Fuzzy Modeling for Handwritten Arabic Numeral Recognition

Dhiaa Musleh, Khaldoun Halawani, and Sabri Mahmoud
Information and Computer Science Department, King Fahd University of Petroleum and Minerals Saudi Arabia

Abstract: *In this paper we present a novel fuzzy technique for Arabic (Indian) online digits recognition. We use directional features to automatically build generic fuzzy models for Arabic online digits using the training data. The fuzzy models include the samples' trend lines, the upper and lower envelops of the samples of each digit. Automatically generated weights for the different segments of the digits' models are also used. In addition, the fuzzy intervals are automatically estimated using the training data. The fuzzy models produce robust models that can handle the variability in the handwriting styles. The classification phase consists of two cascaded stages, in the first stage the system classifies digits into zero/nonzero classes using five features (viz. length, width, height, height's variance and aspect ratio) and the second stage classifies digits 1 to 9 using fuzzy classification based on directional and segment histogram features. Support Vector Machine (SVM) is used in the first stage and syntactic fuzzy classifier in the second stage. A database containing 32695 Arabic online digits is used in the experimentation. The results show that the first stage (zero/nonzero) achieved accuracy of 99.55% and the second stage (digits from 1 to 9) achieved accuracy of 98.01%. The misclassified samples are evaluated subjectively and results indicate that humans could not classify ≈35% of the misclassified digits.*

Keywords: *Automatic fuzzy modeling, arabic online digit recognition, directional features, online digits structural features.*

Received November 17, 2014; accepted April 12, 2015

1. Introduction

Automatic recognition of handwritten text is important in providing a natural and convenient two way communication between users and computers. Online text recognition deals with handwriting captured by a tablet or similar touch screen devices. In the online recognition, the two-dimensional coordinates (x, y) of successive points of writing are stored based on their order whereas the complete writing is available as an image in the offline handwriting. In general, online handwritten recognition uses spatio-temporal representation of the input, whereas the offline handwritten recognition analyzes the spatio-luminance of an image [23].

In recent years many researches have been conducted on the recognition of offline [9, 11, 20, 24] and online [12, 25] Latin digits. Most of the research on Arabic digits addressed offline Arabic handwritten digits [1, 5, 16, 17, 18, 19, 21] while few addressed online Arabic digits [2, 6, 7, 13, 14, 15]. The reasons for this may be attributed to the challenges related to online Arabic handwriting recognition systems. These challenges include the need for special hardware; the lack of comprehensive benchmark database for Arabic online text; techniques of other languages may not work for online Arabic text recognition. In addition, the writing using online devices is less controlled than writing using a pen on paper. As a result, the variability in writing is more which makes the problem of online Writer-independent handwriting recognition a challenging pattern recognition problem.

Automatic recognition of handwritten Arabic (Indian) digits has a variety of applications including banking systems and forms filling. Figure 1 shows samples of handwritten Arabic digits from 0 to 9 along with their printed versions.

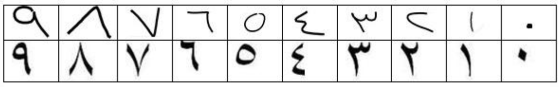


Figure 1. Arabic (Indian) digits 0 to 9.

In this paper we present a novel technique for the automatic recognition of Arabic online digits. While the skeleton (generated using offline Arabic character clustering) is used to define the character models in [3] and models were not used (only fuzzy similarity measure is used) in [22]. In this work, we automatically generate the fuzzy intervals based on the analysis of the training samples (and not set manually as in [3, 22]). In addition, our proposed technique automatically generates weights for the different segments using the training samples. These weights are integrated in the fuzzy models while weights were not used in [10, 22] as the segments are assumed of equal weight.

For the classification, we use a two stage approach where Support Vector Machine (SVM) is used in the first stage using statistical features and fuzzy-based approach using the automatically generated models in the second stage. The second stage has an integrated feedback verification step which verifies the recognized test sample label. If the first label does not pass validation then the next label in the list is selected in the feedback loop and so on. Figure 2 shows the overall block diagram of the implemented system.

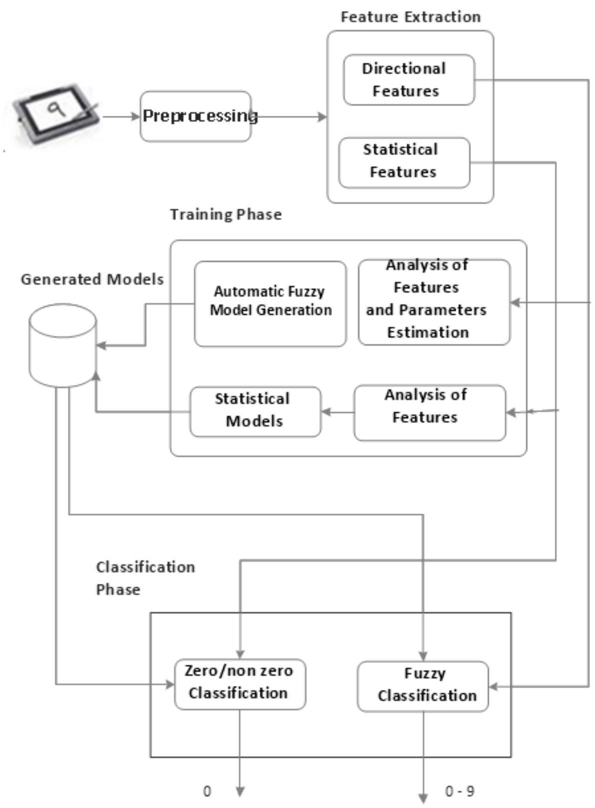


Figure 2. Overall block diagram of our approach.

