1	0
1	p	1	becomes	1	1	1
0	0	1		0	1	1

Fig. 6 Bridging Process

Removing: remove isolated pixels (individual 1s that are surrounded by 0s), such as the centre pixel in this pattern. For each pixel p in the binary image I, check the two neighbors (as shown in the below figure) and decide whether P is to be 0 or 1, if B(p) =0, where B(p) is the number of non-zero neighbors of p.

0	0	0		0	0	0
0	p	0	becomes	0	0	0
0	0	0		0	0	0

Fig. 7 Removing process

Another kind of morphological operation used in this paper is Dilation:

Dilation: an operation that grows or thickens objects in a binary image. The specific manner and extent of this thickening is controlled by a shape referred to as a structuring element. This morphological technique exposes an image to a small shape or template; the structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood. Fig. 8 shows how dilation works.

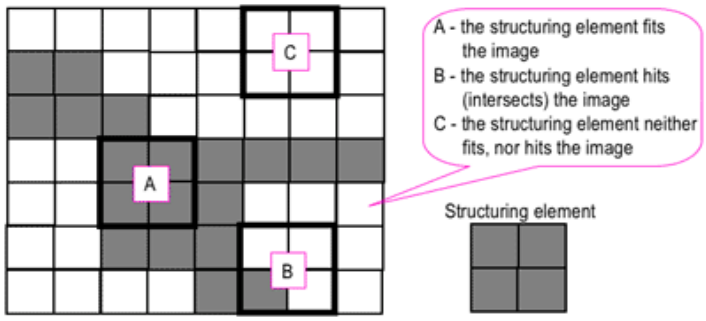


Fig. 8 Exposing an image to a structuring element (white and grey pixels have zero and non-zero values, respectively)

A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

The dilation of *A* by *B* is defined by:

$$A \oplus B = \bigcup_{b \in B} A_b \tag{5}$$

The dilation is commutative, and can also given by:

$$A \oplus B = B \oplus A = \bigcup_{a \in A} B_a \tag{6}$$

If *B* has a centre on the origin, then the dilation of *A* by *B* can be understood as the locus of the points covered by *B* when the centre of *B* moves inside *A*.

The dilation can also be obtained by:

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \emptyset\} \tag{7}$$

where ∅ is the empty set and B is the structuring element and B^s denotes the symmetry of B, that is:

$$B^s = \{x \in E | -x \in B\} \tag{8}$$

In this work we create a square of 2x2 of ones as a structuring element. For an example of bad and good dilations see Fig. 9

a b c d

Fig. 9 Dilation for Yaa character, (a) Original image; (b) primary Preprocessed; (c) badly Dilated; (d) perfectly Dilated

D. Feature Extraction

1) Structural Features:

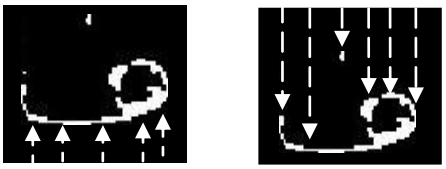
Upper and Lower profile*:* The upper and lower profiles capture the outlining shape of a connected part of the character.

Upper (or lower) connected part profile is computed by measuring the distance (pixel count) of each column group from the top (or bottom) of the bounding box to the connected part of the closest ink pixel in that column group.

1. Read the image into a two-dimensional array.

2. Divide the width into g column groups.

3. For each column group: **a**. compute the distance from the top (or bottom) of the bounding box of the connected part to the closest ink pixel in that group by counting the number of white pixels. **b**. get the ratio of distance to the number of black pixels of each group. Fig. 10 shows the upper and lower profiles of the character "Faa".



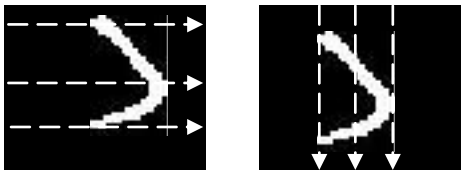
a b

Fig. 10 (a) Upper profile; (b) Lower profile for character "Faa"

Horizontal and Vertical projection profiles: projection profile based feature extraction method delivers excellent results even in the absence of some important preprocessing steps such as smoothing and thinning. In fact, in this type of feature extraction it will be disadvantageous to apply the thinning process because there will be a huge loss of important information related to the count and position of white pixels present in the character image.

