DOI:10.24996/ijs.2018.59.2A.14

ISSN: 0067-2904

Printed Arabic Characters Recognition Based on Minimum Distance Classifier Technique

Nabeel Mubarak Mirza

Department of Physics, Faculty of Education, University of Al- Mustansiriyah, Baghdad, Iraq.

Abstract

The printed Arabic character recognition are faced numerous challenges due to its character body which are changed depending on its position in any sentence (at beginning or in the middle or in the end of the word). This paper portrays recognition strategies. These strategies depend on new pre-processing processes, extraction the structural and numerical features to build databases for printed alphabetical Arabic characters. The database information that obtained from features extracted was applied in recognition stage. Minimum Distance Classifier technique (MDC) was used to classify and train the classes of characters. The procedure of one character against all characters (OAA) was used in determining the rate of recognition. The suggested approaches have yielded unique and encouraging results in terms of accuracy in which the recognition rate reached to 97.28%. These approaches are faster and more efficient than other methods.

Keywords: printed Arabic characters, minimum distance classifier, recognition rate.

التعرف على الحروف العربية المطبوعة بالإعتماد على تقنية مصنف المسافة الصغرى

نبيل مبارك ميرزا

قسم الفيزياء، كلية التربية، الجامعة المستنصرية، بغداد، العراق.

الخلاصة

إن التعرف على الحروف العربية المطبوعة يواجه تحديات عديدة بسبب هيئة الحرف التي تتغير تبعا لموقعها في أي جملة (في بداية أو في منتصف أو في نهاية الكلمة). هذا البحث يصف إستراتيجيات التعرف. وتعتمد هذه الإستراتيجيات على عمليات جديدة قبل المعالجة، وإستخراج الخصائص الهيكلية والعددية للأحرف العربية الأبجدية المطبوعة لبناء قواعد بيانات. تم تطبيق المعلومات المخزونة في قاعدة البيانات والتي تم الحصول عليها من الميزات المستخرجة في عملية التعرف, حيث أُستخدمت تقنية مصنف المسافة الصغرى (MDC) لتصنيف وتدريب فئات الحروف. وقد تم إستخدام إجراء حرف واحد ضد جميع الأحرف (OAA) في تحديد نسبة التعرف. هذه الأساليب أسرع وأكثر كفاءة من الأساليب الأخرى إذ أعطت نتائج مشجعة وكبيرة من حيث الدقة وقد بلغت نسبة التعرف إلى 97.28٪.

1. Introduction

It is clear for every one that people communicate with each other through the languages. The Arabic language is one of the important languages in communication between people. This language is utilized by more than 422 million persons.

After many years of computers invention, it became clear that the computer can perform non-mathematical operations beside mathematical tasks that enable us to get advantage to take decisions to

*Email: nabeel.mirza@uomustansiriyah.edu.iq distinguish, recognize, and read between letters precisely, this lead to new field that classified as artificial intelligence dealing and processing natural languages.

OCR which stands for Optical Character Recognition is a main process that can classify and deal with optical patterns that include digital image [1].

Character recognition can be classified into two main categories which are on-line and off-line. The off-line subcategory also is divided into two fields:

- Magnetic Character Recognition (MCR)
- Optical Character Recognition (OCR)

OCR is separated into two types: handwritten character and printed character [2]. In this research, the focus is on the off-line recognition which leads to printed character to explore or recognize Arabic characters whether the positions of the letters are connected or isolated types letters.

The direction of written Arabic words is from right to left. The Arabic language contains 28 letters (3 vowels and 25 consonants). The main challenges which face the process of recognizing written characters of Arabic texts are the difference in the form of the character depending on letter position in the word (starting, in the mid, and the end of the word). The same latter can be written in different ways (i.e. disconnected from the main word or connected to it). In addition, there are a lot of Arabic letters have dots; this makes the recognition more difficult.

