Journal of Theoretical and Applied Information Technology
15th August 2017. Vol.95. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GEOMETRIC-TOPOLOGICAL BASED ARABIC CHARACTER RECOGNITION, A NEW APPROACH

HAMED TIRANDAZ, MOHSEN AHMADNIA AND HAMIDREZA TAVAKOLI

Electrical and Computer Engineering Department, Hakim Sabzevari University, Sabzevar, Iran
Email: {tirandaz,m.ahmadnia,tavakoli}@hsu.ac.ir

ABSTRACT

Optical Character Recognition (OCR) is a very old and of great interest in pattern recognition field. In this paper, a new algorithm based on morphological structure is proposed for Arabic character recognition. Our proposed method uses center of mass calculation. It is invariant with the size, translation and rotation of the target image. In addition, topology-based landmarks like intersection pixels masking the intersection of loops and multiple strokes, as well as end points have been used to compute centers of mass of these points located in the individual quadrants of the circles enclosing the characters. After doing initial pre-processing operations like binarization, resizing, normalization, removing noise, skeletonization, the total number of intersection pixels as well as the total number of end points are determined and stored. The character image is then encircled and divided into four quadrants. The center of mass of the character image as well as the masses of each of its four quadrants are determined and the Euclidean distances (ED) of the intersection and end points in each of the quadrants with the massed are calculated. These quantities are determined for both the target and prototype image and then the best match is achieved with the character having the minimum ED. Results show that the presented method opens up a new direction for dealing with the complex problems of OCR.

Keywords: *Arabic Character Recognition, OCR, Center of Mass, Geometric-Topological features*

1. INTRODUCTION

Character recognition is one of the most popular area of research in pattern recognition because of its immense application potential. It means translating images of characters into an editable text; in other words, it represents an attempt to simulate the human reading process.
Obviously, different languages have very different characteristics of their alphabets. Arabic character recognition is different from Latin character recognition. It is a cursive script and it has many different features.
Character recognition for both printed [17] as well as handwritten of any language characters [5] is still a challenging problem, despite the huge works on it. Also success of the commercially available OCR system is yet to be extended to handwritten text.

Most of the existing OCR systems have been designed for recognition of characters of Latin languages and also East Asian scripts such as Chinese and Japanese [13]. There are distinct differences between Farsi and Latin letters, especially the cursive nature of Farsi alphabet in both printed and handwritten texts, and thus, it is not possible to use provided techniques for Latin text recognition directly for Farsi text recognition without first making some fundamental changes [14].

Many researches have been carried out in the application of OCR technology for handwritten Arabic texts [15][16].
Research in Farsi OCR technology started in the early 1980's by Parhami et al [17], and this effort was followed by many research labs and universities across the globe with nearly acceptable results for printed Farsi documents or texts.

Although there are some researches for recognition of handwritten Farsi digits with nearly acceptable results, but available methods – due to of Farsi's alphabet special nature – are not completely extendable to Farsi alphabet characters. Also, the complex nature of cursive handwriting poses challenging problems in FOCR systems, and researches are still being carried out to find satisfactory solutions. Hence, available FOCR systems have a large distance from their real place, and research on this topic is still hot and demanding.

In this paper we focus on Arabic language, which, because of its cursive script, the problem

Journal of Theoretical and Applied Information Technology
15th August 2017. Vol.95. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

of OCR is more sophisticated than Latin languages that has separated characters. Furthermore, Unlike Latin characters, *Arabic* words may not always be placed in disjoint positions, parts of a single character may be interleaved with another character, and Modified shapes. Arabic text does have delimiters like spaces, separating different words. Also, several characters in the Arabic alphabet could be homomorphic, i.e. have similar shape pattern which could add to the complexity of the recognition process. Thus, Arabic OCR is a very challenging task and many research efforts have been conducted to perform these task. But evidences of research on OCR of *Arabic* characters, as observed in the literature area are a few in numbers [4]. A survey of the previous approaches to OCR for the Arabic language have been discussed in [1].

This paper proposes a geometric topological based algorithm for Arabic character recognition by combining the Size Translation Rotation Invariant Character Recognition and Feature vector Based (STRICR-FB) algorithm originally proposed by Barnes et al [5] along with some topological features of the individual characters.