Vertical profile is the sum of white pixels perpendicular to the y axis. It is computed by scanning the character column along the y-axis and counting the number of white pixels in each column.

Similarly, the horizontal projection profile is the sum of black pixels but it is perpendicular to the x axis. The character is traced horizontally along the x-axis using the sum of number of white pixels present in each row. See Fig. 11.



a b

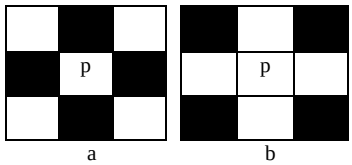
Fig. 11 Character Daal projection profile, (a) horizontally; (b) vertically

2) Statistical Features:

Connected components*: w*e consider the important "middle ground" between the individual foreground pixels and the set of all foreground pixels. This leads to the notion of connected components, also referred to as the following objects:

A pixel p at coordinates (x, y) has two horizontal and two vertical neighbours whose coordinates are (x+1, y), (x-1, y), (x, y+1) and(x, y-1).

This set of 4 neighbors of p denoted N4 (p) is illustrated in Fig. 12, where the four diagonal neighbors of p have coordinates (x+1, y+1), (x+1, y-1), (x-1, y+1) and (x-1, y-1) respectively.



a b

Fig. 12 (a) Pixel p and its 4 neighbours; (b) Pixel p and its diagonal neighbours

Connected components are an important feature because most Arabic characters contain one or more connected components like (أ, ب, ت, ث, **خ**, ذ, ض, **غ**, ق, ك, ي), so this feature can be effective for distinguishing among those characters which contain connected components and others which don't contain connected components.

3) Topological features:

End points: another feature that is useful is the number of end points in the character. Those points have only one neighbour and the other three neighbours are noise.

Input: binary preprocessed character image

Output: feature vector, F, representing the most common end point of the character.

Steps:

1. Read the character image, preprocess it (Binarization, normalization, ...)
2. Dilate the image to join all its parts. Dilation essentially just adds pixels around existing white pixels.
3. Shrink the image to points (removes pixels so that objects without holes shrink to a point, and objects with holes shrink to a connected ring halfway between each hole and the outer boundary).
4. Find the pixel that is still white, which I is the row and j is the column of the white pixel in image such that image (I,j) is equal to 1. All the other pixels should be 0 in the image.
5. Gather I and j in one vector "G".
6- Normalize each of the vector elements using Min–max method [39].
7- Determine the most repeated values by calculating differences between adjacent elements.
8- Finally, determine the most frequent values in the character array.
9- Put all the end points for the same character in a feature vector.

Algorithm 1. End points of a character image

Notes about the algorithm:

-At the 2nd stage "Dilation". We did it twice because we used Otsu method in binarization.

-At the 3rd stage "Shrinking". We shrank the character image until we reached the fewest number of white pixels.

The figure below describes all the stages for the above algorithm.

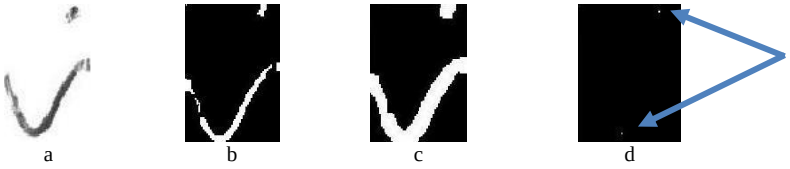


Fig. 13 End point stages for Zaay character, (a) original; (b) preprocessed; (c) dilated; (d) end points

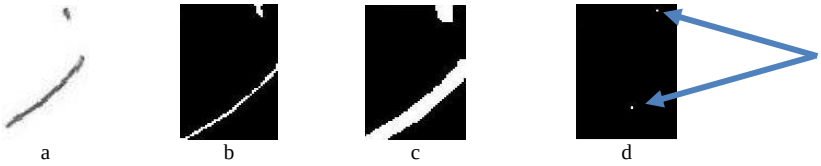


Fig. 14 End point stages for another Zaay character, (a) original; (b) preprocessed; (c) dilated; (d) end points

It is obvious that the outer boundary points (end points) of the two Zaay characters are very close although they do not have the same shape. When we check the coordinates (step 4) of the end points of the two characters we find that the two characters have exactly the same coordinates of one from the two end points.

***Pixel ratio*:** The character area is the total number of white (foreground) and black pixels (background) of the character. Pixel ratio is: (the number of white pixels / the number of black pixels).