2. Related Work

The research on Arabic character recognition is begun in 1980s, since then there is various recognition methods have been proposed to avoid the above problems. Where, different methodology systems have been applied in dealing with such challenges to improve effectiveness and exactness; besides the using of modern techniques for off-line handwritten recognition used recently by Dehghan et al. [4], Amir et al. [5], Amrouch et al. [6], and Ahlam et al. [7].

Concerning printed Arabic character recognition, many of applied techniques can be seen in Talaat and Ramez [8] who proposed a method for the recognition of isolated printed letters based on set of Fourier descriptors in which obtained from the coordinate sequences of the contour points. These set of descriptors used especially for isolated Arabic characters based on three techniques. The first technique was minimum distance a multi-category, the second was a pairwise classifier and the last technique was hierarchical structure classifier.

Bushofa and Spann [9] mentioned that Arabic words classified into two types: characters and secondaries; the later type was removed by using new algorithm. They also applied the skeletonized character for classification by using decision tree; this method resulted in a recognition rate to 97.23%. Nadia et al. [10] have proposed a combined method for printed Arabic characters recognition through applying special algorithm depending on wavelet transform for features extraction and a neuro-fuzzy method for classification. The results of such combinations lead the rate recognition to reach 95.64%. Nadia et al. [11] proposed a hybrid technique to recognize printed Arabic characters based on two transforms (Wavelet and Hough) for features extraction. They have also applied hidden Markov for classification. The recognition rate was varied from 94% to 98%. Sabri and Ashraf [12] proposed the Fast Hartley Transform (FHT) algorithm for features selection based on Hartley descriptors technique that applied on printed Arabic characters in which the rate of recognition was 97.3%.

Also, there are more attempts for on-line handwritten recognition which can be found in Mohamed and Amin [13], Neila et al. [14], Jakob et al. [15], Majid et al. [16], Redouane and Abdelkader [17], and Sherif and Hany [18].

The motivation of this work is to create and extend a dependable off-line optical character recognition system for printed alphabetical Arabic character. The databases established are consist of 16 features concerning printed Arabic characters excluded dotted characters, Table-1 shows samples of printed Arabic characters.

Table -1 Samples of Arabic characters.

لـ	مـ	كـ	صـ	سـ	ط	حـ	هـ	عـ	و	ر	ا	د	لا
Lam	**Mem**	**Kaf**	**Sad**	**Seen**	**Tta**	**Hha**	**Ha**	**Ain**	**Waw**	**Raa**	**Alif**	**Dal**	**Lam Alif**

In section II, the theoretical background of the proposed method is presented. Section III is highlights the method that dealing with pre-processing to build databases for printed alphabetical

Arabic characters, the features extraction and character recognition. Section IV presents the numerical results and discussion, and finally section V is the conclusion.

3. The Related Theories of The Proposed Method

This section gives a brief description of the theoretical aspect of the proposed algorithm.

MDC method, which considered fast and simple comparing with other classifiers methods, is used in many subjects of pattern recognition [3]. This method categorizes unknown patterns into their original classes.

In case writing down the description of MDC, there should be the definition of (x) by using the equation 1, in which represents unknown pattern. The MDC is defined as:

$$X \in w_i, \ if \ d(x, Z_i) = \min_j d(X, Z_j) \qquad (1)$$

Concerning the equation (1): Z_i is a prototype of category w_i, $i = 1, \ldots, n$. Where (x and z) are m-dimensional vectors in the feature space and (n) is the number of categories, and m is the number of dimensions of the feature space.

Where d (·) is the Euclidean distance function,

$$d(x, Z_i) = \sqrt{\sum_{i=1}^{m} (x_i - Z_{il})} \qquad (2)$$

There are several edges detection methods such as Canny, Sobel, Roberts, Prewitt and Laplace [19]. However, a Sobel mask for gradient was applied to detect the number of points in edges of an image.

It is to be mentioned that Sobel mask has two masks for the gradient components (the horizontal and vertical directions), as shown in Figure-1.

