Segmentation of Overlapped Handwritten Arabic Sub-Words

Hashem Ghaleb
Dept. of Studies in Computer Science, University of Mysore
Mysore 570006

P. Nagabhushan
Dept. of Studies in Computer Science, University of Mysore
Mysore 570006

Umapada Pal
CVPR Unit
Indian Statistical Institute
Kolkata-108

ABSTRACT

Arabic script is cursive in both handwritten and printed form. Segmentation of Arabic script- especially handwritten- is a very challenging task. Many difficulties arise due to the inherent characteristics of Arabic writing such as the overlapping of Arabic sub-words wherein the sub-words share the same vertical space, and vertical ligatures wherein characters are stacked upon each other in a word. In this paper, an algorithm to resolve the overlapping of handwritten Arabic sub-words is introduced. The proposed algorithm is based on pushing strategy; sub-words are pushed in order to obtain a clear vertical cut separating the sub-words. The proposed algorithm was tested using handwritten text selected from four different datasets and the results are quite promising.

General Terms

Handwritten Arabic segmentation.

Keywords

Arabic sub-words, Overlapping Arabic sub-words, Resolving overlapped Arabic sub-words.

1. INTRODUCTION

Arabic is written from right to left and it consists of 28 letters. Arabic letters-along with few additional letters- are used to transcribe several languages including Arabic, Persian and Urdu. Each letter has two to four shapes based on its position in the word (beginning, middle, end, or isolated). There exist several letters which share the same shape but differ only in number and position of dots. Additional small markings, called "diacritical marks" or "diacritics", are used to represent short vowels or other sounds. They are normally omitted from handwriting.

Letters are joined to form a word. However, there exist six letters (ا, د, ذ, ر, ز, and و) which are not joined to the letter succeeding them. When any of these letters is encountered in the beginning or middle of the word, it separates the word into many pieces (see Fig.1). Each piece is called a sub-word (it is frequently called piece of Arabic word; abbreviated as PAW). Three Arabic words are presented in Fig. 1. The first word (Fig. 1.a) consists of one PAW, the second (Fig 1. b) consists of two PAWs, and the third (Fig. 1.c) consists of three PAWs. In the literature, dots and diacritics are called "secondary components" whereas the main body of the sub-word is called the "primary component". Sub-words may horizontally overlap; i.e., share the same vertical space (see Fig.2, overlapping is highlighted by encircling). This overlapping is called "inter sub-word overlapping" and it induces problems for both the word and the character segmentation [1] [6].

Ideally, the main body of a sub-word is connected. However, in handwritten script a sub-word (more specifically the primary component comprising a sub-word) may be disconnected due to pen lifting. This may introduce an overlapping within the body of the sub-word; i.e., "intra sub-word" overlapping which further complicates the problem (see fig.2,a, intra sub-word overlapping is indicated using a rectangle).

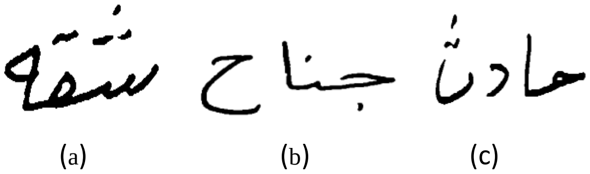


Figure 1. Examples of Arabic words.

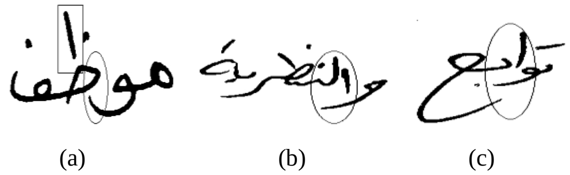


Figure 2. Overlapping Arabic sub-words.

Researcher are convinced that extrapolating successful methods from Western OCR is insufficient for Arabic due to its own characteristics [16][17]. In this regard the specific characteristic of Arabic sub-words overlapping is addressed in this paper with the goal of clearly segmenting such overlapped sub-words.