***Height to width ratio*:** since different people write the same characters in different sizes, the absolute width and height are not reliable features for Arabic handwritten characters. However, some Arabic characters are wider than others. Therefore, the aspect ratio (height/width ratio) of the character is a useful feature.

Feature normalization: The simplest normalization technique is the Min–max normalization method [39]. Min–max normalization is best suited for the case where the bounds (maximum and minimum values) of the scores produced by a match are known. In this case, we can easily shift the minimum and maximum scores to 0 and 1, respectively. However, even if the matching scores are not bounded, we can estimate the minimum and maximum values for a set of matching scores and then apply the min–max normalization.

We use this method because it has one of the best recognition performances. Also it is efficient but sensitive to outliers in the training data, but if the parameters of the matching score distribution is known, this method would suffice.

When the minimum and maximum values are estimated from the given set of matching scores, Min–max normalization retains the original distribution of scores except for a scaling factor and transforms all the scores into a common range [0, 1].

III. CLASSIFICATION

The classification stage is the decision making part of the recognition system.

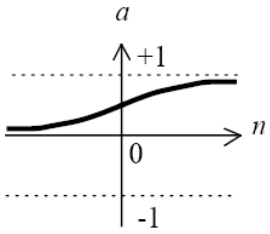
The Feed Forward Neural Network (back propagation) (2-NN) [40] is one of the most powerful classifiers. The number of hidden units of ANN should be selected to be high enough to model the problem at hand but not too high to avoid overfitting. The number of hidden units is selected to have the best performance on the validation set.

1) Network Architecture:

The input vectors are defined as a matrix called alphabet (133x1848), representing 1848 samples (66 character images x 28 "number of Arabic characters") of 133 elements (features). The target vectors are also defined with variable called target. Each target vector is a 28-element vector with a 1 in the position of the character and 0's everywhere else. For example, the letter "إ" is to be represented by a 1 in the first element (as "إ" is the first character of the alphabet), and 0's everywhere else. It is then required to identify the letter by responding with a 28-element target vector. The 28 elements of the target vector each represent a letter. To operate correctly, the network should respond with a 1 in the position of the letter being presented to the network. All other values in the output vector should be 0.

The neural network needs 133 neurons in its input layer and 28 neurons in its output layer to identify the letters.

The network is a two-layer network. The log-sigmoid [41] activation function at the output layer was picked because its output range (0 to 1) is perfect for learning to output Boolean values and because it is differentiable. The function generates outputs between 0 and 1 as the neuron's net input goes from negative to positive infinity. See Fig. 15.



a = logsig(n)

Fig. 15 Log-sigmoid transfer function

In building the network, the data was divided randomly into two categories. Training data consisted of 80% of the data (1540 characters consisting of 55 samples of each of the 28 characters). The remaining 20% of the data was assigned to the testing data (308 characters consisting of 11 samples of each of the 28 characters). Fig. 16 shows the network architecture.

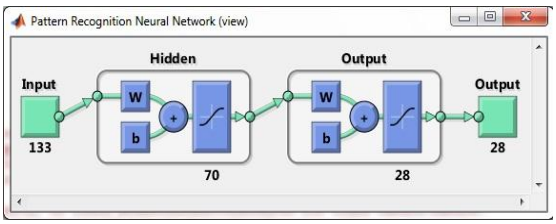


Fig. 16 network architecture

2) Network Training:

To create a network that can handle noisy input vectors (characters) it is best to train the network on both ideal and noisy characters. The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0's.

Back propagation training method was followed. This method was selected because of its simplicity and because it has been previously used on a number of pattern recognition problems. The method used in this work is called principle of gradient descent [42]. This function is useful because the conjugate gradient algorithms have relatively modest memory requirements. Memory is important when working with large networks, and yet it is much faster than other algorithms [43].

The gradient descent updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written as:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{g}_k \tag{9}$$

Where $\mathbf{x}_k$ is a vector of current weights and biases, $\mathbf{g}_k$ is the current gradient, and α_k is the learning rate. This equation is iterated until the network converges. The scaled conjugate gradient algorithm [43] is based on conjugate directions, but does not perform a line search at each iteration.

For evaluating the performance we choose to use the mean square error mse; the average squared error between the network outputs and the target outputs **t**. It is defined as follows:

$$F = mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \quad (10)$$

In this paper we stop the network training when the sum of squared error falls below 0.001.