The rest of this paper is organized as follows: The structure of Arabic language is presented in the next section, followed by a description of allied work in the following section. In section IV, a review of the original STRICR-FB is presented. The proposed approach is presented in Section V. Finally, experimental results of the proposed method is evaluated in section VI, and conclusions are discussed in section VII.

2. ARABIC LANGUAGE STRUCTURE

As mentioned above, a comprehensive database have to model existing limitation of a real application. Therefore, it is important to consider the Arabic language features alongside collecting data.

The most important of the Arabic language writing feature is perhaps its continuity and also its writing side (right to left) unlike English language. Also *Arabic* words may not always be placed in disjoint positions, parts of a single character may not be interleaved with another character, and Modified shapes. In this section we will concentrate on these challenges:

A. Similarity

Furthermore, several characters in the Arabic alphabet as shown as in Fig 1 are homomorphic, i.e. have similar shape pattern which could add to the complexity of the recognition process, such a herculean problem.

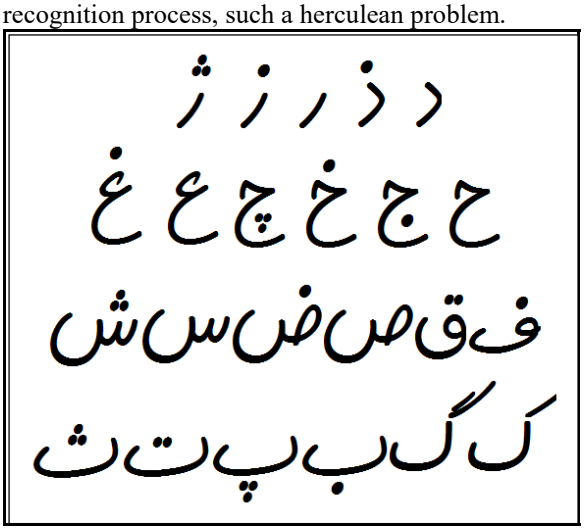

Fig 1. . Arabic Character Classification With Similar Shapes

B. Sub-words

Arabic language is consisted of 32 alphabet. Like Arabic, Arabic is a right to left script, but there are some differences like number of alphabets, font styles, vocabulary and signs, which make Arabic OCR somehow different from Arabic.

Arabic alphabets can be connected from both side inside a word except for seven alphabets that can be connected just from left side. These seven characters are

و ژ، ز، ر، ذ، د، ا،

Consequently, every word containing some of these seven alphabets divided into separate parts as name as sub-words. These sub-words are separated with a space between two parts. An example of these sub-words is depicted in Fig 2.

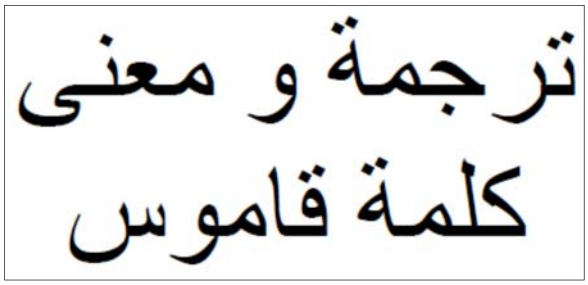

Fig 2. Examples Of Words Containing Some Sub-Words

C. Alphabet Variation

Journal of Theoretical and Applied Information Technology
15th August 2017. Vol.95. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The shapes of alphabet characters in Arabic language is a function of its location inside of the word. An alphabet could have four different shapes depend on its location in the first, middle, the end, and also apart (appearance separately). Fig 3 shows these four different shapes of an alphabet inside of a word. The corresponding alphabet is depicted with a distinct color.

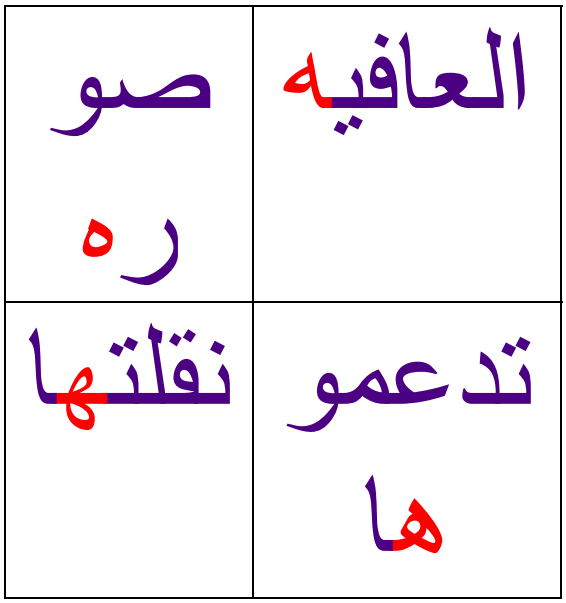

Fig 3. Four different shapes of an alphabet variation.