-1	0	+1
-2	0	+2
-1	0	+1

+1	+2	+1
0	0	0
-1	-2	-1

Figure -1 Sobel masks for gradient.

4. The Proposed Method

The recognition process typically comprises of three phases: preprocessing, feature extraction and recognition. Figure-2 describes the diagram of alphabetical Arabic characters recognition method.

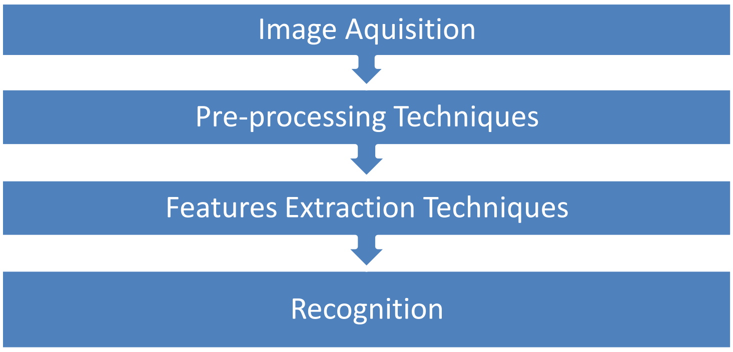


Figure -2 Scheme of characters recognition.

4.1 Image Acquisition

In image acquisition stage, Snipping Tool Program (STP) is applied to obtain (JPEG) format images for Arabic characters excluded doted characters as mentioned in Table-1, taking consideration that these images have taken by using (24 and 27) font sizes and (Times New Roman) font face. Table-2 shows the pronunciation of each Arabic character.

Table -2 The Arabic alphabets and their pronunciation in Arabic and English.

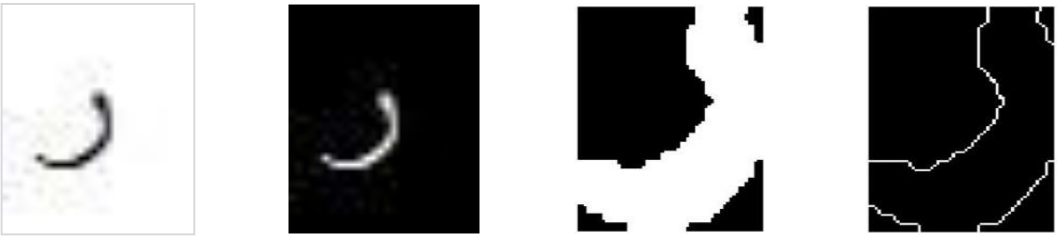
Character in Arabic	Arabic Pronunciation	Character Representation	
ا	Alif	تراب	turAb
لا	Lam Alif	لارا	Lara
ر	Raa	ريف	Reef
و	Waw	نمرود	NamrOUd
طـ	Tta	طريق	Ttareeq
مـ	Mem	ميسان	Mesan
سـ	Seen	سماء	Samaa
صـ	Saad	مصرف	maSSraf
د	Dal	رداء	reDa'a
كـ	Kaf	كركوك	KerKuk
عـ	Aeen	عام	Aamm
حـ	Hha	حيدر	HHaider
هـ	Ha	هادي	Hadi
لـ	Lam	دليل	daLeel

4.2 Pre-processing Techniques

The pre-processing techniques are essential to complete recognition process. This stage consists of four steps, its details in algorithm (1). First process is converting a text image to binary image which is (0-1), where (1) represents the character and (0) represents surrounding area; as shown in Figure-(3b). The second step is removing the black space that surrounding the character (top, bottom, right and left). The third step is changing the size of the image into (100 x 60) pixels to highlight the character image then filling the holes of image regions as shown in Figure-(3c). The final step in this stage is using the Sobel operator to detect the edges of the image as shown in Figure-(3d).