The rest of the paper is organized as follows; related literature is presented in section 2. The different stages of the proposed algorithm are presented in section 3. The experimental results are reported in section 4. Comparative results are reported in section 5. Finally, we conclude the paper in section 6.

2. RELATED LITERATURE

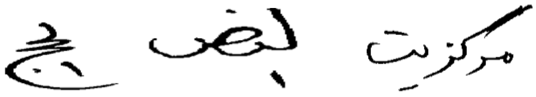
Numerous research results in the field of handwriting recognition have been reported over the last few dacades, especially for Latin and Chinese scripts. However, the state of the art for Arabic handwritten text recognition falls far behind [4][13]. Most of the work in Arabic handwriting recognition has dealt with character, digit or word recognition. Most of these works are hidden Markov model (HMM) based Arabic handwritten word recognition [13]. This situation has two aspects: most of the current recognition systems are word recognition systems mainly because they were designed and tested on databases cosisting of word images. However, the extraction of a word -which is a crucial step for word recognition systems- from Arabic textline is a very challenging task due to sub-word phenomena (a research group in CEDAR has obtained only 60% success rate for word segmentation[15]). The second aspect is the usage of HMM models in order to avoid character segmentation step which is a very difficult task to be accomplished due to both sub-words overlapping and vertical ligatures. According to [13], only one group of researchers has reported results on

Arabic handwritten text recognition on a full page. There still remains a gap in research for developing of OCR systems that work on a full Arabic text page. To accomplish such task the segmentation of Arabic handwritten text is a major challenge which needs to be intesivelly studied. As brought out in the previous section (earlier paragraph) overlapping of Arabic sub-words can be one of the major reasons the complicates the segmentation problem.

There exist many segmentation algorithms in the literature which address the issue of segmenting Arabic handwritten text (word) into characters. In [13], the valleys in the polygonal approximation of Arabic sub-word are utilized for segmentation. Collinear-points suppression technique is used to avoid finding false valleys. In addition, several rules are used to find segmentation points which are not located in valleys. A multi-phase segmentation approach is introduced in [3, 12]. It starts by detecting and resolving sub-word overlaps, then a large number of candidate segmentation points is identified on the thinned version of input image; each column which contains only one foreground pixel is considered to be a candidate segmentation point. Later on, these points are reduced using a set of heuristic rules which utilize structural features (loops, dots, end points and branch points) to obtain the (actual) segmentation points. The segmentation technique proposed in [10, 11] is based on modified vertical histogram which is obtained after removing dots, ascenders and descenders, and applying thinning operation. Modified vertical histogram is calculated based on the distance between top and bottom foreground pixels for each column. Minima in the histogram indicate prospective segmentation points. The best segmentation points are selected based on the analysis of distance between successive prospective segmentation points. The algorithm proposed in [14] first traces the baseline of the input text-line image and straightens it. Subsequently, it over-segments each word/sub-words using features extracted from histogram analysis and then removes extra segmentation points using some baseline dependent as well as language dependent rules.

In spite of the hinderance caused by the sub-words overlapping phenomena affecting character segmentation, to the best of our knowledge there exist only few attempts which addressed the overlapping of Arabic sub-words. Cheung et al [1] is one of the earliest works which addressed this problem. They utilized a specific characteristic of Arabic writing; that is Arabic is written from right to left. As a result overlapping is expected to occur only between the left-hand side contour of the word and the right-hand side contour of the succeeding word. Resolving the overlapping was accomplished in two stages. In the first stage, the beginning of the word is determined by computing of the vertical projection at every column of the image starting at rightmost column of the image. As soon as a column wherein forground pixels appear is encountered, it is considered as the beginning of the word. Later, the left-hand side contour of the right sub-word and the right-hand side contour of the succeeding word is traced in order to resolve the overlapping and obtain the segmentation path. The proposed algorithm was successful as the scope of its usage is the recognition of printed (and hence more uniform) text. However, this algorithm may have some limitations in case of handwritten text due to the variability of handwritten text. For example, this algorithm may suffer in many cases in identifying the beginning of the word. (see Fig. 3.a and b).