The contribution of the paper is automatically generating robust fuzzy models for Arabic online digit recognition using the training data. The models include the trend lines, the upper and lower envelops of the samples‟ of each digit.

The fuzzy durations are automatically generated and are set at the digit segment level. In addition, automatic generation of weights at the segment level was integrated into the model which indicates the importance of the different segments of the digit for recognition. This an improvement over previous works [10, 22] in several aspects (viz. automatic model generation over both [10, 22]; weights of the different segments were not used in [10, 22] assuming equal weights; only similarity estimate is used in [22] and no fuzzy modeling; Parvez and Mahmoud in [22] applied it to the skeleton of the Arabic offline characters and Halawani in [10] applied it to the polygonal approximation of the Arabic offline characters and here we apply it to Arabic online digit recognition). A fuzzy syntactic classifier is implemented with integrated feedback stage that verifies the selected classes for test samples using segments' histogram features. This stage improved the syntactic fuzzy classification.

The rest of the paper is organized as follows. Section 2 presents literature review of Arabic online digits. Features used in this work are detailed in section 3; section 4 describes the proposed analysis and automatic fuzzy models' generation approach; the classification phase is discussed in section 5; the experimental results are reported in section 6; and finally the conclusions are presented in section 7.

2. Literature Review

Comparatively few works were reported on online Arabic digits‟ recognition. An early approach was proposed by Beigi *et al.* [7] based on spatial features. In their approach, a five dimensional feature vector is used (the difference between adjacent coordinates, the sine and cosine at each point and the absolute y-coordinate of each point shifted by the computed baseline value). Hidden Markov Models (HMMs) was used in the classification phase. For training, 6000 samples of Arabic digits written by 20 different writers were used whereas 700 samples written by 14 writers were used for testing. The reported recognition rate was 93.14%. This recognition rate is not adequate for practical applications.

Trajectory/velocity modeling approaches were applied in several works. Kherallah *et al.* [13] proposed a neural network system to recognize online digits based on Beta-circular approach. Their system combines the kinematics and geometry in the trajectory modeling. Each stroke is represented by dynamic Beta and static circular parameters. These parameters are extracted from the curvilinear velocity of the points. A dataset containing 10000 digits was collected to test the performance of the proposed system. 7000 digits (70%) were used in the learning phase and the remaining 3000 digits (30%) were used to test the system. The reported recognition rate was 95%. In their later work [14] they improved the modeling quality of handwriting by introducing elliptical parameters. Low pass Chebyshev filter was applied to the data points as a preprocessing step to eliminate duplicate data points. They used Kohonen maps in this work. The reported global recognition rate was about 90%. A database of 24000 digits was used. 30% of the data was used for the training and the remaining 70% for testing. This data may justify the reduced recognition rate compared to the previous work. To reduce the dimensionality, the authors proposed a new method of the handwritten trajectory modeling based on Multi-Layers Perception (MLP) developed in fuzzy concept [15]. The training is based on an association of the Self Organization Maps (SOM) with Fuzzy K-Nearest Neighbor (F-KNN). The interaction between the different techniques (MLP, F-KNN and SOM) contributes to the increase in the recognition rate of the proposed method. A database of 30,000 Arabic digits written by 24 persons is used to test the performance of the system. The reported recognition rate was about 95.08%, an improvement over the previous work.

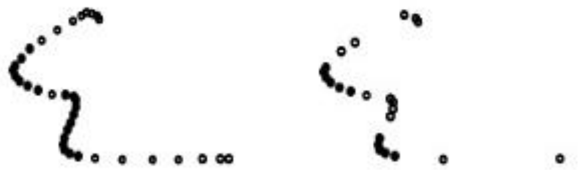
Al-Taani [6] proposed a technique based on structural features and transition network. In their approach, the slope values for each sample were calculated from the (x, y) coordinates and used to record the change of direction. Then, the primitives of the samples were extracted based on the changing of signs of the slop. The string of these primitives describes the digit. Each string is considered as a production of a specific grammar. In order to recognize a digit, they identify to which grammar its string belongs. A finite transition network, which contains the grammars of the digits, was used for classification purpose. The transition network compares the string primitives with the corresponding digit. The proposed technique was tested using a dataset of 3000 digits with a reported recognition rate of 95%.

In 2012, Abdelazeem *et al*. [2] proposed a system that uses both temporal and spatial features to recognize Arabic online digits. In their work, a database containing 30,000 digits were collected from 300 writers to test the proposed system. The classification phase is composed of two stages, the first stage recognizes digit zero and the second stage recognizes the remaining digits. The overall reported recognition rate was 98.73%. Although their system was developed mainly for Arabic online digits, offline features were used by converting the user’s strokes into a bitmap image. Table 1 shows summary of the surveyed techniques for Arabic online digits recognition.

3. Feature Extraction

Online digits are captured using a digital tablet, touch screen, Personal Digital Assistants (PDAs), etc. Writing using these devices is less controlled than writing using a pen on paper. Additionally, data collected by using these devices is affected by hardware imperfections and the trembles in writing. Hence, preprocessing is crucial to achieve better recognition rate. In this work, simplification, smoothing and normalization are applied to the data before feature extraction.

Douglas-Peucker algorithm [8] is used to simplify the digit curve by reducing the number of points representing the curve. The algorithm removes unimportant points from the original curve to simplify the curve as shown in Figure 3.



a) Before simplification. b) After simplification.

