JOURNAL OF PATTERN RECOGNITION RESEARCH 1 (2010) 23-37

Received January 15, 2010. Accepted July 2, 2010.

Recognition of On-line Arabic Handwritten Characters Using Structural Features

Ahmad T. Al-Taani *ahmadta@yu.edu.jo*
Department of Computer Sciences Yarmouk University, Irbid, Jordan

Saeed Al-Haj *shaj@cs.nmsu.edu*
Department of Computer Science, New Mexico State University, New Mexico, USA

Abstract

In this study, an efficient approach for the recognition of on-line Arabic handwritten characters is presented. The approach is based on structural features and decision tree learning techniques. The proposed approach consists of three phases: First, the user writes the character on a special window on the screen, and then the coordinates of the pixels forming the character is captured and stored in a special array. Second, a bounding box of 5x5 is drawn around the character, and five features are extracted from the character that used in step three for the recognition of the character through the use of a decision tree learning techniques. The proposed approach is tested on a set of 1400 different characters written by ten users. Each user wrote the 28 Arabic characters five times in order to get different writing variations. Experiment results showed the effectiveness of the novel approach for recognizing handwritten Arabic characters.

Keywords: Character Recognition, Feature Extraction, Structural Primitives, Document Processing, Primitives Selection.

WWW.JPRR.ORG

1. Introduction

The main problem encountered when dealing with handwritten Arabic characters is that characters written by different persons representing the same character are not identical but can vary in both size and shape. The fast variation in personal writing styles and differences in one person's writing style depending on the context is another problem encountered when trying to recognize Arabic handwritten characters. In addition, the mood of the writer and the writing situation can have an effect on writing styles.

Considerable work has been undertaken in the area of Arabic character recognition but with limited success, this is due to the nature of Arabic characters and to the problems mentioned above. Arabic alphabet consists of 28 basic characters. Some characters may have different shapes depending on there position within a word (beginning, middle, end) and different size (height and width). In addition, sixteen of the Arabic characters have a single dot, or double, or triple dots, or zigzag, which are used to distinguish between characters having identical main parts. A review of the Arabic character recognition research has shown that techniques developed for the recognition of Latin text are not directly applicable to the recognition of Arabic text [2].

In this study, we introduce a novel approach to the recognition of Arabic handwritten characters using structural features and decision trees. Each character has different features that distinguish it from other characters. These features include: number of segments, left-right density ratio, bottom-up density ratio, and other features. The proposed system consists of three main phases. First, while a user writes a character on a special window on the screen, the (x, y) coordinates of the pixels forming the character are captured and stored in an array. Second, a bounding box is drawn around the character and then features that give structural information of the character are extracted. Then, these features are used as input to the decision tree to recognize the character in question.

© 2010 JPRR. All rights reserved. Permissions to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or to republish, requires a fee and/or special permission from JPRR.

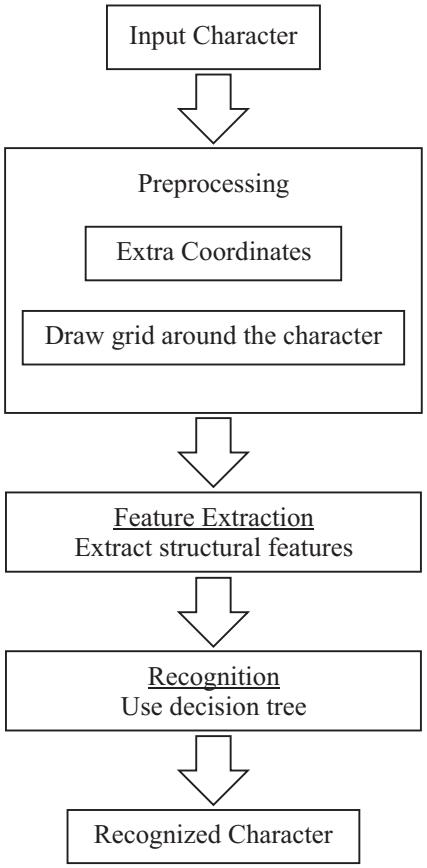

Fig. 1: A block diagram of the proposed system.

The block diagram of the proposed system is shown in Fig. 1.