N. Farah et al [2] have used boundary following algorithm to solve sub-words problem. A recent work [3], has proposed solution to the sub-word overlapping problem based on the classification of (word) image components into main components (those which intersect with the baseline) and secondary components (dots and diacritics). They proposed a solution based on conducting a distance analysis on the bounding boxes of the main components along the x-axis in order to identify the baseline overlapped main components and their corresponding distance. This operation is followed by applying a set of rules to associate secondary components with their corresponding primary componets. This algorithm suffers the limitation of being baseline-dependent (i.e., the overlapping will be resolved only for the components which touch the baseline). However, in some cases the main body of the sub-word may not intersect with the baseline, and hence any overlapping caused by such component would not be resolved. It also considers the x-axis coordinates as the criteria for the resolving. Such situation is not the only possible situation; sometimes a preceding sub-word appears to the left of a succeeding sub-word, i.e., out-of-sequence sub-words (see Fig. 3).



(a) (b) (c)

Figure 3. Arabic words; sequence violated.

In this paper, a baseline-independent strategy to resolve the overlapping of handwritten Arabic sub-words is introduced. The proposed algorithm pushes (translates) each sub-word by a suitable amount in order to arrive at overlapping-free image. Unlike the method introduced in [3], where only the horizontal arrangement of bounding boxes of the main components is considered for resolving, the proposed algorithm considers the horizontal arrangement of zones of overlapping along with the vertical arrangement of components in such zones. This enables the algorithm to handle out-of-sequence sub-words efficiently. To the best of our knowledge this work is the first attempt in this regard.

3. METHODOLOGY

The proposed algorithm comprises of three stages: preprocessing, resolving the overlapping of primary components, and associating secondary components with their corresponding primary components. The overlapping-resolution algorithm introduced here assumes that overlapped sub-words are already extracted from textline using a suitable segmentation algorithm. This can be achieved through simple and straightforward vertical projection at textline level, and the analysis of resultant segemnts after the removal of secondary component(s) present in each segment (if any). If number of components is greater than or equal to two, then the segment is considered to be consisting of overlapping sub-words. The work presented in this paper is devoted to resolving the overlapping of Arabic sub-words, and hence the input images used to test the algorithm are randomly seleceted such that they contain overlapped sub-words.

3.1 Preprocessing

This stage aims at removing of the secondary components in order to prepare the input image for the subsequent stages. It commences with the binarization of the input image. To close short gaps and fill small holes the method introduced in [12] is applied. Thereafter, connected component labeling algorithm is applied on the binarized image. Then each component is enclosed in minimum rectangular bounding box.

The removal of secondary components is achieved through the analysis of these boxes; if the width of the bounding box of a component is less than a specific threshold, it is considered to be a secondary component and hence removed (refer experimentation section).

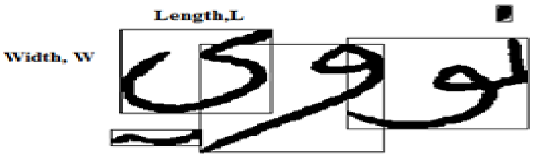


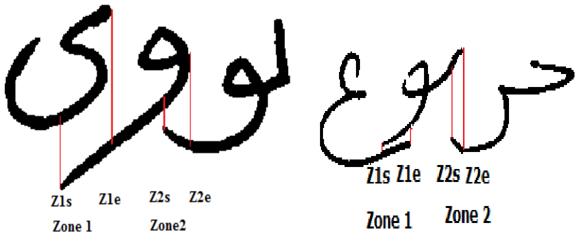
Figure 4. Identification of secondary components

3.2 Resolving the overlapping of primary components

This stage accepts the output of preprocessing stage (an image which contains only primary components), and resolves the overlapping of primary components through two sub-stages: components' overlapping analysis, and component translation.