• **Algorithm (1):**

```
Input: read the character image (Img).
Output: Arabic characters images with size (100 x 60) pixels.
            Detect the edges of the character image.
Start:
  i.  Convert an image (Img) to binary (0-1).
  ii.  Remove the black space that surrounding the character.
  iii. Resize the image into (100 x 60).
  iv. Apply Sobel effect to detect the edges of the character image.
End
```



(a) (b) (c) (d)

Figure -3 Preprocessing steps: (a) Original image, (b) Converted image to a binary, (c) Removing the black space and filling holes, and (d) Edge detection.

4.3 Features Extraction Techniques

The technique of features extraction can be defined as the process of extracting unique information from the matrices of digitized characters applying for OCR. The techniques enable OCR system to recognize all characters classes.

It is to be mentioned that the images obtained were selected in different font sizes (10, 16, 18, and 26) to build a database with a very high stability. In this study, the database is consisting of 16 features and its steps of establishing are detailed in algorithm (2).

- **Algorithm (2):**

```
Input: read the character image (Img).
Output: Save the features to be obtained in file with name ($DB_j$) Database in a matrix of dimensions (i, j). Where (i): represent the characters, and (j): represents the number of features to be found and saved in the database.
Start:
    i. Read the image that was resized in the algorithm 1.
    ii. Divide an image into four equal parts ($P_i$) through determining the center of image character ($c_x$, $c_y$).
    iii. Calculate the area of each part $AP_i$, where i=1:4. This means four numerical values can be extracted such as ($AP_1$, $AP_2$, $AP_3$, and $AP_4$).
    iv. Applying Sobel operator to extract edge of each character. Then repeat step (ii) to find the number of edge points in every part. Thus, four edges data were extracted ($APe_1$, $APe_2$, $APe_3$, and $Ape_4$).
    v. Determining four positions of ($x_i$, $y_i$) image character. This step consists of four sub steps:
        1. Determining first point that equal (1) in the first row ($x_1$, 1) of image character.
        2. Determining first point that equal (1) in the last column ($y_1$, 1) of image character.
        3. Determining last point that equal (1) in the last row ($x_2$, $I_h$) of image character.
        4. Determining last point that equal (1) in the first column ($y_2$, $I_w$) of image character.
Where: ($I_w \times I_h$) represent the size of image.
    vi. Calculating the length of each side of the quadrilateral ($d_1$, $d_2$, $d_3$, and $d_4$) by distance Law:
        $d = \sqrt{(x_2 - x_1) + (y_2 - y_1)}$        (3)
    vii. Finding the longest and shortest line ($T_{max}$, $T_{min}$) through calculating the number of pixels that equal (1) at first and last point in the same row, and determining their positions in row ($T_{po1}$, $T_{po2}$) .
Where: $T_{po1}$ and $T_{po2}$ represent the position of the longest and shortest line in the character respectively.
    viii. Create a matrix of two-dimensions. The first dimension represents the number of rows that equal to the number of characters in the database, and the second dimension represents the number of columns that equal to the number of features in each character.
ix. Normalizing the features, that have obtained from all character images, through Finding the maximum and minimum value for each feature of ($Max_j$, $Min_j$) respectively, where j=1:16.
        x. Applying the following equation:
            $DB(i,j) = \frac{F(i,j) - Min_j}{Max_j - Min_j}$        (4)
Where F: [$AP_i$, $AP_{ei}$, $d_i$, $T_{max}$, $T_{min}$, $T_{po1}$, and $T_{po2}$], $F(i,j)$: represent the value of j feature for i image character, and ($Max_j$, $Min_j$): the maximum and minimum value for each feature respectively.
        x. Saving the database.
End
```