The rest of the paper is organized as follows. In Section 2, an overview of the Arabic language is presented. Section 3 presents some related work. An overview of the proposed approach including the feature extraction algorithms are discussed in Section 4. Experimental results and discussions are presented in Section 5. Conclusions and suggested future work are presented in Section 6.

2. Overview of the Arabic Language

Arabic alphabet consists of 28 characters. Words are written in horizontal lines from right to left. The Arabic character set is shown in Table 1.

Each character has two to four different forms that depend on its position in the word [2], see Fig. 2.

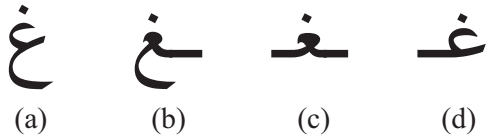

Fig. 2: Different forms of "GHYN غ" character. (a) Single form, (b) Ending form, (c) Middle form, (d) Beginning form.

Table 1: Arabic Alphabet and their forms at different positions in the word

Letter	Single	Ending	Middle	Beginning
Alef	ا	ـا	ـا	ا
Baa	ب	ـب	ـبـ	بـ
Taa	ت	ـت	ـتـ	تـ
Thaa	ث	ـث	ـثـ	ثـ
Jeem	ج	ـج	ـجـ	جـ
Haa	ح	ـح	ـحـ	حـ
Khaa	خ	ـخ	ـخـ	خـ
Dal	د	ـد	ـد	د
Thal	ذ	ـذ	ـذ	ذ
Raa	ر	ـر	ـر	ر
Zai	ز	ـز	ـز	ز
Seen	س	ـس	ـسـ	سـ
Sheen	ش	ـش	ـشـ	شـ
Sadd	ص	ـص	ـصـ	صـ
Dadd	ض	ـض	ـضـ	ضـ
Tah	ط	ـط	ـطـ	طـ
Thah	ظ	ـظ	ـظـ	ظـ
Ayn	ع	ـع	ـعـ	عـ
Ghyn	غ	ـغ	ـغـ	غـ
Faa	ف	ـف	ـفـ	فـ
Qaf	ق	ـق	ـقـ	قـ
Kaf	ك	ـك	ـكـ	كـ
Lam	ل	ـل	ـلـ	لـ
Meem	م	ـم	ـمـ	مـ
Noon	ن	ـن	ـنـ	نـ
Ha	ه	ـه	ـهـ	هـ
Waw	و	ـو	ـو	و
Yaa	ي	ـي	ـي	ي

Characteristics of Arabic writing include:

1. Arabic text, both handwritten and printed, is cursive. The letters are joined together along a writing line. This is similar to English handwriting, which is also cursive, but in which the characters are easier to separate.

2. In contrast to English text, Arabic is written right to left, rather than left to right. This is perhaps more significant for a human reader rather than a computer, since the computer can simply rotate the images.

3. More importantly from the point of view of automated recognition, Arabic contains dots and other small marks that can change the meaning of a word, and need to be taken into account by any computerized recognition system.

4. The shapes of the letters differ depending on whereabouts in the word they are found. The same letter at the beginning and end of a word can have a completely different appearance as shown in Figure 2. Along with the dots and other marks representing vowels, this makes the effective size of the alphabet about four times the initial characters set.

Automatic recognition of Arabic texts is complicated by several properties of the Arabic script:

- Connectivity of symbols
- Cursive nature of the language
- Similarity of groups of symbols
- Highly variable widths
- Overlapping between characters

The Arabic alphabet is represented numerically by a standard communication interchange code approved by the Arab Standard and Metrology Organization (ASMO) [3]. Similar to the American Standard Code for Information Interchange (ASCII), each character in the ASMO code is represented by one byte. An English letter has two possible shapes, capital and small. The ASCII code provides separate representations for both of these shapes, whereas an Arabic letter has only one representation in the ASMO table. This is not to say, however, that an Arabic letter has only one shape. On the contrary, an Arabic letter might have up to four different shapes, depending on its relative position in the text.

There are two approaches [4] to tackle the problem of cursiveness in Arabic script: the global approach and the analytical approach. The global approach treats the word as whole. Features here are extracted from the un-segmented word and compared to a model. The analytical approach decomposes the word into smaller units or primary and secondary strokes. This paper deals only with isolated Arabic letters.