D. Overlapping

Arabic characters are not separated vertically even those without any connectivity, thus making character recognition so difficult. Some examples of overlapping between adjacent characters is shown in Fig 4. As it can be seen, several adjacent characters have vertical overlapping, making its recognition so difficult even for human eyes.

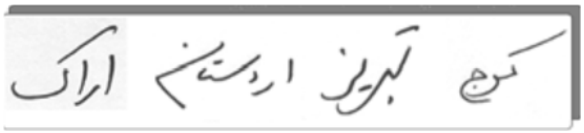

Fig 4 typical examples of overlapping between adjacent characters at a Arabic writing pattern.

A. *Character Features Vectors Identification*

Every character at any language has its own identities and features. By identifying features we can recognize characters from a textual image document. By feature extraction the critical characteristics of characters gets isolated, and that reduces the complexities of the pattern. By extracting every characters of the input text image, their feature vector could be calculated separately. After classification, each character feature vector compares with known patterns and then matched with the character that has the same characteristics. The characters can be further subdivided into segments and the strokes in each of these segments exhibit certain characteristics in terms of shape, angle of inclination. In addition, presence of intersection between characters in writing patterns, smooth and also decline text images change the character feature vectors. All these aspects need to be taken into account in devising significant feature vectors for matching and identification.

3. ALLIED WORK

There has been extremely research done at about character recognition area, implemented on a few common languages such as, Latin, Chinese, etc. However, developing OCR systems for other languages such as Arabic didn't receive the same amount of attention.

In the field of characters recognition, there are several approaches which differ from each other basically in the type of features are used to identify characters and in the computational intelligence techniques exploited to recognize them. Types of features used in the literature include topological and geometrical, directional, mathematical, and structural features.

Generally, most of the research done towards on Arabic character recognition, both for printed and handwritten documents, and either use image morphology/shape attribute or pattern recognition for classification. The most popular pattern recognition approach are include as, the artificial neural network [7], fuzzy systems [9] followed by Genetic Algorithm [10], Support Vector Machine [10], and Hidden Markov Models (HMM) [11]. Statistical analysis [4], structural and syntactic analysis [5]

Morphological or Image shape-based methods typically segment the target character image and measure its similarity with a template image of the prototype character. In some cases, researchers use some of character feature such as: shape, size, extent and angular inclinations of the different strokes. In addition, character image (either target or template) can be partitioned into some parts and the contours of the strokes belong to each part can be considered. These procedure had showed to be useful, especially for hand-

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

written character recognition where there exist many undistinguished variations. Pattern matching [6], Thinning methodologies [12] are some of the computational techniques applied in this problem [5].

Some of the major efforts are stated in the following.

A technique for cursively handwritten *Arabic* script recognition is proposed in [24]. In this technique, words are segmented into strokes, and these strokes are further classified by using the geometric and topological features.

In [18], Arabic words were described by a set of features describing structural characteristics and then classification of this description was performed by a Hidden Markov Model (HMM). In [19] again used a HMM, this time describing the word with Fuzzy Vector Quantization on the image's contours. An interesting third approach was presented in [20], which used data in a feature vector created by the analysis of the structure of the Arabic word. This information was then combined with Knowledge Based Artificial Neural Network. This technique aims to combine the advantages of Artificial Neural Networks (ANNs) and expert systems, first generating a network based upon domain information which is then trained as usual.

The other approach, character level identification, has two sub-parts; those that seek to identify individual characters for tasks such as postal code recognition, and those that recognize whole words by breaking them down into a sequence of characters. The latter approach has been hampered by the difficulty of accurately separating characters in a word, acknowledged as a main contributor to classification difficulties encountered in [21].