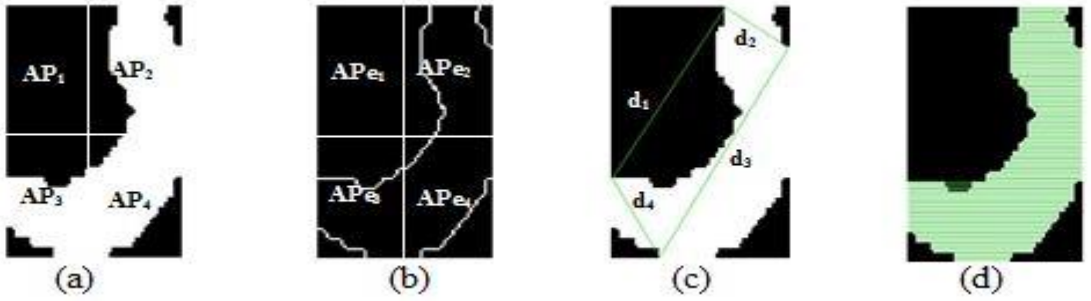


Figure -4 Features extraction steps: (a) AP_i features, (b) AP_{ei} features, (c) d_i features, and (d) (T_{max},T_{min}, T_{po1},T_{po2}) features.

4.4 Characters Recognition

This stage is one of the most important stages in this study. In this paper, the interpolation method is used. It is well-known method that has been used in several researches and proved its effectiveness and efficiency. This method has been used with the help of the minimum distance parameter to obtain best results. To build a database with high setting, three databases are built for image characters in different sizes (10, 16, 18, and 26) with using the same font type. Then comparison has used between image characters in new sizes (24 and 27) with the established database. The recognition stage depends on the algorithm (3) which is detailed below:

- **Algorithm (3)**

```
Input: read the character image (Img).
       Database DB (i, j) created in the algorithm (2).
Output: Arabic characters recognition.
Start:
   i.   Loading character image to be ready for recognizing.
   ii.  Determining features as applied from the steps (i-ix) in algorithm (2).
   iii. Applying minimum distance technique to recognize input character image by matching
        their features with the corresponding rows of the features in the database (DB).
   iv.  Saving the results.
End
```

The database (DB) for unknown alphabetical character is established then compared with 16 characters database features. The final recognition is depending on minimum distance classifier is should be inserted after that measured by using the following equation:

$$D_i = \sum_{j=1}^{F} \frac{|A_{ij} - DB(i, j)|}{|A_{ij} + DB(i, j)|} \qquad (5)$$

Where Di stands for the difference between the input image character and database, *F* represents the total number of features in the database; A_{ij} is the jth feature of the ith input image character, while *DB (i, j)* represents feature j of the input image character in the database.

In the recognition stage, the Euclidean distance between feature of the testing image characters and all features in the database is calculated by using equation 5. The output of the minimum Euclidean distance is considered to be a recognized character sample as shown in Figures (5 and 6). For example the results of the figures such as the characters Aeen - (ع) and Dal - (د) indicate that they are recognized according to their values. Thus, other characters are identified and recognized.

The value of D_i, which represents the minimum difference between the tested image characters and reference databases, is found. After obtaining the minimum value [i.e. min (D)] which is represents the recognition of the targeted character that matches as a close result to the obtained features in the database.