3. Related Work

For the past few decades, intensive research has been done to solve the problem of Arabic character recognition. Various approaches have been proposed to deal with this problem. Challenging problems are being encountered and solutions to these are targeted in various ways to improve accuracy and efficiency.

Khorsheed [4] presented a method for the recognition of on-line handwritten Arabic script based on hidden Markov models and structural features.

El-Sheikh *et al.* [5] [6] proposed two algorithms to recognize Arabic handwritten characters and cursive words. The first 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.* [7] 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.

El-Wakil *et al.* [8] proposed a method for the recognition of isolated handwritten Arabic characters drawn on a graphic tablet. Two types of features are extracted from the characters. Features that are independent of the writer style are represented as a list of integer values, while those that are subjected to more variations are represented using a Freeman-like chain code.

El-Dabi *et al.* [9] presented a recognition system for typed Arabic text, which involves a statistical approach for character recognition.

Sabri Mohmoud [10] 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.

Amin *et al.* [11] [12] presented a technique for the recognition of Arabic words and Chinese characters using the C4.5 machine learning system. The technique is divided into three major steps; digitization, pre-processing feature extraction, and classification.

Cheung *et al.* [13] proposed an Arabic OCR system, which uses a recognition-based segmentation technique to overcome the classical segmentation problems. There is also a feedback loop to control the combination of character fragments for recognition.

Kharma *et al.* [4] proposed the use of mapping for the recognition of on-line handwritten characters. This mapping produces the same output pattern regardless of the orientation, position, and size of the input pattern.

Mezghani *et al.* [15] investigated a method for on-line Arabic characters recognition. This method is based on the use of Kohonen maps and their corresponding confusion matrices which serve to prune them of error-causing nodes, and to combine them consequently.

Ayman *et al.* [16] proposed a recognition system for handwritten Arabic characters using neural network classifier. The proposed system is trained on 600 images and tested on 250 images. The classification rate for the system reached 90%.

Benouareth *et al.* [17] described an offline Arabic handwritten word recognition system based on segmentation-free approach and hidden Markov models. Several experiments are performed using the IFN/ENIT benchmark database.

4. Materials and Methods

The goal of this work is to develop a system that recognizes on-line Arabic handwritten characters that can be adapted to the demands of hand-held and digital tablet applications. Features needed for the recognition process include: number of segments, left-right density ratio, bottom-up density ratio and others. Decision trees are then used to classify the characters based on the features that were extracted from the input character.

4.1 Tracing the Character

After writing a character on the screen, we get a sequence of points representing the $x - y$ coordinates of the pixels forming the character. The tracing process must be done in parallel with the writing process (online), so we can keep track of the input character.

The outputs of this step are number of segments and a string for each segment that contains x-y coordinates of the input character. Every mouse click is considered as one segment, for example, the letter ”SEEN” must be written by one mouse click and drag Fig. 3. The proposed system will not recognize the letter in Fig. 3(a) correctly as ”SEEN”, because the letter ”SEEN” is classified as one-segment letter. The letter in Fig. 3(a) will be stored in two separate lists and to recognize ”SEEN” letter we deal with the first list only.

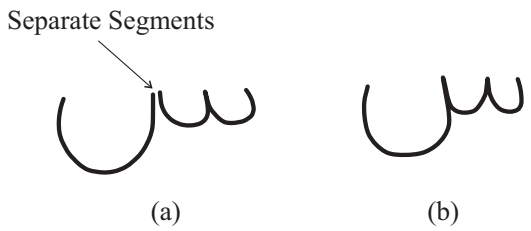

Fig. 3: (a) Two-segment SEEN letter (b) One-segment SEEN letter.

4.2 Placing the grid

We draw a 5x5 grid around the character in order to extract features needed for the recognition step. An example of such features is the location of the dot. Fig. 4 shows two examples of Arabic letters ”JEEM” and ”KHAA”. The only difference between ”JEEM” and ”KHAA” letters is the location of the dot; ”KHAA” letter has a dot above the main shape (in the first layer), while ”JEEM” has a dot location in the middle layer.

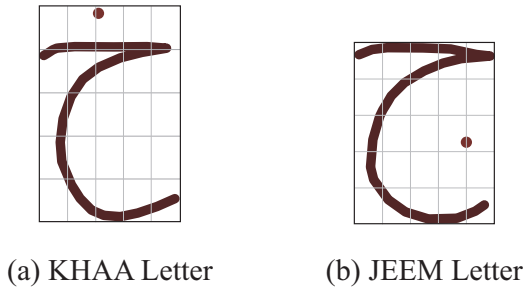

Fig. 4: The effect of the dot location on character recognition.