Bhattacharya *et al.* (2002) presented a hybrid scheme for hand-printed numeral recognition based on a self-organizing network and multilayer perceptron (MLP) classification techniques.

Convergence of the shortest path has been used as a criterion for segmentation in [11]. An algorithm that takes into account the variations of angles due to pen position trajectories has been presented in [22]. Cursiveness and context sensitivity [8] is also another technique that has been used.

A new template matching scheme and applied to the recognition of cursive Arabic script has been presented. in [23]. In their works, template Matching and Correlation is used to match patterns pixel by pixel to identify them.

Reference [3] proposes a technique that normalized the text in bounding box by estimating the partial height of character using which the normalized size and height of the characters are defined [23]. In [3], statistical and structural method are combined for feature extraction and 12 features are extracted from both main and secondary parts of the characters.

A survey on handwritten text recognition has presented in [13].However, a comprehensive study of all the presented methods is beyond of this paper.

In both pattern recognition and Morphology approaches, the target characters are matched with template (prototype) characters. The drawbacks with the approaches mentioned above are that any change in size, translation and rotation affect the features, and hence the recognition process. A tremulous line might be longer or shorter than a normal line. Loops and swirls might be open or tight. In addition, the slanting orientation of right and left handed writers could be different. These orientations might result in variations of angles of the individual strokes of the characters. Calligraphy stills might be plain or ornate. Handwritten patterns can be affected by all of these challenges. Furthermore, as we mentioned at the previous sections, homomorphic, cursive scripts character variations and overlapping are some of important difficulties in Arabic character recognition problem. The STRICR-FB algorithm [3], consider all type of these mentioned challenges. The primary method utilized on Japanese character recognition [2]. The extension of the STRICR-FB algorithm proposed in this paper has been tested on several handwritten samples and a high accuracy has been obtained by including topological features of the individual characters of the character sets. These topological features included intersection pixels as well as end points. Furthermore, by dividing the image into four quadrants and finding the COM and EDs (ED) within the four quadrants enhanced the accuracy.

4. OVERVIEW OF STRICR-FB

The original STRICR-FB algorithm is performed in two phases: (i) the construction of Character Unique Feature Vectors (CUFV) (ii) passing of these CUFVs through a neural network for recognition. After determining the Center of Mass of each of the prototype and examine characters, the four CUFV, namely, mean variance, character density and decentricity have

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

been calculated. In the second phase, an unsupervised clustering algorithm has been used, where clusters are built with real data.

5. PROPOSED ALGORITHM

We have created our method based on STRICR-FB [3] in the first phase, where the center of mass (COM) is first calculated. Then the character image will be normalized. Subsequently, the extraction of intersection pixels are performed. The features of the prototype characters are compared by template matching with the candidate character features obtained from its corresponding image, using an ED measure defined below.

The entire character image is enclosed by a circle whose center and radius are computed from the character image. This circle is subsequently divided into four quadrants. The CoM of each sub-image of either the prototype or the examine character image is determined using the occupied percentage of pixels (the number of black pixels versus white pixels) and their spacial locations (x, y). Once the COM is obtained, we try to locate the COM position among the four quadrants of the encircled character.

Due to presence of multiple stoke type and orders; there are characters with intersection pixels and non-intersection pixels. An intersection point is a point in the image where multiple strokes intersect. The characters with multiple strokes and intersection point(s) create a pattern of relationship between COM location and the number of intersection pixels for each of the individual characters of the Arabic script. Then, a template matching scheme based on the distances of the intersection pixels from the COMs in each of the quadrants provides a robust method.

Arabic characters are homomorphic and so the computation of COM alone might not be sufficient to recognize individual characters. Furthermore, handwritten characters vary widely from person to person. Thus distinctive landmarks are required to impart a unique identity to the character. In particular, intersection pixels, which mark the intersection of closed loops and also multiple strokes play an important role in identifying the topology of the character. Thus, after normalizing a handwritten character for size, shape and orientation and subsequent fitting into a circle, the intersection pixels in each quadrant are located and COMs are measured for each character w.r.t to the intersection pixels of that character. EDs of test and template images are calculated for these intersection pixels. The outline of the process is given in Figure 2.