Figure 3. Digit four.

An algorithm to remove the small curve variations and noise is applied. Any small segment at the beginning or at the end is considered as noise. In addition, if there are two consecutive segments that are very close to each other but having different directions, the first one is considered as a noise. In this work we also normalize the data by adjusting all the samples‟ lengths to 25 points. This length is derived based on the common lengths of samples. We experimentally found that this length is enough to model the samples.

Extracting good and representative features is very important in any recognition system as it contributes to the recognition performance [4]. Three sets of features are extracted in this work. Shape features are extracted first to classify digits into zero or nonzero. Then, directional and histogram-based features are used to recognize nonzero digits (digits from one to nine).

Table 1. Summary of results for Arabic (Indian) online digits recognition.

Author(s)	Features	Classifiers	Data (digits)		Accuracy	# Writers
			Training	Testing		
Beigi et al., 1994 [7]	Geometric features	HMM	6000	700	93.14%	34 writers
Kherallah et al., 2002[13]	Circular and Beta features	Neural Networks	7000	3000	95%.	Writer dependent
Kherallah et al., 2004[14]	Elliptical and Beta features	Self-organizing map (SOM). Kohonen Neural Networks	7200	16800	90 %	Writer dependent
Kherallah et al., 2008 [15]	Trajectory-based features (elliptic parameters) and velocity based features (Beta function parameters).	MLPNN + SOM + FKNN	20000	10000	95.08%	24 writers
Al-taani, 2010 [6]	The changes in the slope’s values + The primitives’ string	Finite Transition Network (template matching)	--	3000	95 %	100 writers
Abdul Azeem et al., 2012[2]	Temporal (online) and spatial (offline) features	SVM	24000	6000	98.73%	300 writers

Figure 4 shows digit 6 before and after preprocessing.

a) Before preprocessing. b) After Preprocessing.

Figure 4. Digit six.

3.1. Shape Features

Unlike other digits, Arabic digit zero is just like a dot. The way it is written when magnified have different shapes as shown in Figure 5. These shapes are sometimes confused with other digits like 1, 5, 8, etc. A human seeing the samples in Figure 5 will not be able to identify them as zeroes.

Figure 5. Different samples of Arabic digit zero ‘0’.

Due to these difficulties, we address the ‘zero’ digit separately based on its length, width, height, heights’ variance and aspect ratio (height-to-width ratio).

3.2. Directional Features

Each digit is represented by taking the directional information of all points representing that digit. Let $P=(xi, yi)$, where $i=1, 2,\ldots, n$ be the sequence of points that represent digit D. The directional features are $\Theta_D=[d_1, d_2, \ldots, d_{n-1}]$, where di is the angle between the points (x_i, y_i) and (x_{i+1}, y_{i+1}).

3.3. Histogram-based Features

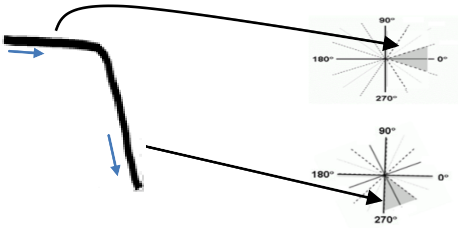
Basically, each digit consists of a set of segments and each segment has its own orientation. In our histogram-based features, a segment’s orientation takes value from a finite set of orientations or directions called standard writing directions. Standard directions can be regarded as quantization of strokes’ directions. Figure 6-a shows the standard writing directions with 22.5^o gap (i.e., number of directions is sixteen). The proposed histogram-based features are based on the histogram of digits’ segments orientation. Let $Segi=1, 2, .., n,$ be the segments of a digit D. The histogram features vector $S_D=[p_1, p_2, \ldots, p_{16}]$, where 16 represents the number of directions used in our work, and p_j is the percentage of the segments of digit D in direction j such that:

$$P_i = \frac{\textit{Number of segments of direction } j}{\textit{Number of all segments}} \tag{1}$$

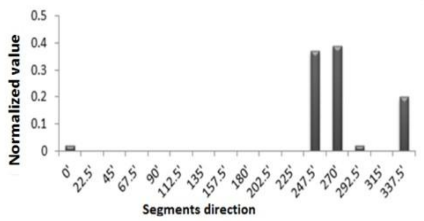
a) Standard writing direction with 22.5^o gap. b) The overlap between direction 337.5^{o}and direction 0^o.

Figure 6. Standard writing directions.

Due to the variations in writing style, the computation of the segments’ directions should tolerate some level of variations in writing directions. To alleviate this problem, we represent each direction i by i ± 22.5^o (i.e. direction i represents all segments having directions from i - 22.5 up to i + 22.5). For example, direction 0^o includes all segments having directions from -22.5^o (337.5^o) up to +22.5^o as shown in Figure 6-b. Accordingly, there will be an overlap of 22.5^o between consecutive directions’ segments. The overlap between direction 337.5^o and direction 0^o is shown in Figure 6-b. Figure 7-a shows a sample of digit ‘six’ with the directions of its segments; the histogram vector values for this sample are [0.02, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ,0.37, 0.39, 0.02, 0, 0.20] and its histogram is shown in Figure 7 -b.



a) Sample of digit six ‘6’ with the directions of its segments.



b) Segments histogram.

Figure 7. Histogram-based features.

4. Automatic Fuzzy Models’ Generation

In this section we present the use of fuzzy models to address the problem of variability in writing. The following sections address the automatic fuzzy models’ generation and the classification using the generated fuzzy models.