4.3 Feature Extraction

4.3.1 Number of Segments (NS)

The most important feature used in this work is the number of segments. By segment we mean the separate letter component that must be written without lifting the pen. Fig. 5 shows the Letter ”THAH” that has three segments. The use of the number of segments as an attribute in the decision tree allows for classifying Arabic letters into four classes Fig. 6.

As we can see from Fig. 6, one-segment and two-segment letters need more attention in the recognition phase, since the number of segments is not sufficient for the recognition, so we need other features for the recognition task.

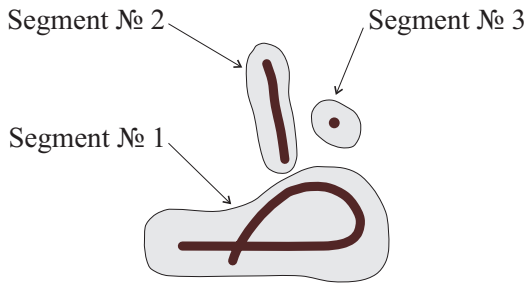

Fig. 5: Three-segment Letter (THAH Letter).

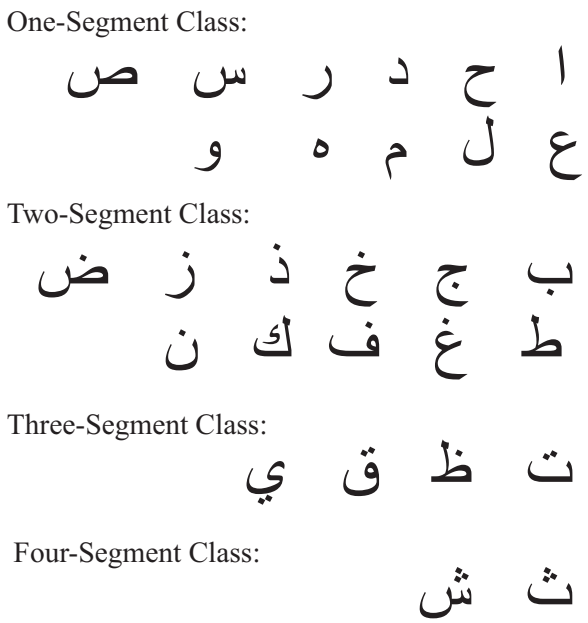

Fig. 6: Classification of Arabic characters based on number of segments.

4.3.2 Cross-Points (Loop)

Another feature that is useful is whether the written letter contains a loop or not. Nine of the Arabic letters contain loops Fig. 7. We have developed an algorithm to detect a loop in a written letter.

Fig. 7: Arabic letters that have a loop.

4.3.3 Sharp Edges (ShE)

Sharp edge detection is the most difficult feature to be extracted. Sharp edge is similar to 20-40 degree angle. To illustrate this feature see Fig. 8 that shows some letters that have sharp edges.

There are two types of sharp edges with regard to the direction of the edge. In Figure 8 (a) the letter ”AYN” is a y-direction sharp edge type while in Fig. 8(b) the letter ”SAAD” is an x-direction type. Y-direction sharp edge is detected during the movement of the pen from upward to downward, and the x-direction is detected when a sharp turning point exists with the movement from right to Left.

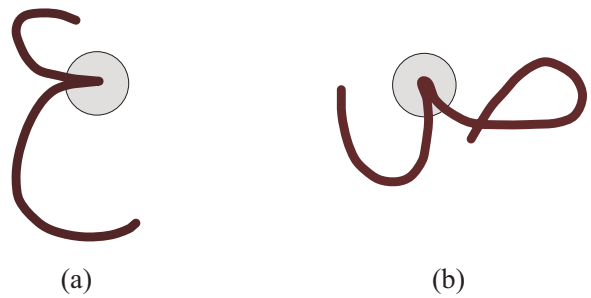

Fig. 8: Letters containing sharp edge: (a) y-direction, (b) x-direction sharp edges.