3) Network testing:

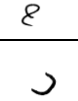
After many trials of eliminating, adding and modifying features and also adjusting network hidden layers (starting with 20 then 35 …70 neurons). After 70 neurons, network performance does not increase. A network of 70 neurons in hidden layer was able to predict about 88% of the input characters correctly (96% in the training phase and 88% in the testing phase).

IV. EXPERIMENT AND DISCUSSION

Our database of handwritten Arabic samples is CENPRMI popular dataset [44]. It includes Arabic off-line isolated handwritten characters. The database contains 11620 characters (about 332 samples for each character). These characters were written according to 12 different templates by 13 writers, with each template adopted by 5–8 writers. The 11620 character samples are divided into three groups 198, 67 and 67 for training, validating and testing purposes respectively.

It was carefully selected to represent initial, medial, and final of the 28 Arabic character forms. Table 2 shows a collection of samples of isolated Arabic characters.

TABLE 2 VARIATIONS IN CHARACTERS FROM DATASET

Letter	Variations			
ش				
ط				
و				
ك				
م				
ر				
ق				

As shown above, Table 2 contains variations of our dataset (by CENPRMI). These samples show that loops are sometimes introduced or filled and secondary objects are written in multiple styles. It contains also loops, under baseline, above baseline cavities, left and right directions. They differ in shape and style, making the recognition process very hard.

The test set size used in the experiments is 308 characters (11 different samples of each of the 28 characters); experimental results are presented in Table 3.

TABLE 3 RECOGNITION RATES FOR ARABIC CHARACTERS IN OUR SYSTEM.

Character	Recognition Percentage	Character	Recognition Percentage	Character	Recognition Percentage
أ	96%	ز	80%	ق	61%
ب	87%	س	78%	ك	81%
ت	69%	ش	88%	ل	96%
ث	76%	ص	**100%**	م	92%
ج	**100%**	ض	72%	ن	**100%**
ح	94%	ط	91%	ه	97%
خ	**100%**	ظ	83%	و	**100%**
د	83%	ع	66%	ي	
ذ	88%	غ	99%		
ر	89%	ف	**100%**		

The achieved results are very promising. Experimental results showed that the proposed method gives a recognition rate of about 88% for all letters although we got a success rate of 100% for some letters (in bold).

TABLE 4 COMPARISONS BETWEEN PREVIOUS RESULTS AND OURS

Previous study	Data Used	Approach	Feature set (number of features)	Results
A.A.Aburas *et al* [45]	Handwritten Arabic isolated characters collected from 48 volunteers	Haar Wavelet transform	Not mentioned (18)	70%
M. Z. Khedher, *et al* [46]	Handwritten Arabic isolated characters collected from 48 volunteers	Not mentioned	Width, Height, and Aspect Ratio, number of black pixels to character area ratio, Secondary Objects, Closed Loops, Filled Circles, Concavities, Stroke Sequence, (18)	73.4%
G. Abandah *et al* [47]	Handwritten Arabic isolated characters collected from 48 volunteers	Combination of multi-objective genetic algorithm and SVM	Normalized central moments, Zernike moments. letter form, secondary type and position, low-order elliptic Fourier descriptors, some statistical features extracted from the main body or the boundary (20)	not mentioned exactly 9% reduction of the classification error between combination of efficient feature extraction techniques versus moments of the Whole body alone.
A. T. Al-Taani *et al* [48]	Handwritten Arabic isolated characters written online on a special window on the screen by volunteers	Decision tree	Number of Segments, Cross-Points (Loop), Sharp Edges, Secondary Segment types, Similarity of Secondary Segments, Density Ratios, Horizontal–Vertical Orientation (not mentioned)	75.3%
G. A. Abandah *et al* [49]	Handwritten Arabic isolated characters collected from 48 volunteers	- Detected the secondary parts of the letter and extracting features from these parts. - Removed the secondary parts. - Extracted additional features from the raw main body, the main body's skeleton, and main body's boundary. - Classified by Linear Discriminant Analysis	Secondary components features, Main body features, Skeleton features, Boundary features (95)	87%
Our Research	CENPRMI dataset, Arabic isolated handwritten characters according to 12 different templates by 13 writers, each template adopted by 5–8 writers.	- Novel preprocessing operations. - Extracting of novel features (statistical, structural, and topological). - Classified by neural network (FFD) then updating weights. - Testing another group of characters.	- Structural Features (Upper and Lower profile- Horizontal and Vertical projection profiles) - Statistical Features (Connected components) - Topological features (End points- Pixel ratio- Height to width ratio) (133)	88%

From the previous table, it is obvious that our system does the best when compared with other systems in terms of recognition rate, although other systems make great contributions especially in terms of varieties of features, reducing of time consumed and mixing of modern classification techniques.