After preprocessing and normalization, the matching process involves (1) COM determination, (2) location of intersection pixels and end points (3) measuring the ED (4) comparison of template and target images and (5) check matching. These steps have been described as follows:

A. COM determination

The COM determination is consisted by the computation of $\bar{x}_{com}$ and $\bar{y}_{com}$ points where $\bar{x}_{com}$ represents COG of x-th axis and $\bar{y}_{com}$as that of y-th axis coordinate. These parameters can be determined as follows:

$$\bar{x}_{com} = (\sum_{x=1}^{m}\sum_{y=1}^{n} x\, I_{xy}\,)/(\sum_{x=1}^{m}\sum_{y=1}^{n} I_{xy}\,) \quad (1)$$

And

$$\bar{y}_{com} = (\sum_{x=1}^{m}\sum_{y=1}^{n} y\, I_{xy}\,)/(\sum_{x=1}^{m}\sum_{y=1}^{n} I_{xy}\,) \quad (2)$$

Where I_{xy} indicates the (x, y) position of the character image. These computation of COM are performed for a prototype Arabic alphabet images are listed in table.1. As we can see, characters with similar morphology have nearly equivalent COM figures.

B. *End points identification and intersection* pixels *(s)*

The intersection pixels are located initially visually inspection. The total number of intersection pixels and the distance between those and COM vary from of any character to others. For the characters which don't have intersection pixels, end points are calculated. And then the coordinate of the intersection and end points are banked up.

C. *Euclidean Distance(ED)*

The EDs between COM and each of the intersection pixels are calculated next (for the characters and intersection of their multiple strokes). The characters which don't have intersection pixels, the EDs were measure this COM and end points. The calculation of ED is measured as:

$$d(x,y) = \sqrt{[(x-\bar{x}_{com})^2+](y-\bar{y}_{com})^2} \quad (3)$$

Where 'd(x,y)' represents ED and (x, y) is the coordinate of an intersection point of the sample character in the image and $(\bar{x}_{com}, \bar{y}_{com})$ is the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

COM of the character image obtained from equation (1) and (2), respectively.

D. *Prototype and Objective Images Comparison*

Once the COM and ED is calculated for the all prototype character images, the steps A, B and C is repeated for the Objective image of the target - generally handwritten- text, too. And then the obtained values of the prototype image with all of the values of (handwritten) target image(s) are compared. If the character has two intersection pixels, then the calculation will be followed for prototype is:

$$d1_{ij} = \sqrt{(x_1 - \bar{x}_{COM})^2 + (y_1 - \bar{y}_{COM})^2} \quad (4)$$

The EDs of the handwritten character in target image with COM and three intersection pixels will computd as:

$$d2_{ij} = \sqrt{(x_4 - \bar{x}_{COM})^2 + (y_4 - \bar{y}_{COM})^2} \quad (5)$$

If the deference of the average value of the ED of prototype and target image tends to zero then the target image that is chosen to be matched with prototype image is considered as similar.

E. *Template Matching*

Applying on the formula (7), given below, we can get the percentage of matching for target image to template (prototype) image, followed by the average ED of the collected intersection pixels (6). The target character can be identified as:

$$d_{av} = \frac{1}{n}\sum_{i=1}^{n} d_i \quad (6)$$

$$Match = \frac{d_{av(target)}}{d_{av(prototype)}} \times 100\% \quad (7)$$

We have calculated the average ED based on the total summation of 'n' intersection pixels for prototype and target images and then we can calculate the match amount of the target with the prototype character image.

PREPROCESSING

1. Binarization
2. Image adjustment
3. Rotation
4. Cropping
5. Removing noise using Gabor filters

POINT COORDINATES

1. Skeletonization
2. Find of the total number of intersection pixels
3. Storing the intersection pixels coordinates
4. Obtain of the total number of end pixels
5. Storing the coordinates of end pixels

COM COMPUTATION

1. Encircle each character image
2. Determine the COM for each character.
3. Partition each circle into four quadrants
4. Determine the COM in quadrant (n)
5. Find intersection pixels with multiple strokes
6. Get ED between COM and each intersection pixels.
7. Calculate average ED for all intersection pixels
8. Calculate percentage of match between every prototype and target images

All of the prototype images and target images are pre-processed, encircled, partitioned, normalized and the relevant figures of their features are calculated by visually inspected.