4.1. Automatic Fuzzy Modeling

The directional feature representation of digits reflects their shapes. This representation is used to build generic models that represent the digits (i.e., fuzzy models). In order to generate the fuzzy models of digits, the lengths of all digits' samples are normalized to a unique length. In this work, we normalized all the samples lengths to 25. This length is derived based on the common lengths of the samples. After normalizing the samples, the directional features of the training samples are extracted. The length of the directional features' vector is 24 since it represents the sequence of angles between each two consecutive points. Figure 8 shows the directional representation of several samples of digit 'three' written by different writers. We implemented two approaches; Model Trend Line (MTL) and Model Envelope (ME). MTL is a single trajectory that represents the mean of all samples. ME uses the top and bottom envelops of the aggregated samples. Top and bottom envelops of the samples form a good model to describe the shape of the digits' models.

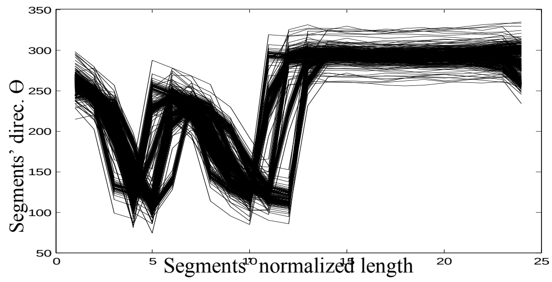


Figure 8. Directional representations of several samples of digit three.

4.2. Model Trend Line

The Trend Line (TL) is a trajectory that represents the center of the aggregated samples and hence the general shape of the samples of the classes. The aggregation is done by estimating the mean at each line segment (sampling point):

$$TL_m(s) = \frac{1}{n_s} \sum_{i=1}^{n_s} Y_i \tag{2}$$

Where s is the line segment index, n_s is the number of points in segment s, y is the segments' direction at point i of the segment. The standard deviation is given by the following equation:

$$TL_s = \sqrt[2]{\frac{1}{n_s} \sum_{i=1}^{n_s} \left(Y_i - TL_m(s)\right)^2} \tag{3}$$

This representation shows the general shape of the digit. Figure 9 shows an example of the MTL for digit three.

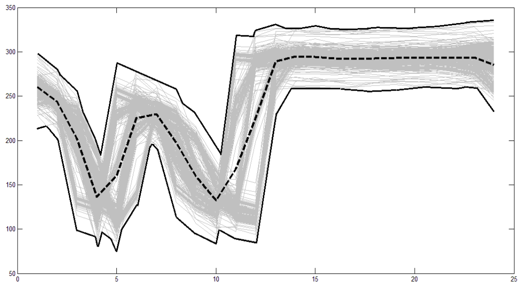


Figure 9. MTL and ME for digit three.

5. Model Envelop

ME is represented by two curves that surround the main body of the samples aggregation. The top and bottom envelops describe the range and shape of each digit main body. Every sample is represented as a function y=C(x). Given a set C={C_1, C_2, …, C_n } of x-curves that represent n samples. The bottom envelope is defined as the point-wise minimum of all samples. The bottom envelope for the set C at position i can be defined as follows:

$$B_C\,[i] = min\ C_i\,(x) \tag{4}$$

Where x=1, 2, …, n_S , and i=1, 2, …, N, is the number of points in each sample. Similarly, the top envelope of C is the point-wise maximum of the samples curves in the set:

$$T_c\,[i] = max\ C_i(x) \tag{5}$$

Where x = 1, 2, … n_S , and i = 1, 2, ..N. Figure 9 demonstrates the aggregated samples that form digit 'three'. The top envelop, shown in bold, represents the upper limit of the model and the bottom envelop, shown in bold, represents the lower limit.

In our experiments, we used both the MTL and the ME to build the fuzzy models that represents each digit class. For each training sample, the fuzzy model generated for that sample is a series of points that represent the directional features of that sample with the standard deviation (σ) of y-values of all the training samples from the same class at each point.

The standard deviation for each class (digit) is calculated from the training samples of that class. Figure 10 shows a fuzzy model for digit 'three' which is generated from the directional representations of the training samples of this digit.

The shaded area in Figure 10 represents the fuzzy tolerance of the model. The width of this area on both sides is related to the standard deviation which is estimated in the training phase. The width of the fuzzy tolerance varies from one point to another according to the model as estimated in the training phase. Figure 11.a shows the directional representation of several samples of digit 'Six' written by different writers.

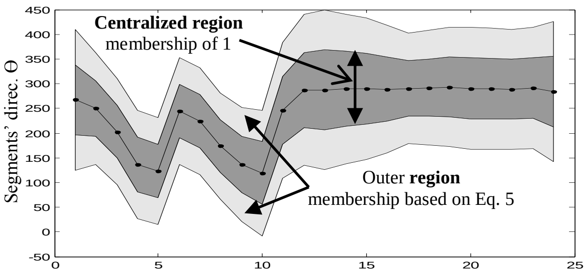
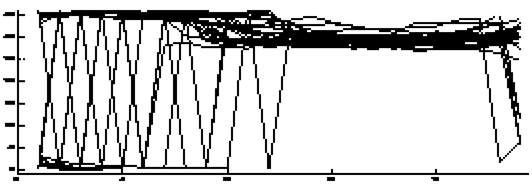
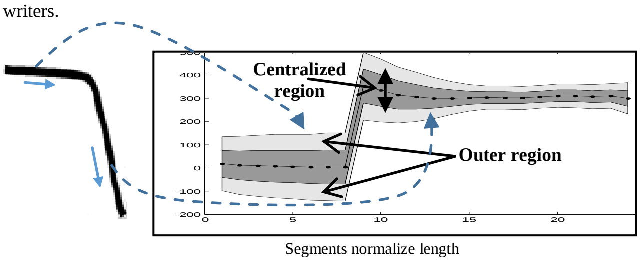


Figure 10. Generated Fuzzy models for digit 'Three'.



a) the directional representation of several samples of digit 'Six' written by different writers.



b) shows a sample of digit 'Six' mapped to the fuzzy model generated for these samples of digit 'Six'.