The accuracy of the system depends on many factors such as whether there is noise in the tested data or not, if the letter is poorly written, or is an unusual writing style. Because CENPRMI dataset contains most of those writing defects, the accuracy rate is not as high enough as expected.

The main contribution of this research includes the new offline Arabic handwritten character recognition system which is developed based on the novel extracted feature after some new techniques of preprocessing operations. The evaluation of our system is done by applying those features on feed forward neural network. The proposed method obtained competitive accuracy rates at 88%. We think that preprocessing operations as well as selecting most proper features can minimize classification error. For example, we use different kinds of noise removal (statistical and morphological) for erasing useless parts of the character which can occur during the hand writing process, ink stains or even by digitizing the image. We also use dilation for fixing damaged pixels of the character occurring as a result of preprocessing operations (binarization- noise removing) or during the digitizing process.

We extract features from the whole character, as well as its main body and secondary components which provide more valuable features that exploit the recognition potential of the secondary components of handwritten Arabic letters. These results also confirm the importance of the secondary components of the handwritten Arabic characters. For example, if we make a comparison between س and ش, ص and ض we will find no differences between each pair of them except for the secondary component.

We use not only different kinds of features (structural features, statistical features, topological features) which represent different aspects of the character's characteristics, but also (after many trials) we chose the most significant features for distinguishing between characters. We use structural features because a number of Arabic letters share the same primary shape, but differ only in the presence/absence and location/number of dots. For example, the three characters BAA (ب), TAA (ت), and THAA (ث). S. Mozaffri *et al* [50] commented that structural features remained more common for the recognition of Arabic script than that of Latin script.

We use also statistical features which are numerical measures computed over images or regions of images. We used vertical and horizontal projections [23, 24] which give us more valuable information to capture the distribution of ink along one of the two dimensions in the character. We also used both structural and statistical features [50]. Another kind of useful feature is topological features. We used end points, pixel ratio and height to width ratio as they give more dependable results in [45, 49].

Some notes on the features:

-We tried many kinds of features. One example is a feature which extracts the number of holes in the character because a lot of handwritten Arabic letters contain holes which can be a very important feature to distinct letters like (ع, ي, ق, س) from others (أ, ب, د) which don't contain holes. Unfortunately this feature doesn't perform well and doesn't give dependable results because of variation in writing Arabic handwritten characters, so we eliminated this feature from the feature list.

-Another kind of feature which has been examined is secondary component position. Unfortunately because of the variety of secondary components, they may take different positions and it is highly dependent on writing styles, so this feature doesn't give us dependable results so we decided not to use this feature.

We found that the Recognition rate is between 100% for easy to recognize letter forms and 61% for the hardest letter forms. Table 4 shows the ten letter forms that have the lowest classification rate.

TABLE 5 WORST 10 RECOGNIZED CHARACTERS

No	Character	Recognition Rate	Often mistaken as
1	ق	61%	ف
2	ع	66%	ح
3	ت	69%	ث , ز
4	ض	72%	ظ , ص
5	ث	76%	ف , ق
6	س	78%	ص
7	ز	80%	ر
8	ك	81%	ع , ح
9	د	83%	ك
10	ظ	83%	ش , ر

These ten letters are always drawn with loops or drawn with loops with some writing variations. There are substantial similarities among multiple Arabic letters that have loops. Often the sole difference between such letters is a subtle difference in the loop's shape. Moreover, letters with secondaries tend to have low recognition accuracies because of the variations in drawing the dots or hamzas which give inaccuracies in extracting the secondary type feature. Moreover multiple dots in some writing styles may be isolated (Naskh writing style) or continued dash (Rekaa writing style) as is obvious in table above (ن, ش, ق). After careful examination of the samples that were incorrectly recognized, we concluded that most of these samples are hard to recognize even by a human expert reader. However, we think that the door is open to search for extracting new features that capture subtle differences in loop shapes and secondary types.