3.2.1 Components' Overlapping Analysis

The image is scanned vertically and the number of components in each column is obtained. If the number of components is greater than or equal to two, then the column is labeled as "overlapping column". The sequence of consecutive overlapping column forms what we call "overlapping zone". Each overlapping zone, Z, is characterized by it starting point (Zs) and end point (Ze).



(a) **(b)**

Figure 5. Overlapping zones characterization

After obtaining overlapping zones, we obtain what is so called "to retain" and "to shift" components in each overlapping zone according to the following strategy:

The image is scanned vertically from the top at zone start point (Zs), as soon as we encounter foreground component we identify it as "to retain" component. Similarly, the component "to shift" is obtained by scanning the image vertically from the bottom at Zs. A special attention has to be paid for the leftmost zone; after obtaining "to retain" and "to shift" components we identify the row in which the component "to retain" is encountered and scan the image horizontally (towards left). If a foreground component is encountered, the components "to retain" and "to shift" are interchanged (see Fig.5, b).

3.2.2 Components Translation

The core operation of resolving the overlapping of sub-words is carried in this sub-stage. Overlapping zone(s) are sequentially traced starting at the leftmost zone. The component "to retain" is determined in each zone and translated to the right by a specific distance (we call it Absolute Shift Amount *(ASA)*). This amount is computed using two terms: the amount of shift due to overlapping zone (we call it Relative Shift Amount *(RSA)*) and the accumulative shift due to the propagation of shifting distance. i.e., Absolute Shift Amount.

Algorithm 1: ALGO-PUSH

```
Step 1: Accumulative shift = 0
Step 2: Trace overlapping zone(s) from left to right.
Step 3: Obtain the component "to retain"(R) in the current
        zone.
Step 4: Retrieve the RSA of R computed in the algorithm 1.a.
Step 5: ASA =RSA + Accumulative shift
Step 6: Push the component R to the right with the ASA.
Step 7: Accumulative shift = ASA.
Step 8: if current zone is the rightmost zone go to step 9.
          Otherwise, go to step 2.
Step 9: Obtain the component "to shift", S.
Step 10: ASA = RSA + Accumulative shift.
Step 11: Push the component S to the right with the
          ASA.
```

Algorithm 1.a: Computation of relative shift amount

```
Step 1: Create a matrix, Relative Shift Matrix (RSM).
Step 2: Trace the overlapping zone(s) from left to right.
Step 3: Obtain the components "to retain"(R), and "to shift"(S)
Step 4: Obtain end point of R (Re) and start point of S (Ss).
Step 5: Relative Shift Amount =Re – Ss + O. O is
        an offset.
Step 6: Store the component S and the corresponding relative
shift amount in RSM (row-wise).
Step 7: If the current zone is the rightmost zone go to step 8.
Otherwise, go to step 2.
Step 8: Retrace overlapping zone(s) from left to right. if at
any zone, the component R does not appear in RSM, store
the component R in the RSM and set the corresponding shift
to zero.
```

3.3 Association of secondary components with their corresponding primary components

After the overlapping of primary components is resolved, the secondary components are associated with their corresponding primary components according to the following strategy: If a secondary component overlaps with only one primary component, then it is associated with such component. Otherwise, the secondary component is associated with the component "to retain" (R). Finally, each secondary component is translated with the same amount by which it's corresponding primary component was translated