Figure 11. Generated Fuzzy models for digit 'Six'.

5.1. Classification

The proposed classification technique has two stages, namely Zero/Nonzero and Fuzzy classification. The first stage recognizes digit zero '0' and the second stage recognizes digits one '1' to nine '9'.

5.2. Zero/Nonzero Classification

In this stage, the shape features are used to classify digits into zero and nonzero. SVM classifier with Radial Basis Function (RBF) as the kernel was used in this stage. In this classification, we use the length, width, height, heights' variance and aspect ratio of the digit as features.

5.3. Fuzzy Classification

This stage is used to classify nonzero digits (digits 1 to 9). Directional and histogram-based features are used in this stage. Directional features' vector of each test sample is compared with all fuzzy models to find the most similar one.

As a result of the normalization process, the directional features of each digit are represented by 24 features (angles). At each point, fuzzy comparison between the sample and the models is done to estimate their similarity. Figure 12 illustrates the fuzzy comparison between a sample and a model. The shaded area in Figure 12 represents the fuzzy tolerance of the model. The width of this area is based on the standard deviation estimated in the training phase, and it is different from one point to another.

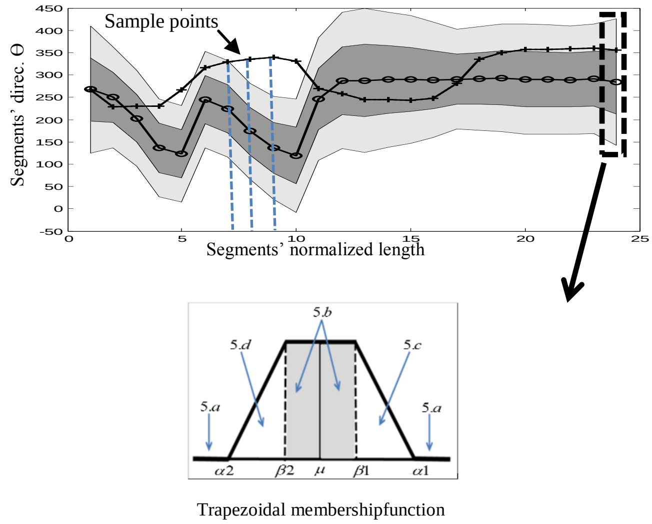


Figure 12. Fuzzy comparison between a sample and a model.

The membership of any point outside the tolerance region (shaded area) is considered as zero. For the shaded area, the membership of any point in the middle area (dark area) is taken as 1 which represents the peak of a fuzzy set. The membership outside this area (light area), Membership Value (Mv), is calculated based on Equation 5.

$$M_v(x,m) = \begin{cases} 0, & x > \alpha_1 \text{ or } x < \alpha_2 & \text{(a)} \\ 1, & \beta_2 \le x \le \beta_1 & \text{(b)} \\ \dfrac{\alpha_1 - x}{\alpha_1 - \beta_1}, & \beta_1 < x \le \alpha_1 & \text{(c)} \\ \dfrac{x - \alpha_2}{\beta_2 - \alpha_2}, & \alpha_2 \le x \le \beta_2 & \text{(d)} \end{cases} \qquad (6)$$

The values of α_1, α_2 and β_1, β_2 were taken as fixed values in [10] and [22] while here these values are automatically generated from the training samples.

The MV represents the weight of the similarity between sample *x* and the model at sampling point *m*. The value of MV ϵ [0, 1] such that '0' (5-a) means least similarity between the sample (outside the shaded region) and the model at that sampling point. Membership value '1' (5-b) represents the maximum similarity (centralized region). Membership value in the light shaded area (outer region) is calculated using (5-c) and (5-d) of Equation 5. Figure 12 shows these four cases for digit sample *S* and a digit model *M*, *N* is the number of points. The overall similarity between *M* and *S* is given by

$$Sim(S,M) = \frac{1}{N}\sum_{i=1}^{N} M_v(S(i),M(i)) \qquad (7)$$

The zoomed part in Figure 12 illustrates the fuzzy model in 2-dimentional representation. This illustration shows the real form of the fuzzy model, including the two tolerance regions. The figure shows a number of sampling points of the model; the similarity check is done at each one of these points.

To enhance the achieved recognition rates, we used histogram-based features to validate the decision taken by the fuzzy classifier as follows. Two sets of features are extracted from the testing sample, directional and histogram-based features. The first one is used by the fuzzy classifier to identify to which class the sample belongs to. The second one is used to validate the recognized class. The decision taken by the fuzzy classifier is evaluated based on the histogram-based features of the sample and the selected class. If they are similar, then the classification confidence of this decision is accepted. Otherwise, the selected class is rejected and the control goes back to the fuzzy classifier to select the next candidate class. The validation process is illustrated in Figure 13.