V. CONCLUSIONS

This paper presents an approach for extracting features to achieve high recognition accuracy of handwritten Arabic characters. We present some useful techniques during the preprocessing phase including binarization, normalization and some noise removing methods. Our algorithm extracts useful features not only from the main body, but also from the secondary components of the character. It also overcomes some of the handwritten characters variations.

Paying more attention to the preprocessing operations including noise removal, filtering and dilation, as well as selecting

proper features for recognizing handwritten Arabic characters can give better recognition accuracy, therefore our features included statistical, morphological and topological features. Although, there are some challenges with some characters, the overall recognition rate is nearly perfect especially when compared to other handwritten Arabic characters systems.

After examining the recognition accuracy of each character we found that the recognition rate is between 100% for the easiest recognized characters such as (و, ن and ص) and 61% and 66% for the hardest recognized characters such as (ق and ع), respectively. We think that this is because those characters contain loops and also they can have different characteristics from various writing styles. Characters with dots also tend to have low recognition accuracies because of the variations in drawing the dots (dots can be linked to the main body of the character or even can be removed in some writing styles). This causes errors in extracting the secondary type feature.

Our future work includes increasing the efficiency of the proposed approach especially for the characters that were not recognized well by finding other powerful features, also including variations in writing the main body of the character and the secondaries. We hope also that we will complete a system for recognizing handwritten Arabic texts passing through segmentation techniques for segmenting the words to characters.

REFERENCES

[1] M. S. Khorsheed, "Automatic recognition of words in Arabic manuscripts," PhD thesis, University of Cambridge, 2000.

[2] N. Arica, Y.Vural, "Optical character recognition for cursive handwriting," IEEE Trans Pattern Anal Mach Intell, 24(6), 2002, pp. 801-813.

[3] L. Lorigo, V. Govindaraju, "Offline Arabic handwriting recognition: A survey," IEEE Trans Pattern Anal Mach Intell, 28(5), 2006, pp. 712-724.

[4] B. Al-Badr and S. Mahmoud, "A Survey and bibliography of Arabic optical text recognition," Signal Process. 41, 1, pp. 49-77, 1995.

[5] A. Nazif, "A system for the recognition of the printed Arabic characters," Master's thesis, Faculty of Engineering, Cairo University, 1975.

[6] V. Margner and H. El Abed, "Databases and competitions: strategies to improve Arabic recognition systems," Springer, vol. 4768, 2008, pp. 82-103.

[7] M. Cheriet, "Visual recognition of Arabic handwriting: challenges and new directions," Springer, vol. 4768, 2008, pp. 1-21.

[8] L. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: a survey," IEEE Trans. Pattern Anal. Mach. Intell. 28, 5, 2006, pp. 712-724.

[9] M. Khedher and G. Al-Talib,"Recognition of secondary characters in handwritten Arabic using fuzzy logic," Proceeding of the International Conference on Machine Intelligence, 2005.

[10] S. El-Dabi, R. Ramsis and A. Kamel, "Arabic character recognition system: A statistical approach for recognizing cursive typewritten text," Pattern Recognition, vol. 23, no. 5, 1990, pp. 485-495.

[11] T. S. El-Sheikh and S. G. El-Taweel, "Real-time Arabic handwritten character recognition," Pattern Recognition, vol. 23, no. 12, 1990, pp. 1323-1332.

[12] F. El-Khaly and M. A. Sid-Ahmed, "Machine recognition of optically captured machine printed Arabic text," Pattern Recognition, vol. 23, no. 11, 1990, pp. 1207-1214.

[13] A. Sabri, "Arabic character recognition using Fourier descriptors and character contour encoding," Pattern Recognition, vol. 27, no. 6, 1994, pp. 815-824.

[14] J. Ayman and M. Laheeb, "Arabic handwritten characters recognized by Neocognitron Artificial Neural Network," University of Sharjah, Journal of Pure & Applied Sciences, vol. 3, no. 2, 2006.

[15] A. Amin, H. Al-Sadoun and S. Fischer, "Hand-printed Arabic character recognition system using an Artificial Network," Pattern Recognition, 29, 1996, pp. 663-675.

[16] A. Amin, "Recognition of hand-printed characters based on structural description and inductive logic programming," Pattern Recognition Letters, 24, 2003, pp. 3187-3196.

[17] G. Olivier, H. Miled, K. Romeo & Y. Lecourtier, "Segmentation and coding of Arabic handwritten words," Proc. 13th Int'l Conf. Pattern Recognition, 3, 1996, pp. 264-268.