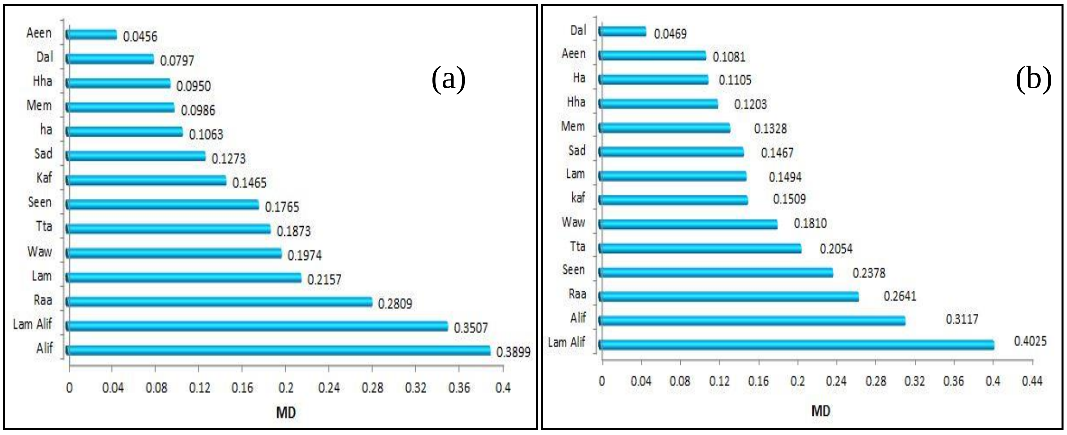


Figure 5-Minimum distances (MD) value per character size of 24: a) Aeen (ع) character, and b) Dal (د) character.

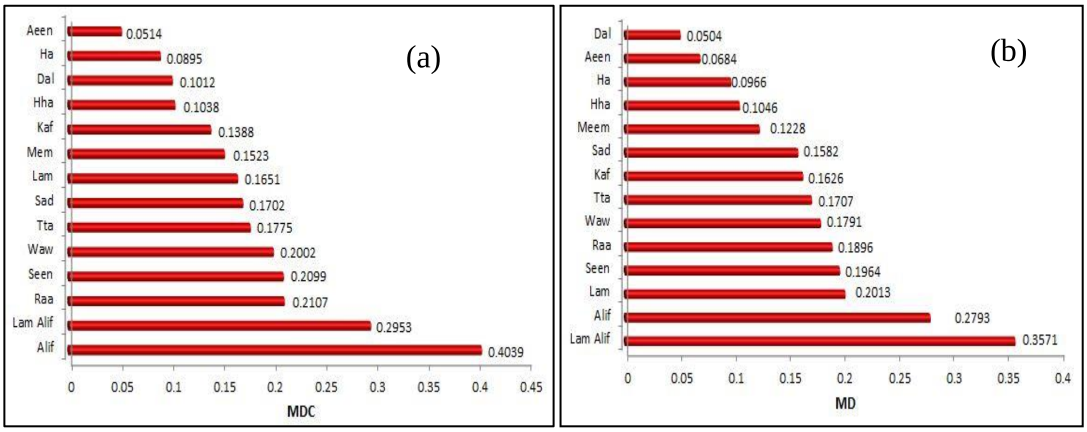


Figure 6-Minimum distances (MD) value per character size of 27: a) Aeen (ـع) character, and b) Dal (د) character.

While the Minimum Distance (MD) of each character was determined, the less value of (MD) is considering being the actual value which is the corrected recognition, the rest values of (MD) are considered to be the expected values.

There are two procedures to determine the rate of recognition. The first procedure is one character against all characters (OAA). The second procedure is one character against one character (OAO).

In this stage, (OAA) procedure was used [i.e. the actual value of targeted character (MD) and the expected values of targeted character ($\overline{\boldsymbol{MD}}$) are used in the percentage error (P.E.)].

The percentage error (P.E.) formula as follows [21]:

$$P.E. = \left|\frac{MD - \overline{MD}}{MD}\right| \times 100\% \qquad (6)$$

Where: ($\overline{MD}$) represents the average of the expected values.

The Recognition Rate (R.R) of each character was obtained by using equation 7:

$$R.R. = (100 - P.E.) * 100\% \qquad (7)$$

Experimental Results

The recognition method, which is discussed in details at section 3.4, have applied using MATLAB program (2015a) and tested with 56 samples of printed Arabic character. These samples, which are written using (Times New Roman) font face with four font sizes 10, 16, 18, and 26, are taken by using Snipping Tool Program (STP). These samples are mentioned in Table-1.

The priority in this study is to develop the databases of printed Arabic characters in order to be used in research fields. To achieve that many features extraction techniques should be using to obtain better results. The achieved results per character are displayed in Table-3.

Table 3-Recognition rate per character by using (MDC).