Eventually, it is clear that (A) the spacial coordinate of the COM of the entire character image is not always at the center, but it is somewhat close to the center (according to the stroke density of the pixels present in the image) and (B) the location of the COM (in the quadrant of the encircled character) is multiple of the total number of the intersection pixels obtained for that particular character.

The intersection pixels are occurred when some strokes intersects at a pixel. And hence the operation is limited to the image characters with multiple stokes and presence of intersection of the strokes. The matching comparison of the characters having single stroke or multiple strokes with zero intersection point can be acquired from equation (7).

6. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, a comparative work has been done where the obtained figures from character image features are compared by our described method. After that, an evaluation is been made based on this comparative results. Finally, for any target character image, a reference character with most similarity of its character image is selected.

As we can see, an accurate match was obtained between the target and the prototype character images by utilizing our presented method. The obtained values are evaluated in order to find the best match character for a target image. The match process is consisted of the groups of measurement performance methods that measures similarity amount between the feature vectors of the target character image and the description of each prototype character.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Different performance measures could be used, but the most common technique is the ED. Table I illustrates the measurement of COM for all prototype Arabic character images.

We have run our presented algorithm on 32 Arabic prototype characters and five different handwritten (target) images of 32 Arabic characters, separately. An example of the calculations performed for the character "ط" is provided in Table II.

Handwritten OCR is more sophisticated because of the various writing patterns by different persons due to various forms of writing. Collection of multiple handwritten samples proves COM location could be one of the factors for identifying the handwritten pictorial character sample. We have examined our proposed method for 960 character samples (30 sample for each 32 characters) with 5 prototype characters. Each prototype had contained 32 pattern characters. As an example, the obtained result for 30 sample of character "ط" is illustrated in Table III. Along matching process, each of the 30 sample characters are compared with all of the each 5 prototype pattern characters. Decision making was done based on ED measurements between each sample and prototype character. Accurate decision is shown by T (True) and inaccurate decision is shown by F (False), as we can see in Table III. As we can see, our proposed method can recognized "ط"character by 76% precision, overall. This precision can be improved to 93% by considering all 5 prototype results because we have 2 state under 50%. We have accomplished these verifications for other 31 handwritten test characters with 30 samples for each of them as like as "ط" character.

7. CONCLUSION

A method for recognition of Arabic characters is developed based on geometrical-topological image features. of the Arabic characters and worked on five different printed scripts and also two different handwritten scripts for each of 32 characters for recognition by template matching. Due to the nature of on geometrical-topological of Arabic character images, we have used the primary STRICR-FB method for COM determination along with topological features, namely, intersection pixels and end points for both printed and handwritten character images which is a development over the result obtained using STRICR-FB for Japanese character recognition. Because using of topological features in the recognition process can improve recognition precision. The implementation results verify the efficiency of our proposed algorithm.

Table 1. Calculation Of COM For Prototype Characters

character	Xbar	Ybar
الف	0.4674	0.5455
ب	0.5455	0.5464
پ	0.4886	0.5446
ت	0.7101	0.5258
ٹ	0.7200	0.5106
ج	0.4828	0.4648
چ	0.4946	0.4857
ح	0.4583	0.4545
خ	0.5690	0.4930
د	0.6909	0.5641
ذ	0.7037	0.6000
ر	0.6471	0.6825
ز	0.6632	0.7206
ژ	0.6000	0.7015
س	0.5972	0.5234
ش	0.6754	0.5818
ص	0.5513	0.5793
ض	0.5955	0.5782
ط	0.7326	0.5375
ظ	0.6988	0.5375
ع	0.5347	0.4143
غ	0.5873	0.4225
ف	0.7465	0.6266
ق	0.6364	0.6133
ک	0.7037	0.6038
گ	0.6563	0.6339
ل	0.6881	0.6286
م	0.3553	0.5102
ن	0.7159	0.5571
و	0.5814	0.5278
ه	0.5811	0.7143
ی	0.6087	0.6049

Table 2. Euclidian Distance (ED) Computation

character	Euclidian Distance
الف	0.16259
ب	0.0854
پ	0.1413
ت	0.0809
ٹ	0.0934
ج	0.1623
چ	0.1434
ح	0.1889
خ	0.732
د	0.0682
ذ	0.0990
ر	0.1493
ز	0.1894
ژ	0.1698
س	0.0315
ش	0.0660
ص	0.0903

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ض	0.0557
ط	0.0031
ظ	0.0692
ع	0.1529
غ	0.1200
ف	0.1482
ق	0.0790
ک	0.1016
گ	0.1022
ل	0.1104
م	0.2753
ن	0.0884
و	0.0486
ه	0.1864
ی	0.0737

Table 3 The results of matching process between 30 sample "ط" character with 5 Prototype pattern dataset characters.