[18] I.S.I. Abuhaiba and P. Ahmed, "Restoration of temporal information in off-Line Arabic handwriting," Pattern Recognition, 26, pp. 1009-1017, 1993.

[19] I.S.I. Abuhaiba, S.A. Mahmoud and R.J. Green, "Recognition of handwritten cursive Arabic characters," IEEE Trans. Pattern Analysis and Machine Intelligence, 16, 1994, pp. 664-672.

[20] M. Dehghan, K.Faez, M. Ahmadi and M. Shridhar, "Handwritten Farsi (Arabic) word recognition: A holistic approach using discrete HMM," Pattern Recognition, 34, 2001, pp. 1057-1065.

[21] H. Al-Yousefi and S.S. Udpa, "Recognition of Arabic characters," IEEE Trans. Pattern Analysis and Machine Intelligence, 14, 1992, pp. 853-857.

[22] S. Abdelazeem and E. EL-Sherif, "Arabic handwritten digit recognition," international Journal on Document Analysis and Recognition (IJDAR),11, 3, 2008, pp. 127-141.

[23] G. Abandah and N. Anssari, "Novel moment features extraction for recognizing handwritten Arabic letters," Journal of Computer Science, 5, 3, 2009, pp. 226-232.

[24] K. Sherif and M. Mostafa, "A Parallel design and implementation for backpropagation Neural Network using MIMD architecture," IEEE, 1996, pp. 1361-1366.

[25] K. Sherif and M. Mostafa, "A Parallel design and implementation for backpropagation Neural Network using MIMD architecture" IEEE, 1996, pp. 1472-1475.

[26] M. A. Ali, "Arabic handwritten characters classification using learning vector quantization algorithm," Image and Signal Processing, Lecture Notes in Computer Science, Springer, vol. 5099, 2008, pp. 463-470.

[27] Y. Chherawala, P. P. Roy and M. Cheriet, "Feature design for offline Arabic handwriting recognition: handcrafted vs automated?," 12th International Conference on Document Analysis and Recognition, 2013, pp. 290-294.

[28] T. Bluche, H. Ney, C. Kermorvant, "Feature extraction with convolutional neural networks for handwritten word recognition," 12th International Conference on Document Analysis and Recognition, 2013, pp. 285-289.

[29] L. Rothacker, S. Vajda, and G. A. Fink, "Bag-of-features representations for offline handwriting recognition applied to Arabic script," in Proceedings of the 3rd International Conference on Frontiers in Handwriting Recognition (ICFHR'12), Bari, Italy, 2012, pp. 149-154.

[30] M. Kozielski, P. Doetsch and H. Ney, "Improvements in RWTH's system for off-line handwriting recognition," 12th International Conference on Document Analysis and Recognition, 2013, pp. 935-939.

[31] S. A. Mahmoud and S. O. Olatunji, "Automatic recognition of off-line handwritten Arabic (Indian) numerals using support vector and extreme learning machines," International Journal of Imaging 2, A09, 2009.

[32] S. A. Mahmoud and S. Owaidah, "Recognition of off-line handwritten Arabic (Indian) numerals using multi-scale features and support vector machines," Arabian Journal for Science & Engineering (Springer), vol. 34, iss. 2B, 2009, pp. 429-444.

[33] M. Gargouri, S. Kanoun and J.M. Ogier, "Text-independent Writer Identification on Online Arabic Handwriting," 12th International Conference on Document Analysis and Recognition, 2013, pp. 428-432.

[34] M. T. Parvez and S. A. Mahmoud, "Offline arabic handwritten text recognition: A Survey," journal of the Association for Computing Machinery, vol. 45, no. 2, 2013, pp. 23:1-23:35.

[35] N. Otsu. A threshold selection method from gray-scale histogram. IEEE Transactions on System, Man, and Cybernetics. 9, 1979, pp. 62-66.

[36] G. Lazzara and T. Géraud, "Efficient multiscale Sauvola's binarization," Springer- Verlag Berlin Heidelberg, 2013.

[37] M. Cheriet, N. Kharma, C.L. Liu and C. Y. Suen, "Character Recognition Systems," John Wiley, 2007, pp. 36-38.

[38] Jae. S.L, "Two dimensional signal and image processing," Prentice Hall Ptr, Upper Saddle River, New Jersey. 07458, 1990.

[39] J. Hann, M.Kamber, "Data Mining: concepts and techniques," 3rd Edition. Morgan Kaufman, 2011.