Character	R. R. (%)	R. R. (%)
	font Size 24	font Size 27
ـع	96.86	97.38
ا	97.27	96.9
د	96.86	97.5
ـحـ	96.82	97.84
ـهـ	97.33	98.14
كـ	97.11	97.71
لا	97.48	96.73
لـ	97.23	97.15
ـمـ	97.62	97.65
ر	97.5	96.84
ـصـ	97.33	96.66
ـسـ	97.51	97.24

طـ	97.55	96.79
و	97.48	96.92
Average Recognition Rate (%)	97.28	97.24

Table -4 Descending comparisons between other publication results and the proposed method.

Publication	Method	Recognition Rate for the Publication
Neila M. & Amar M. [14]	Kohonen network	88.38
Saeed M. et al. [20]	Nearest Neighbor Classifier	94.44
Nadia B et al. [10]	Neuro Fuzzy Classifier	95.64
Majid H. et al. [16]	Max. and Min. local points	96.84
B. Bushofa, & Spann M. [9]	The decision tree Minimum Distance Matching	97.23
Sabri A, Ashraf S. [12]	Fast Hartley Transform	97.3
Proposed method	Minimum Distance Classifier	97.27

It is noticeable that the proposed system of this paper has the best performance comparing with the others in terms the rate of recognition and its significant contribution in terms of the accuracy of recognition. Table-4 summarizes the methods used for characters recognition.

The main principle of this research is to build an advanced system for the recognition of printed alphabetic Arabic characters depending on new extracted features (structural and numerical). The estimation of this system is tested by using MD to those mentioned features. The proposed approuch is given unique encouraging results in which the recognition rate reached to 97.28% and 97.24% at font sizes 24 and 27 respectively.

The outcomes clarify that higher rates of recognition were accomplished by using features extraction approaches. The applied techniques were given percentage recognition of about 97.62% for Mem - (مـ) character using font size 24 and 98.14% for Ha - (هـ) character using font size 27, while ehe poorest rates of the recognized image characters were Hha - (حـ) character and Sad - (صـ) character using font sizes 24 and 27 respectively.

6. Conclusions

The suggested algorithm proved that it is efficient and fast because it depends on a shrunken database, in addition of it is a new way to distinguish and recognize character among set of characters when compared with other research. The particular character found by using technique called Minimum Distance Classifier between the input character images and features that saved in the database. In the Analysis of level character recognition; it was observed that the average of recognition rate reach to 97.28% and 97.24% at font sizes 24 and 27 respectively. This method is recommended to be used for all of Arabic characters excluded dotted characters (beginning, mid, and end of the word) for all font sizes and font faces.

References

1. Chaudhuri, A., Mandaviya, K., Badelia, P., Ghosh, S. **2017.** Optical character recognition systems for different languages with soft computing. *Stud Fuzz Soft Comp*, **248**: 9-42.
2. Ahmed, T., Mohammed, R. and Abdelhay, S. **2014.** Investigating of preprocessing techniques and novel features in recognition of handwritten Arabic characters. *Lect. Notes Artif Int*., 2014; 264–276.
3. Kenz, A., Osei, A. and Ali, M. **2013.** Detection of facial expressions based on Morphological face features and Minimum Distance Classifier. In: IEEE 2013 International Conference on Sciences and Techniques of Automatic Control and Computer Engineering 20-22 December 2013; Sousse, Tunisia: IEEE. pp. 487-493.
4. Dehghan, M., Faezl, K., Ahmadi, M. and Shridhar, M. **2000.** Off-line unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov word models. In: IEEE 2000 Proceedings of the 15th International Conference on Pattern Recognition 3-7 September 2000; Barcelona, Spain: IEEE. pp. 351–354.

5. Amir, M, Karim, F. and Abolfazl, T. **2002.** Feature extraction with wavelet transform for recognition of isolated handwritten Farsi/Arabic characters and numerals. In: IEEE 2002 International Conference on Digital Signal Processing; 1-3 July 2002; Santorini, Greece, Greece: IEEE. pp. 923 – 926.