4. EXPERIMENTAL RESULTS

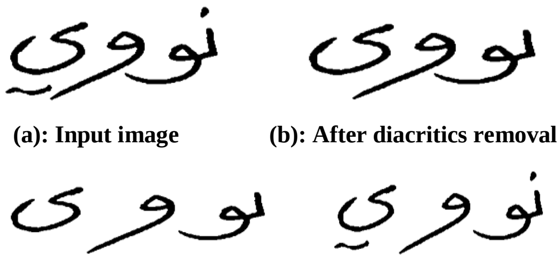
For the purpose of evaluating the proposed algorithm we chose 400 images containing 985 overlapping handwritten sub-words. The images were selected from four handwritten datasets [3][7][8][9] (three databases contain Arabic and the other contains Persian handwriting images; 100 images were selected from each dataset) . We have conducted five different experiments. In the first four experiments the images belonging to a specific dataset is used to find the threshold required for removal of secondary components-which is set as percentage of the average width of the bounding boxes in the image- and the others for evaluating the proposed segmentation algorithm. In the fifth experiment fifty images from each dataset is considered for finding the threshold. The criterion for threshold selection is to maximize the removal of secondary components while minimizing the removal of primary components. Let Sr be the percentage of the secondary components removed, and Pr the percentage of the primary components removed. The threshold value which maximizes the following formula is chosen:

$$\frac{2 * Sr * (100 - Pr)}{Sr + (100 - Pr)}$$

Table 1. Experimentation Results

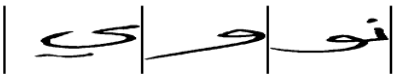
Experiment No.		1	2	3	4	5
No. of Secondary components		185	183	198	229	409
No. of secondary components removed	50%	173	161	170	193	357
	55%	176	167	177	202	369
	60%	**182**	174	181	205	377
	65%	182	**177**	**190**	208	**392**
	70%	182	177	191	**213**	394
No. of primary components		232	276	260	217	493
No. of primary components removed	50%	0	2	0	0	2
	55%	0	5	1	1	5
	60%	**5**	6	2	1	8
	65%	5	**9**	**6**	3	**13**
	70%	5	14	9	**4**	17
Segmentation results	**Perfectly segmented**	84%	85%	87%	84%	86%
	PL	2%	5%	3%	4%	4%
	Secondary not removed	9%	4%	4%	2%	4%
	Primary removed	3%	4%	4%	8%	4%
	DA	2%	2%	2%	2%	2%

Different stages of the proposed algorithm are presented in Fig. 6 (a-d). Finally a vertical line is imposed to indicate vertical cut (Fig. 6, e).



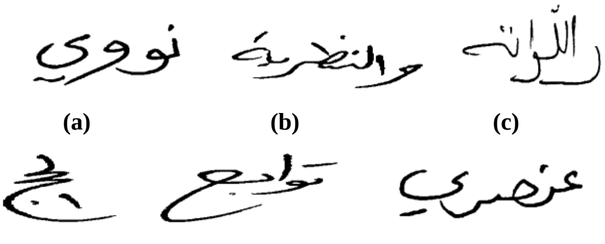
(a): Input image **(b): After diacritics removal**