[40] S. Daniel, K. Vladim ŕ, P. Jiri, "Introduction to multi-layer feed-forward neural networks," ELSEVIER, Chemometrics and Intelligent Laboratory Systems 39, 1997, pp. 43-62.

[41] P. Chandra, "Sigmoidal function classes for Feedforward Artificial Neural Networks," Kluwer Academic Publishers, 2003, pp. 185-195.

[42] M. F. Moller, "A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks," vol. 6, 1993, pp. 525-533.

[43] M. H. Beale, T. Hagan and B. Demuth, "Neural Network toolbox 7, User's Guide," The MathWorks, Inc, 2010.

[44] H. Alamri, J.Sadri, C.Y.Suen, N.Nobile, "A novel comprehensive database for Arabic off line handwriting recognition," The11thInternational Conference on Frontiers in Handwriting Recognition (ICFHR. 2008), pp. 664-669.

[45] A. A. Aburas and S. M. Rehiel, "Off-line Omni-style handwriting Arabic character recognition system based on wavelet compression," Arab Research Institute in Sciences & Engineering ARISER, 2007, pp. 123-35.

[46] M. Z. Khedher, G. A. Abandah, and A. M. Al-Khawaldeh, "Optimizing feature selection for recognizing handwritten Arabic characters," World Academy of Science, Engineering and Technology 4, 2005, pp. 81-84.

[47] G. Abandah, and N. Anssari, "Novel moment features extraction for recognizing handwritten Arabic letters," Journal of Computer Science, 5, 3, 2009, pp. 226-232.

[48] A. T. Al-Taani and S. Al-Haj, "Recognition of On-line Arabic handwritten characters using structural features," Journal of Pattern Recognition Research, 2010, pp. 23-37.

[49] G. A. Abandah, K. S. Younis and M. Z. Khedher, "Handwritten Arabic character recognition using multiple classifiers Based on letter form," Conf. on Signal Processing, Pattern Recognition, & Applications Austria, 2008, pp. 128-133.

[50] S. Mozaffri, K. Faez and M. Ziaratban," Structural decomposition and statistical description of farsi/arabic handwritten numeric characters," The 8th International Conference on Document Analysis and Recognition (ICDAR), 2005, pp. 237-241.

Ahmed T. Sahlol Obtained Bachelor degree from Mansoura University, Egypt in 2004, 2 years Diploma from Mansoura University in 2006, Master degree from Mansoura University, Egypt in 2010. Then he became lecturer assistant teaches computer science labs and tutorials for undergraduate students. He is currently visiting researcher at Concordia University in Canada and he got a governmental scholarship to study his PhD. His research interests include pattern recognition, optical character recognition and image processing.

Ching Y. Suen is the Director of CENPARMI and the Concordia Chair on AI and Pattern Recognition. He received his Ph.D. degree from UBC (Vancouver) and his Master's degree from the University of Hong Kong. He has served as the Chairman of the Department of Computer Science and as the Associate Dean (Research) of the Faculty of Engineering and Computer Science of Concordia University. He has served at numerous national and international professional societies as President, Vice-President, Governor, and Director. He has given 40 invited/keynote papers at conferences and 180 invited talks at various industries and academic institutions around the world. He has been the Principal Investigator or Consultant of 30 industrial projects. His research projects have been funded by the ENCS Faculty and the Distinguished Chair Programs at Concordia University, FCAR (Quebec), NSERC (Canada), the National Networks of Centres of Excellence

(Canada), the Canadian Foundation for Innovation, and the industrial sectors in various countries, including Canada, France, Japan, Italy, and the United States.

M. M. R. EL Basyouni Obtained Bachelor degree from Zagazig University, Egypt in 1979, Diploma from Ismailia University in 1988, PhD degree from Bucharest University, Romania in 1995.He became head of computer teacher preparation department from 2006 until 2011. Then he became professor in 2012 the Dean of the faculty of specific education in 2013. His research interests are computer as a learning aid and human computer interaction.

Abdelhay A. Sallam PhD, is a senior member of the IEEE and Professor Emeritus of Electrical Engineering at the Port Said University. Dr. Sallam has taught courses in power systems, computer methods in power system analysis, conventional machines, distribution systems, logic circuits, and microprocessor structure. In addition, he has served as a consultant, advising companies on the design, installation, and maintenance of power networks, substations, and electric distribution systems.