Sample"ط"	PType1	PType2	PType3	PType4	PType5	Average
1	T	T	F	T	T	80%
2	F	F	T	T	T	60%
3	T	T	T	T	T	100%
4	F	T	F	T	T	60%
5	T	T	F	F	F	40%
6	T	T	F	F	T	60%
7	T	F	T	F	T	60%
8	T	T	F	T	T	80%
9	T	T	T	F	T	80%
10	T	T	T	T	T	100%
11	T	F	T	T	T	80%
12	T	F	T	T	T	80%
13	F	F	T	T	T	60%
14	T	F	T	T	T	80%
15	T	T	T	T	T	100%
16	T	T	T	T	T	100%
17	T	F	T	T	T	80%
18	T	T	T	T	T	100%
19	T	F	F	T	T	60%
20	T	F	F	F	T	40%
21	T	T	T	T	T	100%
22	T	T	F	T	T	80%
23	T	T	T	T	F	80%
24	F	T	F	T	T	60%
25	T	T	T	T	F	80%
26	T	T	T	F	T	80%
27	T	T	T	F	T	80%
28	F	F	T	T	T	60%
29	T	T	F	T	T	80%
30	T	T	F	T	T	80%
Overall	83%	67%	63%	77%	90%	Average=76%

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4 Table 5 the overall results of matching process between 30 sample characters with 5 Prototype pattern dataset characters.

Sample	PT1(%)	PT2(%)	PT3(%)	PT4(%)	PT5(%	Average	Final decision
ا	88	85	92	94	94	90.6%	100%
ب	70	81	67	63	83	72.8%	92%
پ	79	70	83	52	67	70.2%	94%
ت	73	67	76	67	80	72.6%	90%
ث	83	65	76	57	71	70.4%	89%
ج	76	71	70	53	72	68.4%	91%
چ	75	61	87	54	77	70.8%	90%
ح	84	68	67	76	83	75.6%	94%
خ	78	87	67	69	77	75.6%	96%
د	66	93	89	95	89	86.4%	89%
ذ	76	76	87	64	81	76.8%	85%
ر	88	63	65	78	80	74.8%	90%
ز	78	58	86	55	94	74.2%	89%
ژ	75	65	81	82	89	78.4%	85%
س	83	77	86	87	80	82.6%	96%
ش	91	75	66	65	87	76.8%	97%
ص	88	73	67	76	82	77.2%	90%
ض	94	83	75	63	86	80.2%	90%
ط	83%	67%	63%	77%	90%	76%	93%
ظ	91	55	82	90	93	82.2%	89%
ع	86	74	69	81	87	79.4%	91%
غ	85	81	67	85	80	79.6%	88%
ف	92	76	61	78	94	80.2%	95%
ق	82	76	87	71	90	81.2%	97%
ک	75	65	66	80	92	75.6%	89%
گ	84	56	81	87	87	79%	84%
ل	90	60	78	80	92	80%	94%
م	79	84	61	77	88	77.8%	92%
ن	67	82	76	79	76	76%	96%
و	80	75	92	84	91	84.4%	89%
ه	91	80	63	67	94	79%	97%
ی	58	77	60	86	78	71.8%	90%
Overal						77.39%	92%

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] A. Amin, Off-line Arabic character recognition: the state of the art, Pattern Recognition 31 (5) (1998) 517–530.

[2] Soumendu Das, Sreeparna Banerjee, An Algorithm for Japanese Character Recognition, *I.J. Image, Graphics and Signal Processing,* 2015, 1, 9-15.

[3] D. Barnes and M. Manic, STRICR-FB a Novel SIze-Translation Rotation Invariant Character Recognition Method (2010), Proceed. 6th Human System Interaction Conference, Rzeszow, Poland, 163-168

[4] A. M. Sarhan, O. Helalat, "Arabic Character Recognition using ANN Networks and Statistical Analysis", proceedings of world academy of science, engineering and technology volume 21 ISSN 1307-6884, 2007.