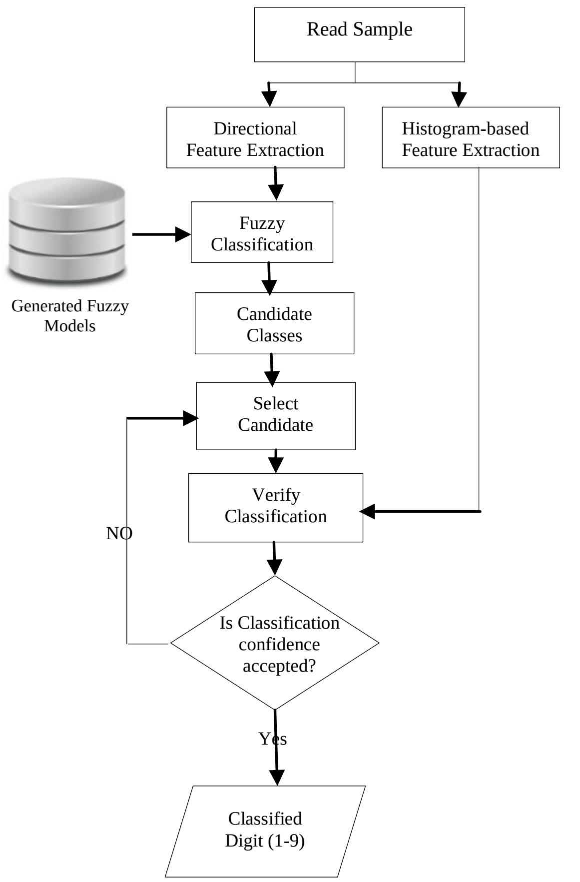


Figure 13. Fuzzy classification block diagram.

6. Experimental Work

In this section we present the experiments that were conducted to evaluate our technique for the recognition of Arabic online digits. In addition, the details of the used database and the results are discussed. In our experiments, we used the Arabic Online Digits Database (AOD*) proposed by Abdelazeem *et al.* [2]. AOD was collected from 300 writers of varying ages and without enforcing any constraints on the digit size, orientation or number of strokes per digit.

*Available at: \\ http://www.aucegypt.edu/sse/eeng/Pages/AOD.aspx

More than 32,000 online Arabic digits were collected by asking each writer to write an average of 10 samples per digits. About 78% of the AOD database was used in training and the remaining 22% of the data was used for testing.

The proposed technique is composed of two stages. In the first stage, all testing data are classified into zero or nonzero digits. SVM classifier with RBF was used in this stage. The classifier was designed for two-class problem with five dimensional feature vector (length, width, height, heights' variance and aspect ratio). An overall accuracy of 99.55% was achieved in this phase. The nonzero digits are classified by using the fuzzy classifier in the second stage. A 20% of the training data is selected as validation data which is used for fuzzy model parameters' estimation. The values for the first tolerance ($\beta1$ and $\beta2$) and the second tolerance ($\alpha1$ and $\alpha2$) are chosen experimentally as ($\sigma/4$) and ($\sigma/2$), respectively. These parameters are used in the experiments with the extracted directional features of the test data. A recognition rate of 93.36% was obtained in this stage using the first candidate. To improve the recognition rate of this stage, we use histogram-based features to validate the decision taken by the fuzzy classifier as discussed in section 5 such that if the decision based on the first candidate is inadequate the next candidate of the fuzzy classification is selected. After applying this improvement, the recognition rate reached 98.01%.

We evaluated our directional features using SVM classifier to classify Arabic digits. An average recognition rate of 93% is achieved. This indicates that our fuzzy models with the proposed fuzzy structural classifier are more effective than using the directional features with SVM classifier for online Arabic digit recognition. Table 2 shows the confusion matrix of our classification which includes fuzzy- based and histogram-based classification.

Table 2. Confusion matrix of the enhanced fuzzy classification.

Digit	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	836	1	5	0	0	11	1	0	0
2	2	653	1	8	2	0	0	0	6
3	4	2	651	0	0	0	0	0	1
4	0	7	1	676	0	3	0	0	2
5	0	0	0	0	694	0	0	1	2
6	0	1	0	3	0	650	0	0	9
7	1	3	4	1	0	2	677	3	0
8	3	0	6	1	0	0	0	661	2
9	1	1	6	1	2	14	0	1	659
R.R.	**98.7**	**97.8**	**96.6**	**97.9**	**99.4**	**95.6**	**99.9**	**99.3**	**96.8**
Average Recognition Rate						**98.01**			

The misclassified samples of the fuzzy classifier were analyzed carefully and it was found that some of these samples are badly written and may not be recognized by human. Therefore, subjective evaluation was conducted on the misclassified samples.

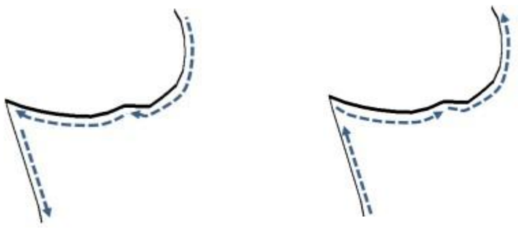
The images of the misclassified samples were printed randomly in a form and the form is given to 30 graduate/undergraduate students. Each student was told that these images are images of Arabic digits and he was asked to label the images as he perceives in the image. The results of this subjective evaluation are summarized in Table 3.

Table 3. The overall classification results of the human subjects.