(c): After resolving primary components **(d): After diacritics Association**



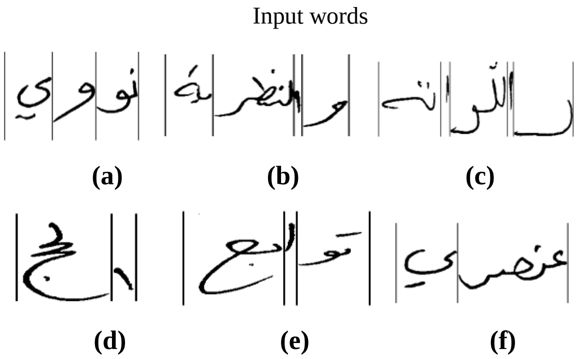
(e): Vertical cuts imposed

Figure 6. Stages of the proposed algorithm



(a) **(b)** **(c)**

(d) **(e)** **(f)**



Input words

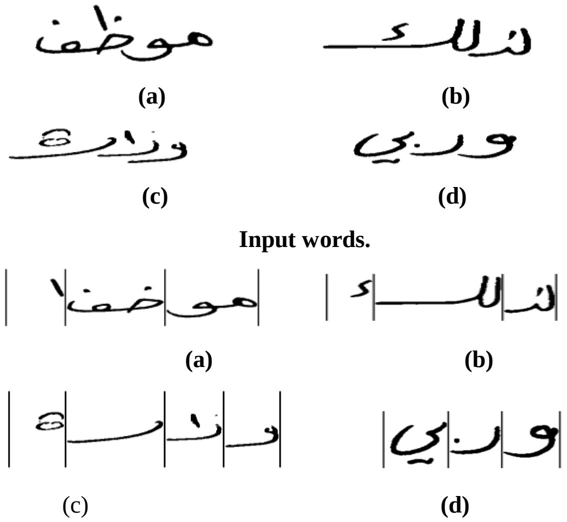
(a) (b) (c)

(d) (e) (f)

Output of the algorithm

Figure 7. Successful results

By manual inspection we found that the algorithm failed in some cases due to different reasons. Firstly, pen lifting (PL) is the source of an inevitable error (see Fig. 7, a). Secondly, in some cases the error occurred due to non-removal of secondary components (see Fig. 7,b) whereas the removal of primary component was the source of errors (see Fig. 7,c). Finally, some the secondary component (DA) was not associated correctly with the actual primary component (Fig. 7, d).



(a) (b)

(c) (d)

Input words.

(a) (b)

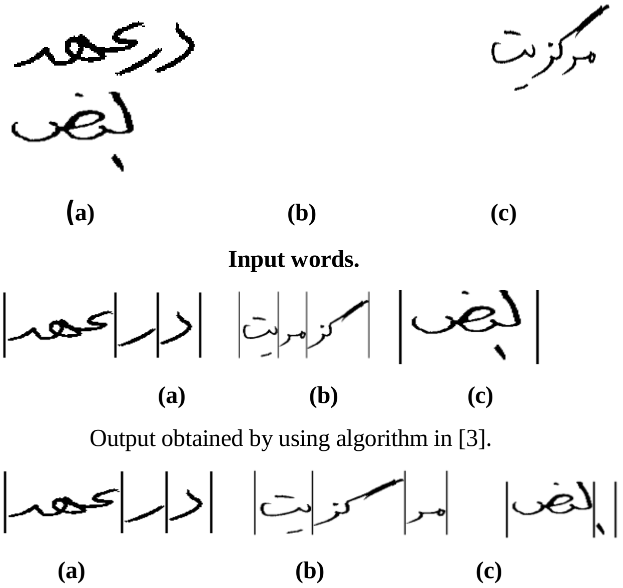
(c) (d)

Output of the algorithm.

Figure 8. Unsuccessful results.

5. COMPARATIVE RESTULTS

The algorithm introduced in [3] was implemented and tested using the same set of handwritten samples. It achieved 72% success rate in segmenting the sub-words. The major source of error is caused by the baseline-dependence nature of the algorithm; in 21% of the samples the algorithm is unable to resolve the overlapping because some primary component(s) do not touch the baseline and hence the overlapping introduced by such components is not resolved. In some cases the error was due to out-of-sequence sub-words (Fig.9, b). The proposed algorithm achieved a minimum of 84% successful sub-word segmentation outperforming state-of-the-art method of [3]. A sample of result obtained by the proposed algorithm and the algorithm of [3] is shown in Fig. 9.



(a) (b) (c)

Input words.

(a) (b) (c)

Output obtained by using algorithm in [3].

(a) (b) (c)

Output obtained by the proposed algorithm.

Figure 9. Comparative segmentation results.

6. CONCLUSION

This paper is an attempt towards segmenting Arabic text at the sub-word level. The approach proposed here is to push the sub-words so that we get a clear vertical separation of the sub-words. The algorithm introduced here is able to handle efficiently out-of-sequence sub-words. The results achieved encouraged us to do more research in this direction.