4.3.4 Secondary Segments (SS)

Any part or component that is written after the primary part is considered as a secondary segment. Secondary segments are stored in lists numbered two to four. There are three types of secondary segments: dot, line, and curve, these types are shown in Fig. 9.

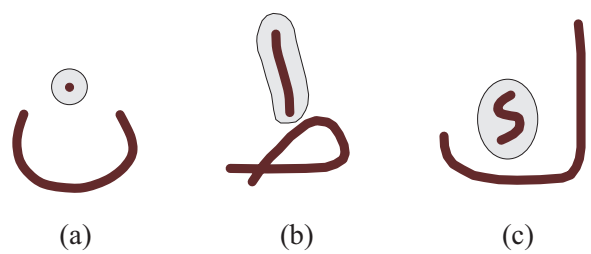

Fig. 9: Types of secondary segments: (a) Dot, (b) Line, (c) Curve.

4.3.5 Similarity of Secondary Segments (SSS)

This feature is useful when we deal with the letters that belong to the three-segment class. Two values can be derived from this feature: existence or absence of similarity. Similarity detection depends on the values stored in lists numbered two and three. In case we have similar secondary segments, then we have two dots.

4.3.6 Bottom-Up (BUDR) and Left-Right (LRDR) Density Ratios

Many letters have a noticeable property that distribution of written character on the grid is not equal to the ratio between the pixels of the written letter in the first two rows and the pixels of the written letter the last two rows; or between the first two columns and the last two columns. The first case is called bottom-up ratio while the second one is called left-right ratio. Fig. 10 shows an example of this feature.

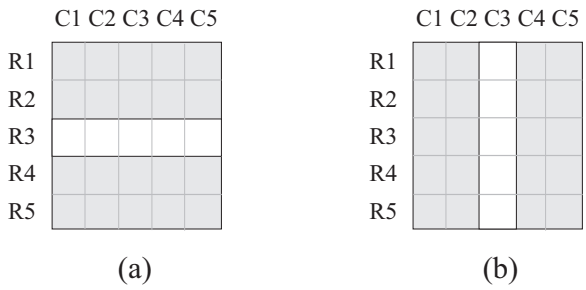

Fig. 10: Density Ratio Calculations (a) BUDR (b) LRDR

Every row or column contains five equal size cells; to calculate bottom-up or left-right density ratios we use the following formulae:

$$BUDR = \#\ pixels\ in\ (R1 + R2)\ /\ \#\ pixels\ in\ (R4 + R5) \tag{1}$$
$$LRDR = \#\ pixels\ in\ (C1 + C2)\ /\ \#\ pixels\ in\ (C4 + C5)$$

Every pixel corresponds to one element in the lists that we used to store the written character after applying the tracing module.

To get control over these features, we defined two constant values as thresholds T1 and T2 (T1 is greater than T2). The values of these thresholds are determined using a trial-error method. For bottom-up density, if the ratio is greater than T1 then we say that the letter is up-oriented and if the ratio is less than T2, we say that the letter is bottom-oriented. If the ratio between T1 and T2 we say that the character has neutral behavior for this feature. We apply the same definitions on left-right density ratio.

Fig. 11 shows some letters that have up-oriented, bottom-oriented, left-oriented, or right-oriented behavior. Some letters can have combination of bottom-up and left-right density ratios.

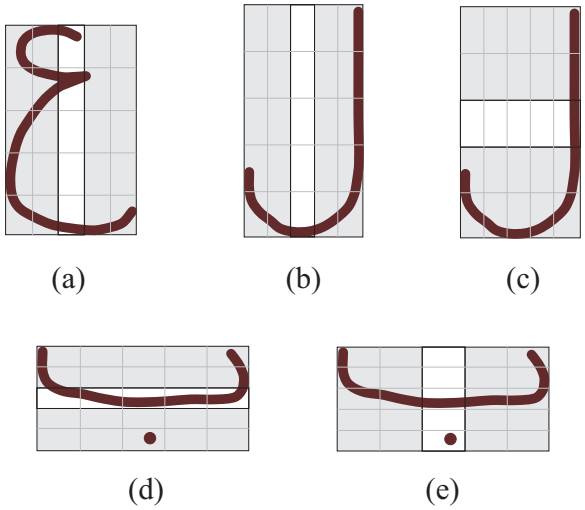

Fig. 11: Density orientation (a) left-oriented, (b) right-oriented, (c) bottom-oriented, (d) up-oriented, (e) neutral left-right orientation.