Digit	Correctly Classified	Incorrectly Classified	Undetermined
1	1.67	42.22	56.11
2	86.89	0.00	13.11
3	80.14	4.93	14.93
4	83.33	10.48	6.19
5	45.83	4.17	50.00
6	64.19	13.87	21.94
7	0	10	90
8	67.33	11.33	21.33
9	64.85	16.36	18.79
Average	64.77	13.12	22.11

As shown in the table, the labels of the students are classified into three groups. The first group includes the number of answers that correctly labeled the digits.

The second group indicates the number of answers that labeled the digit incorrectly. The last group includes the number of responses where the students were unable to label the images. The subjective evaluation of the results shows that most students were not able to classify digits ‘1’ and ‘7’ correctly. Digits 5, 6, 8, and 9 were correctly labeled by no more than 68% of the subjective evaluators. Some samples of digits 2, 3, and 4 were not reflecting the proper directional way of writing these digits. The order of writing of these samples is different from the normal order of writing similar samples as shown in Figure 14. It is not easy to recognize these samples based on their directional features. However, the images of these samples can be recognized by humans easily.



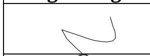
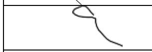
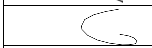
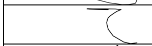
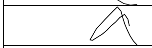
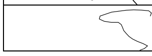
a) Regular order of writing digit ‘three’. b) Irregular order of writing digit ‘three’.

Figure 14. Digit three.

The overall results show that ≈ 65% of the samples were classified correctly by humans, ≈ 13% were incorrectly classified and the remaining ≈ 23% were undetermined. Therefore, ≈ 35% of the misclassified digits by our fuzzy classifier were not classified by humans.

In our work, we are using the same database used by Abdelazeem *et al.* [2]. However, they used offline features in their approach by converting the user’s strokes into a bitmap image while in our work we are using the online features. In addition, they used 30000 digits in their experiments whereas in our experiments we used all the samples of AOD database (32695 digits). Hence, the results of both techniques may not be comparable. We could not use their 30000 samples as we have no information of which 30000 samples were used. In one experiment when we removed unreadable samples by humans (125 samples) our combined recognition rate (stages 1 and 2) reached 99.55.

Table 4. Some misclassified samples of the fuzzy classifier.

Digit image	Digit Class	Classified Class
	٣	٧
	٤	٩
	٥	٢
	٦	٤
	٨	٩
	٩	٤

7. Conclusions

In this paper, a technique based on automatic fuzzy modeling for the automatic recognition of Arabic (Indian) online digits is presented. In this technique we automatically generate fuzzy models of the different digits using the segments’ directions of Arabic online digits of the training data. The models include the samples’ trend line of each digit and the digit’s model upper and lower envelops. The fuzzy intervals are generated automatically based on the analysis of the training samples at the digit segment level and not set manually at the digit level as in the previous works. In addition, we automatically generate weights for the different segments using the training samples. These weights are integrated in the fuzzy similarity estimate.

For the classification, a two stage approach is implemented where SVM is used in the first stage (using statistical features) and fuzzy-based approach using the automatically generated models in the second stage. The second stage has an integrated feedback verification step which verifies the recognized test sample label. If the first label does not pass validation (using other features) then the next label in the list is selected in the feedback loop otherwise it is selected as the recognized digit.

A database containing more than 30,000 Arabic online handwritten digits is used to test the proposed approach. An overall accuracy of 99.55% was achieved in the first stage (zero/nonzero) and the second stage (digits 1 to 9) achieved an accuracy of 98.01%. This result, based on using our fuzzy models and the proposed fuzzy structural classifier, proved to be better than using the SVM classifier with the directional features. The misclassified samples are evaluated subjectively and results indicate that humans were able to recognize ≈ 65% of these samples.

The authors are extending the application of this work to Arabic online text recognition.

Acknowledgment

This work was funded by the National Plan for Science, Technology and Innovation (MAARIFAH) King Abdul-Aziz City for Science and Technology through the Science and Technology Unit at King Fahd University of Petroleum and Minerals (KFUPM) the Kingdom of Saudi Arabia, project no. 11-INF2153-4.

References

[1] Abdelazeem S. and El-Sherif E., "Arabic Handwritten Digit Recognition," *International Journal of Document Analysis and Recognition*, vol. 11, no. 3, pp. 127-141, 2008.

[2] Abdelazeem S., El Meseery M., and Ahmed H., "Online Arabic Handwritten Digits Recognition," in Proceeding of International Conference on Frontiers in Handwriting Recognition, Bari, pp. 135-140, 2012.

[3] Abuhaiba S., Mahmoud S., and Green R., "Recognition of Handwritten Cursive Arabic characters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 6, pp. 664-672, 1994.

[4] Abuzaraida M. and Zeki A., "Feature Extraction Techniques of Online Handwriting Arabic Text Recognition," *in Proceeding of 5th International Conference on Information and Communication Technology for the Muslim World*, Rabat, pp. 1-7, 2013.

[5] AlKhateeb J. and Alseid M., "DBN-Based Learning For Arabic Handwritten Digit Recognition Using DCT Features," *in Proceeding of 6th International Conference on Computer Science and Information Technology*, Amman, pp. 222- 226, 2014.

[6] Al-Taani A. and Al-Haj S., "Recognition of On-line Arabic Handwritten Characters Using Structural Features," *Journal of Pattern Recognition Research*, vol. 5, no. 1, pp. 23-37, 2010.

[7] Beigi H., Nathan K., Clary G., and Subrahmonia J., "Challenges of Handwriting Recognition in Farsi, Arabic and Other Languages with Similar Writing Styles (An On- line Digit Recognizer)," *in Proceeding of the 2nd Annual Conference on Technological Advancements in Developing Countries*, New York, 1994.