6. Amrouch, M., Elyassa, M., Rachidi, A. and Mammass, D. **2008.** Off-Line Arabic handwritten characters recognition based on a hidden Markov models. In: IEEE 2008 International Conference on Image and Signal Processing; 1-3 Jul; Cherbourg, France: IEEE. pp. 447 – 454.

7. Ahlam, M., Akram, H., Khalid, S. and Hamid, T. **2014.** Using HMM toolkit (HTK) for recognition of Arabic manuscripts characters. In IEEE 2014 International Conference on Multimedia Computing and Systems; 14-16 April 2014; Marrakech, Morocco: IEEE. pp. 475 – 479.

8. Talaat, S. and Ramez M. **1988**. Automatic recognition of isolated Arabic characters, *J Signal Process*; **2**: 177- 184.

9. Bushofa, B. and Spann, M. **1997.** Segmentation and recognition of Arabic characters by structural classification. *J. Image Vision Computer*, **15**: 167-179.

10. Nadia, B., Majed, Z. and Najoua, E. **2006.** Neuro-Fuzzy approach in the recognition of Arabic characters. In IEEE 2006 Information and Communication Technologies; 24-28 April 2006; Damascus-Syria: IEEE. pp. 1640 – 1644.

11. Nadia, B., Najoua, E. **2006.** Combining a hybrid approach for features selection and hidden Markov models in multifont Arabic characters recognition. In IEEE 2006 International Conference on Document Image Analysis for Libraries; 27-28 April, Lyon, France: IEEE. pp. 103 – 107.

12. Sabri, A. and Ashraf, S. **2009.** The use of Hartley transform in OCR with application to printed Arabic character recognition. *J Pattern Anal Appl*, **12**: 353-365.

13. Mohamed, S. and Amin, A. **1989.** On-line recognition of handwritten isolated Arabic characters. *J Pattern Recogn*, **22**: 97–105.

14. Neila, M., Amar and M., Mohamed, **2002.** Ch. On-line recognition of handwritten Arabic characters using a kohonen neural network. In IEEE 2002 Proceedings Eight International Workshop on Frontiers in Handwriting Recognition; 6-8 August 2002; Ontario, Canada: IEEE. pp: 490–495.

15. Jakob, S., Jonas, M., Jonas, A. Christer, F. **2009.** On-line Arabic handwriting recognition with templates. *J Pattern Recogn*, **42**: 3278-3286.

16. Majid, H., Dzulkifli, M., Abdolreza, R. **2010**. Deductive method for recognition of on-line handwritten Persian/Arabic characters. In IEEE 2010 International Conference on Computer and Automation Engineering; 26-28 February 2010; Singapore, Singapore: IEEE. pp. 791–795.

17. Redouane, T. and Abdelkader, B. **2012.** Arabic on line characters recognition using improved dynamic Bayesian networks. In IEEE 2012 International Conference on Multimedia Computing and Systems; 10-12 May, Tangier, Morocco: IEEE. pp. 290-295.

18. Sherif, A., Hany, A. **2011.** Recognition of segmented online Arabic handwritten characters of the ADAB database. In IEEE 2011 International Conference on Machine Learning and Applications; 18-21 December, Honolulu, Hawaii, USA: IEEE. pp. 204-207.

19. Surya, V. and Arindama, S. **2008.** Edge detectors based anisotropic diffusion for enhancement of digital images. In IEEE 2008 Indian Conference on Computer Vision, Graphics & Image Processing; 16-19 December 2008; Bhubaneswar, India: IEEE. pp. 33-38.

20. Saeed, M., Karim, F., Majid, Z. **2005.** Structural decomposition and statistical description of Farsi/Arabic handwritten numeric characters. In IEEE 2005 International Conference on Document Analysis and Recognition; 31 August-1 September 2005; Seoul, Korea: IEEE. pp. 237-241.

21. Guang, W., Massimo, B., Mario, F. **1995.** Calculating percentage prediction error: a user's note. *J Pharmacol Res*, 1995; **32**: 241-248.