4.3.7 Horizontal–Vertical Orientation (HVO)

Another helpful feature is the horizontal-vertical orientation. This feature depends on the range of x and y coordinates, HVO ratio can be defined as follows:

$$HVO = (y_max - y_min)/(x_max - x_min) \tag{2}$$

Because of different writing styles we define two threshold values S1 and S2 (S1 is greater than S2). The following production rules are used for the decision:

$$\begin{aligned} &\textit{If} \quad \textit{HVO} > \textit{S1} \quad \textit{then the letter is horizontal-oriented.} \\ &\textit{If} \quad \textit{HVO} < \textit{S2} \quad \textit{then the letter is vertical-oriented.} \\ &\textit{If} \quad \textit{S1} > \textit{HVO} > \textit{S2} \quad \textit{then the letter has neutral orientation.} \end{aligned} \tag{3}$$

In other words, this feature gives us a hint of the grid shape. The grid shape may be a square, vertical rectangle, or a horizontal rectangle. We can notice the feature effect as it appears Fig. 11. Fig. 11 (a-c) shows horizontal rectangle and Fig. 11 (d-e) shows vertical rectangle.

4.4 Letters Attributes

We introduced letters attributes based on the number of classification segments. The values assigned to the attributes were determined by trial and error. According to the number of segments, we have classified the Arabic letters into four classes:

1. One-segment letters.
2. Two-segment letters.
3. Three-segment letters.
4. Four-segment letters.

To distinguish between these classes, one should verify the existence or absence of a sharp edge. If there is a sharp edge then the letter is ”SHEEN”, otherwise the letter is ”THAA”. Tables 2-5 show the letters according to this classification. The following abbreviations are used in these tables: *B: Bottom, U: Up, L: Left, R: Right, V: Vertical, H: Horizontal and DC: Don’t care.*

4.5 The Classification Phase

The features extracted in the previous phase are used in the classification phase. The decision tree learning method is used to recognize the input Arabic letters. The decision tree used is this work is shown in Fig. 12.

Because there are different attributes attached to different classes of characters, the need to split the tree to four sub-trees is obvious. Every sub-tree deals with a specific class of letters. The values of the attributes determine the branch that should be selected. If the written character has a value that isn’t labeled on any branch of the tested attribute, then the process will be stopped and the recognition system fails to recognize this written character.

Table 2: Attributes of one-segment letters

Letter	Loop	H/V	BUD	LRD	ShE
ا	No	H	U	DC	No
ح	No	V	U/DC	DC	Yes/Y-type
د	No	V	B	R	No
ر	No	V	U/DC	DC	No
س	No	V	U/DC	DC	Yes/X-type
ص	Yes	V	U/DC	DC	Yes/X-type
ع	No	H	U/DC	DC	Yes/Y-type
ل	No	H	B	R	No
م	Yes	H	U/DC	DC	No
ه	Yes	V	U/DC	DC	No
و	Yes	V	U/DC	R	No

Table 3: Attributes of Two-segment characters

Letter	Loop	H/V	BUD	LRD	SS	SS Location
ب	No	V	U	DC	Dot	Bottom
ج	No	H	U	DC	Dot	Middle
خ	No	H	U	DC	Dot	Up
ذ	No	H	B	R	Dot	Up
ز	No	H	B	R	Dot	Up
ض	Yes	V	Neutral	DC	Dot	Up
ط	Yes	V	B	DC	Line	—
غ	No	V	Neutral	DC	Dot	Up
ف	Yes	V	B	R	Dot	Up
ك	No	V	B	R	Curv	—
ن	No	V	B	DC	Dot	Up

5. Results and Discussions

Ten different users tested the proposed method. Each user wrote every letter of the Arabic letters five times, i.e. every letter was written fifty times and the test set size used in the experiments is 1400 different letters. Experimental results are presented in Table 6.