[8] Douglas D. and Peucker T., "Algorithms for The Reduction of the Number of Points Required to Represent A Digitized Line or its Caricature," Cartographica: The International Journal for Geographic Information and Geovisualization vol. 10, no. 2, pp. 112-122, 1973.

[9] Duong A., Phan H., Le N., and Tran S., "A Hierarchical Approach for Handwritten Digit Recognition Using Sparse Autoencoder," *Issues and Challenges of Intelligent Systems and Computational Intelligence*, Springer International Publishing, 2014.

[10] Halawani K., "Arabic Online Text Recognition using Syntactic (Structural) Approach," M.S. Thesis, King Fahd University of Petroleum and Minerals, 2013.

[11] Impedovo S., Mangini F., and Barbuzzi D., "A Novel Prototype Generation Technique for Handwriting Digit Recognition," *Pattern Recognition*, vol. 47, no. 3, pp. 1002-1010, 2014.

[12] Jiang W., Sun Z., Yuan B., Zheng W., and Xu W., "User-Independent Online Handwritten Digit Recognition," *in Proceeding of International Conference on Machine Learning and Cybernetics*, Dalian, pp. 3359-3364, 2006.

[13] Kherallah M., Njah S., Alimi A., and Derbel N., "Recognition of On-Line Handwritten Digits by Neural Networks Using Circular and Beta Approaches," *in Proceeding of IEEE International Conference on Systems, Man and Cybernetics*, Tunisia, pp. 164-169, 2002.

[14] Kherallah M., Alimi A., and Derbel N., "On-Line Recognition of Handwritten digits by "Self Organisation Maps" Using Elliptical and Beta Representations," *in Proceeding of First International Congress on Signals, Circuits and Systems*, Monastir, pp. 503-507, 2004.

[15] Kherallah M., Haddad L., Alimi A., and Mitiche A., "On-Line Handwritten Digit Recognition Based on Trajectory and Velocity Modeling," *Pattern Recognition Letters*, vol. 29, no. 5, pp. 580-594, 2008.

[16] Khorashadizadeh S. and Latif A., "Arabic/Farsi Handwritten Digit Recognition using Histogram of Oriented Gradient and Chain Code Histogram," *The International Arab Journal of Information Technology*, vol. 13, no. 4, pp. 367-374, 2016.

[17] Mahmoud S., "Arabic (Indian) Handwritten Digits Recognition Using Gabor-Based Features," *in Proceeding of International Conference on Innovations in Information Technology*, Al Ain, pp. 683-687, 2008.

[18] Mahmoud S., "Recognition Of Writer-Independent Off-Line Handwritten Arabic (Indian) Numerals Using Hidden Markov Models," *Signal Processing*, vol. 88, no. 4, pp. 844-857, 2008.

[19] Mahmoud S. and Al-Khatib W., "Recognition Of Arabic (Indian) Bank Check Digits Using Log-Gabor Filters," *Applied Intelligence*, vol. 35, no. 3, pp. 445-456, 2011.

[20] Niu X. and Suen C., "A Novel Hybrid CNN-SVM Classifier For Recognizing Handwritten Digits," *Pattern Recognition*, vol. 45, no. 4, pp. 1318-1325, 2012.

[21] Parvez M. and Mahmoud S., "Offline Arabic

Handwritten Text Recognition: A Survey," *ACM Computing Surveys*, vol. 45, no. 2, pp. 1-35, 2013.

[22] Parvez M. and Mahmoud S., "Arabic Handwriting Recognition Using Structural and Syntactic Pattern Attributes," *Pattern Recognition*, vol. 46, no. 1, pp. 141-154, 2013.

[23] Plamondon R. and Srihari S., "Online and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63-84, 2000.

[24] Subhashini P. and Prasad V., "Recognition of Handwritten Digits Using Rbf Neural Network," *International Journal of Research in Engineering and Technology*, vol. 2, no. 3, pp. 393-397, 2013.

[25] Teredesai A., Ratzlaff E., Subrahmonia J., and Govindaraju V., "On-Line Digit Recognition Using Off-Line Features," *in Proceeding of Indian Conference on Computer Vision, Graphics and Image Processing*, Ahmadabad, 2002.

Dhiaa Musleh received his B.Sc. in Computer Science from Mosul University, Iraq, and then he joined the faculty of applied science at Taiz University as a teaching assistance. He received his.S. in Computer Science from King Fahd University of Petroleum and Minerals (KFUPM),Saudi Arabia; he is currently a PhD candidate at the Information and Computer Science Department at KFUPM. His research interests include pattern recognition, Arabic document analysis and recognition.

Khaldoun Halawani received his B.Sc. degree in Information Technology form Palestine Polytechnic University (PPU), Palestine. And his M.Sc. degree in Information and Computer Sciences from King Fahd University of Petroleum and Minerals (KFUPM), Kingdome of Saudi Arabia, in 2009 and 2013, respectively. His research work and interests are on machine learning, pattern recognition, multimedia processing and artificial intelligence. He worked for four years as research assistance in PPU and KFUPM.

Sabri Mahmoud is a Professor of computer Science in the ICS Department, KFUPM. His research interests include Arabic Document Analysis and Recognition, Arabic NLP, Image Analysis and applications of Pattern Recognition. Dr. Mahmoud is a life senior member of IEEE. He published over 80 papers in refereed journals and conference proceedings.