7. REFERENCES

[1] A. Cheung, M. Bennamoun, and N.W. Bergmann. An Arabic optical character recognition system using recognition-based segmentation, Pattern Recognition, Vol. 34, No. 2, pp.215-233, 2001.

[2] N. Farah, L. Souici, and M. Sellami. .Decision fusion and contextual information for Arabic word recognition for computing and informatics, Computing and Informatics, Vol. 24, No. 5, pp. 463-479, 2012.

[3] M. Elzobi, A. Al-Hamadi , Z. Al Aghbari, and L. Dings. .IESK-ArDB: a database for handwritten Arabic and an optimized topological segmentation approach, International Journal of Document Analysis and Recognition , Vol. 16, No. 3, pp. 295-308, 2013.

[4] L.M. Lorigo and V. Govindaraju. Offline Arabic Handwriting Recognition: A Survey, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 5, pp.712-724, 2007.

[5] A.M. AL-Shatnawi, F.H. AL-Zawaideh, S.AL-Salaimeh, and K.Omar. Offline Arabic Text Recognition – An Overview, World of Computer Science and Information Technology Journal, Vol. 1, No. 5, pp.184-192, 2011.

[6] M. Zand, A.N.Nilchi, and S. A.Monadjemi. Recognition-based Segmentation in Persian Character Recognition, World Academy of Science, Engineering and Technology, Vol. 2, pp.183-187, 2008.

[7] A. Alaei, P. Nagabhushn, and U. Pal. A New Dataset of Persian Handwritten Documents and Its Segmentation, In proceedimgs of 7th Iranian conference on Machine Vision and Image Processing, pp.1-5, 2011.

[8] S.A. Mahmoud, I. Ahmad, W.G. Al-Khatib, and M. Alshayeb. KHATT:An open Arabic offline handwritten text database, Pattern Recognition, Vol. 47, No. 3, pp.1096-112, 2014.

[9] M. Pechwitz, S. S. Maddouri, V. Märgner, N. Ellouze, and H. Amiri. IFN/ENIT- Database of Handwritten Arabic Words, In CIFED : colloque international francophone sur l'écrit et le document, 2002.

[10] H.A. AlHamad and R.A. Zitar. .Development of an efficient neural-based segmentation technique for Arabic handwriting recognition, Pattern Recognition, Vol. 43, No. 8, pp. 2773–2798, 2010.

[11] H.A. AlHamad. Over-Segmentation of Handwriting Arabic Scripts using an Efficient Heuristic Technique, In proceedings of the International Conference on Wavelet Analysis and Pattern Recognition, pp. 180-185, 2012.

[12] M. Elzobi, A. Al-Hamadi, L. Dinges, and B. Michaelis. .A Structural Features Based Segmentation for Off-line Handwritten Arabic Text, In proceedings of 5th Internationl Symposium on I/V Communication and Mobile Network, pp. 1-4, 2010.

[13] M.T. Parvez and S.A. Mahmoud. Arabic handwriting recognition using structural and syntactic pattern attributes, Pattern Recognition, Vol. 46, No. 1, pp. 141-154, 2013.

[14] A. Alaei, P. Nagabhushan and U. Pal. A Baseline Dependent Approach for Persian Handwritten Character Segmentation, In proceeding of the twentieth International Conference On Pattern Recognition, pp. 1977-1980, 2010.

[15] S.N. Srihari, G.R. Ball and H. Srinivasan. Versatile Search of Scanned Arabic Handwriting, In Arabic and Chinese Handwritten Recognition Summit, SACH 06, pp. 57-69, 2006.

[16] D. Lopresti, G. Nagy, S. Seth, and X. Zhang. Multi-Character field recognition for Arabic and Chinese handwriting, In Arabic and Chinese Handwritten Recognition Summit, SACH 06, pp. 93-100, 2006.

[17] A. Zidouri. ORAN: a basis for an Arabic OCR system, In proceedings of International Symposium on Intelligent Media, Video, and Speech Processing, pp. 703-706, 2004.