Experimental results showed that the proposed method gave a recognition rate of about 75.3% for all letters, but it did not perform well on the letters that contains sharp edges:

”ج, ح, خ, ع, غ”,

in which it gave an average performance of about 48.8% for these letters. The system gave an average performance of about 85.3% when we exclude the letters with sharp edges from the calcu-

Table 4: Attributes of Three-segment characters

Letter	Loop	BUDR	SSS	SS Location
ت	No	B	Yes	Up
ظ	Yes	B	No	—
ق	Yes	B	Yes	Up
ي	No	U	Yes	Bottom

Table 5: Attributes of Four-segment Characters

Letter	ShE
ث	No
ش	Yes

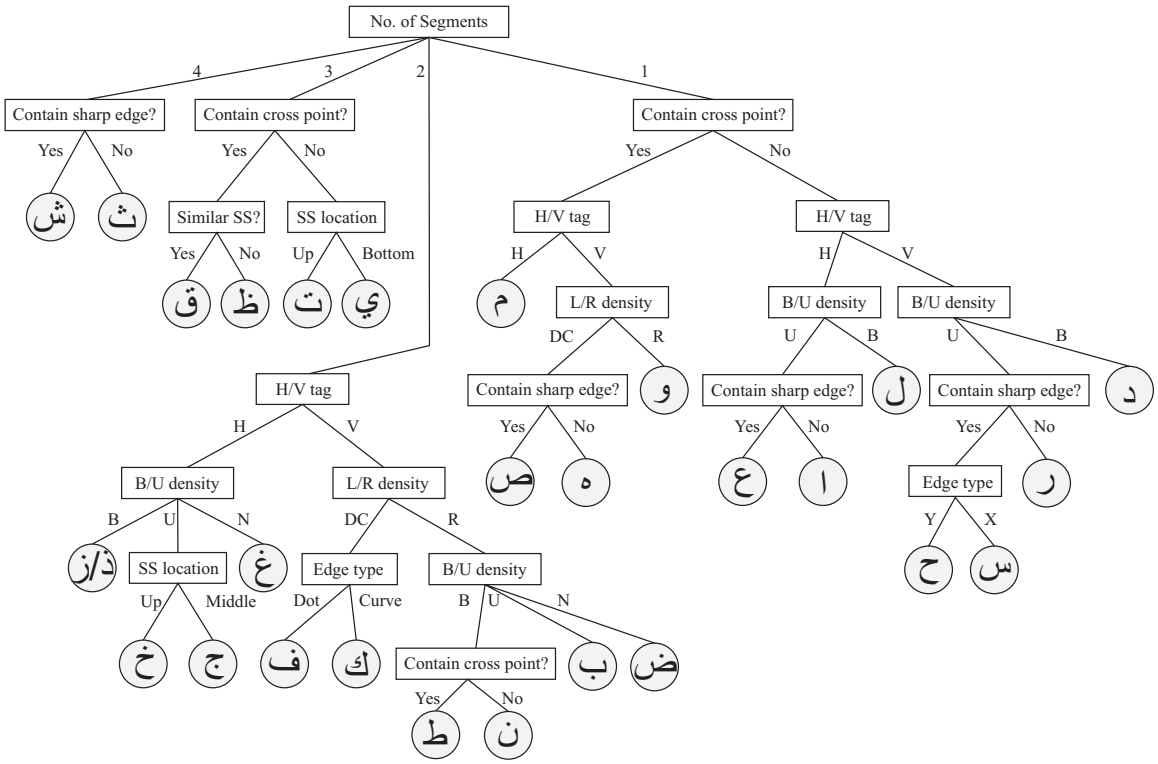

Fig. 12: Decision tree used in the recognition phase.

lations.

From the testing process we noticed the following important remarks:

1. The drawing speed may affect the recognition process. If the user draws very quickly, the system might not capture all the input pixels representing the letter, i.e. the drawing must be connected, so the user has to draw the letter as one connected line.

Table 6: Recognition rates for Arabic letters in the system.

Character	Percentage	Character	Percentage
أ	93%	ض	80%
ب	90%	ط	85%
ت	88%	ظ	75%
ث	90%	ع	**50%**
ج	**50%**	غ	**53%**
ح	**48%**	ف	80%
خ	**43%**	ق	83%
د	90%	ك	90%
ذ	88%	ل	93%
ر	88%	م	83%
ز	85%	ن	93%
س	83%	ه	70%
ش	85%	و	83%
ص	78%	ي	85%

2. The accuracy of the system depends on many factors like whether there is noise in the test data, if the letter is poorly written, deliberately written in some strange and unusual way, or with zig-zag line segments. We should also take into account that the writing process itself is subjective and depends on the person writing style. If the test data are carefully selected then the system could give higher accuracy rate. The results achieved are very promising as compared to the previous works.