[5] B.Q. Vuonga, S. C. Hui a, Y. He, "Progressive structural analysis for dynamic recognition of on-line handwritten mathematical expression", Pattern Recognition Letters 29, 647–655, 2008.

[6] S.D. Connell, A.K. Jain, "Template-based online character recognition", Pattern Recognition 34, 1-14, 2001.

[7] S. Sural , P.K. Das, "Fuzzy Hough transform and an MLP with fuzzy input/output for character recognition", Fuzzy Sets and Systems, Vol. 105, pp. 489-497, 1999.

[8] MS Khorsheed. Off-line arabic character recognition–a review. *Pattern analysis & applications*, 5(1):31–45, 2002.

[9] B. Lazzerini, F. Marcelloni, "A linguistic fuzzy recognizer of off-line handwritten characters", Pattern Recognition Letters 21, 319±327,2000.

[10] C.Pornpanomchai, M.Daveloh,"Printed thai character recognition by genetic algorithm", 1-4244-0973, IEEE, 2007.

[11] Alama S.' adeed (2008). , Recognition of Offline Handwritten Arabic Word Using Hidden Markov Model Approach" , 16th International Conference on Pattern Recognition (ICPR'02) - Volume 3.

[12] L. Lam, S. W. Lee, and C. Y. Suen, ―Thinning methodologies—A comprehensive survey,‖ *IEEE Transaction on. Pattern Analysis and Machine Intellgence.,* vol. 14, pp. 869–885, Sept. 1992.

[13] Parvez, M.T., & Mahmoudi, S.A. (2013). Offline Arabic Handwritten Text Recognition : A Survey. *ACM Computing Survey, 45*(2), 23.

[14] Elzobi, M., Al-Kamdi, A., Dinges, L., & Michaelis, B. (2010). A Structural Features Based Segmentation for Offline Handwritten Arabic Text. *5th IEEE International Symposium on Image/Vision Communication over Fixed and Mobile Networks*, pp. 1-4.

[15] Abandah, G.A., & Anssari, N. (2009). Novel Moment Features Extraction for Recognizing Handwritten Arabic Letters. *Journal of Computer Science*, *5*(3), 226-232.

[16] Dinges, L., Al-Hamadi, A., Elzobi, M., Al-Aghbari, Z., & Mustafa, H. (2011). Offline Automatic Segmentation based Recognition of Handwritten Arabic Words. *International Journal of Signal Processing, Image Processing, and Pattern recognition*, *4*(4), 131-143.

[17] Parhami, B., & Taraghi, M. (1981). Automatic recognition of printed Farsi Texts. *Pattern Recognition, 14*(6), 395-401.

[18] Khorsheed, M.S. ; Clocksin, W.F.: Structural Features Of Cursive Arabic Script. In: British Machine Vision Conference 2, Session 7 (1999)

[19] Deghan, M. ; Faez, K. ; Ahmad, M. ; Shridhar, M.: O_ Line Unconstrained Farsi Handwritten Word Recognition Using Fuzzy Vector Quantization And Hidden Markov Word Models. In: 15th International Conference on Pattern Recognition 2000, Proceedings 2 (2000), S. 351_354

[20] Meslati, L.S ; Sellami, M.: A Hybrid Approach For Arabic Literal Amounts Recognition. In: The Arabian Journal for Science and Engineering Volume 29, Number 2B (2004), S. 177_194

[21] Fahmy, M.M.M: Automatic Recognition Of Handwritten Arabic Characters Using Their Geometrical Features. In: Studies in Informatics and Control Journal (SIC Journal) 10, No 2 (2001)

[22] A. Amin. Off-line arabic character recognition: the state of the art. *Pattern recognition*, 31(5):517–530, 1998. A. Amin. Off-line arabic character recognition: the state of the art. *Pattern recognition*, 31(5):517–530, 1998.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[23] M. Hanmandlu., Murali K.R Mohan, Chakraborty S, Goyal S,Choudhury D.R. (2003) "Unconstrained handwritten character recognition based on fuzzy logic" Pattern Recognition 36 pp. 603 – 623.

[24] Almuallim, Hussein, and Shoichiro Yamaguchi. "A method of recognition of Arabic cursive handwriting." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 5 (1987): 715-722.