3. The proposed system works only on Arabic isolated letters.

Despite these factors the proposed approach has the advantage of using structural features together with a decision tree for the recognition process. Experimental results show the usefulness of the structural features in achieving good recognition results since these features are used by people visually to recognize the letters. Also, we used a decision tree since it works in the same manner as the human information processing system does. This reflects one of our main objectives in this work, to design an intelligent agent which behaves rationally like humans.

6. Conclusions and Future work

We have presented a novel approach to the recognition of Arabic letters based on novel features. Although, there are some challenges with some letters, the overall recognition rate is acceptable.

The proposed method can easily be applied to any application that requires Arabic handwritten character recognition, regardless of its computing power. This is due to low computational requirement. Thus, the proposed algorithm can be implemented on any type of hardware or software platform, such as PDA's platform. The method can also be applied to an off-line system if the coordinate data sent into the system can be sent in as a time ordered sequence of data.

Future work will consider increasing the efficiency of the proposed approach especially for the letters that were not recognized well by the system.

These letters contain sharp edges:

”ج, ح, خ, ع, غ”.

References

[1] A.K. Jain, R.P.W. Duin and J. Mao, ”Statistical pattern recognition: a review,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, 2000, pp. 4-37.

[2] Adnan Amin, ”Off-Line Arabic Character Recognition: The State of The Art”, Pattern Recognition, Vol. 31, No. 5, 1998, pp. 517-530.

[3] Karim Hadjar and Rolf Ingold, ”Arabic Newspaper Page Segmentation”, proceeding of the seventh international conference on document analysis and recognition, Vol. 2, 2003, pp. 895 - 899.

[4] M.S. Khorsheed, ”Recognizing handwritten Arabic manuscripts using a single hidden Markov model”, Pattern Recognition Letters, Vol. 24, 2003, pp. 2235-2242.

[5] 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.

[6] T. S. El-Sheikh and Ramez M. Guindi, ”Computer recognition of Arabic cursive scripts”, Pattern Recognition, Vol. 21, no. 4, 1988, pp. 293-302.

[7] 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.

[8] Mohamed S. El-Wakil and Amin A. Shoukry, ”On-line recognition of handwritten isolated arabic characters”, Pattern Recognition, Vol. 22, no. 2, 1989, pp. 97-105.

[9] Sherif El-Dabi, Refat Ramsis and Aladin Kamel, ”Arabic character recognition system: A statistical approach for recognizing cursivetypewritten text”, Pattern Recognition, Vol. 23, no. 5, 1990, pp. 485-495.

[10] Sabri A. Mahmoud, ”Arabic character recognition using Fourier descriptors and character contour encoding”, Pattern Recognition, Vol. 27, no. 6, 1994, pp. 815-824.

[11] Adnan Amin, ”Recognition of printed Arabic text based on global features and decision tree learning techniques”, Pattern Recognition, Vol. 33, 2000, pp. 1309 -1323.

[12] Adnan Amin and S. Singh, ”Recognition of Hand-printed Chinese Characters using Decision Trees/Machine Learning C4.5 System”, Pattern Analysis and Applications, Vol. 1, no. 2, 1998, pp. 130-141.

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

[14] Nawwaf Kharma and Rabab K. Ward, ”A novel invariant mapping applied to hand-written Arabic character recognition”, Pattern Recognition, Vol. 34, 2001, pp. 2115 - 2120.

[15] Neila Mezghani, Mohamed Cheriet, and Amar Mitiche, ”Combination of Pruned Kohonen Maps for On-line Arabic Characters Recognition ”, In proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003), pp. 900 - 904.

[16] Ayman J. Alnsour and Laheeb M. Alzoubady, ”Arabic Handwritten Characters Recognized by Neocognitron Artificial Neural Network”, University of Sharjah Journal of Pure & Applied Sciences, Vol. 3, No. 2, 2006.

[17] A. Benouareth, A. Ennaji and M. Sellami, ”Arabic Handwritten Word Recognition Using HMMs with Explicit State Duration”, Journal on Advances in Signal Processing, Volume 2008, pp. 1-13, 2008.