

Detection and Compensation of Undesirable Discontinuities within the Farsi/Arabic Subwords

Majid Ziaratban and Karim Faez

Electrical Engineering Department, Amirkabir University of Technology, Iran

Abstract: *In this paper, an unexplored subject in the domains of Farsi/Arabic handwritten word preprocessing is introduced. Subwords play a vital role in many applications such as cheque amount recognition, text recognition, lexicon reduction and subword-based word recognition. Correcting the faults occurred in subwords will improve the overall performance of these applications. A subword is a connected-component in the main body of a word. The occurrence of a discontinuity in a subword, divides the subword into two isolated parts. These parts are detected as two incorrect subwords. In our algorithm, before correcting these faults, the baseline of each subword is corrected using the proposed baseline correction method. Then, to limit the exploration area in matching stage, the dots are removed. Undesirable discontinuities in subwords are detected by using a template matching algorithm. Disconnected parts of a subword are joined together by using three different methods. Experiments show that the cubic polynomial-based compensation method causes the best results and 2.87 % improvement in the subword recognition rate.*

Keywords: *Detection, compensation, Farsi/Arabic subword, and cubic polynomial curve fitting.*

Received May 12, 2009; accepted November 5, 2009

1. Introduction

Preprocessing is one of the most important phases in the image processing and pattern recognition applications. In this paper, a subject in the domain of Farsi/Arabic handwritten word preprocessing is introduced. Some special characteristics of Farsi and Arabic scripts make them absolutely different from other scripts. Farsi and Arabic are cursive writing languages. Alphabetic characters contain up to 4 different shapes due to their position in a word. Only one of these 4 shapes is written isolated from the other characters. Two out of them are connected from one side to the preceding or the following letter. The fourth shape is connected from both sides. The word recognition in these scripts can be divided into three main groups: word-based [8], subword-based [9] and character-based [5] word recognition. In the first group, holistic features are extracted from the whole word and sent to a classifier. Since the variety of words is too wide, the first group is limited to small-lexicon applications. In the character-based group, an input word must be segmented into small constructor parts like characters. The most important, time-consuming and complex part of these approaches is the segmentation stage. There is no accurate and reliable segmentation approach for Farsi/Arabic handwritten words. The second group is located between the word-based and character-based methods; because on one hand, the variety of subwords is less than words. On the other hand, the segmentation of a word into subwords can be easily accomplished. A subword consists of some connected letters and is completely

isolated from other characters or subwords. In other words, a subword is a connected-component in the main body of a word. Two main faults occur in subwords:

1. If a discontinuity (DC) occurs in a subword, it will be divided into two isolated parts and incorrectly considered as two subwords.
2. Furthermore, if two successive subwords connect together, it will be erroneously determined as one subword.

Since the performance of the subword detection and recognition directly affects the final word recognition rate, the detection and correction of these faults can improve the final performance. In addition, in the fields of Farsi/Arabic text recognition and check amount recognition [1, 16], the connected components play an important role. Since a Farsi or Arabic words may consist of one or more subwords, in a handwritten text, the beginning and end of a word are not known for OCR systems. In these applications, different combinations of subwords are investigated to find the best combination that builds an expressive word. Error in a subword causes errors in recognizing a number of consecutive words in a text; because, some subwords before and after the correct word may be considered as parts of the word incorrectly. Also, some subwords at the beginning or end of a word may be omitted and considered as the previous or next word erroneously. Furthermore, the number of subwords of a word can be used as a feature for lexicon reduction applications [2, 11, 21, 24]. In these approaches, the words with the specific number of subwords are categorized in a same

class. By occurring faults in subwords, the number of subwords of a word is calculated incorrectly and makes errors in the categorization. To overcome these problems, subwords must be recognized as correct as possible. To recognize subwords correctly, their faults must be detected and corrected. Some studies have been done to segment the connected handwritten numerals [4, 10, 12, 13, 17, 20] and Chinese characters [23]. But the subject of removing an undesirable gap occurring in subwords is a novel view to the word preprocessing and retrieval field and there is no research in this unexplored domain. In this paper, we study on the correction of undesirable discontinuities in Farsi/Arabic sub words. The rest of the paper is organized as follows: Detection and compensation of the discontinuities are described in section 2. A discussion about the experimental results is given in section 3 and finally, the conclusions are drawn in section 4.

2. Proposed Algorithm

























Since our objective is the removing undesirable gaps between two parts of a subword, first of all, some information about the shape, different types and probabilities of occurrence of these gaps in handwritten subwords is required. By performing a study on a set of 2700 real handwritten images, 24 different types of DCs have been extracted and shown in Table 1. These gaps have befallen in 576 subwords (542 images) of the IFN/ENIT dataset [19]. In Table 1, the frequency of a DC indicates its importance. Our algorithm must be able to detect and compensate the DCs with the large number of repetitions more accurately than others. Figure 1 shows five various shapes of the DC of type 3 in different writing styles. The flowchart of the proposed algorithm is illustrated in Figure 2. In this flowchart, the baseline of the input word is estimated and corrected. Having the baseline,

dots and additional diacritics are eliminated to obtain the image, which contains only the body of the subwords. The subwords are indexed from right to left. Each pair of successive subwords are tested to be two parts of a DC or not. If both of them are the parts of a DC, their distance is checked to be lower than the maximum allowable distance. If the distance condition is satisfied, these subwords are connected to each other in the DC compensation phase. More details about the stages of the algorithm are discussed in the following subsections.

2.1. DC Detection

One of the important stages in the compensation of failures in systems is the detection of their occurrence and locating their exact positions. In the detection phase of our algorithm, the positions of DCs are found and sent to the next phase to be compensated. Since a DC consists of two isolated parts, therefore a DC can be detected only when both parts satisfy some conditions. In other words, each part must have some individual characteristics so that the combination of these parts must be in one of 24 types of DCs. Due to this, the first part (right one) is tested to see if it has the characteristics of the first parts of any of the 24 DC types. The indexes of DCs which their first parts have the same properties as the right part in the input image, are stored in array *A*. In the next stage of the detection phase, the second parts (left parts) in the input image and in the DC types, which their indexes are in the array *A*, are compared. The DC with the maximum likeness in the second part is selected as a winner DC (DC_j). Likeness computation is done by searching a special template in an input image. A number of templates for each part of all 24 DC types are extracted from 3200 words of set *b* of the IFN/ENIT dataset. The templates are extracted so that all varieties of each DC in different writing styles are covered.

Table 1. 24 Different DC types with the number of their repetition in a set of 2700 images.

DC Type	Type 1	Type 2	Type 3	Type 4	Type 5	Type 6	Type 7	Type 8
An Instance Image								
Frequency in the Dataset	130	110	81	49	39	39	34	27
DC Type	Type 9	Type 10	Type 11	Type 12	Type 13	Type 14	Type 15	Type 16
An Instance Image								
Frequency in the Dataset	21	16	5	4	3	3	2	2
DC Type	Type 17	Type 18	Type 19	Type 20	Type 21	Type 22	Type 23	Type 24
An Instance Image								
Number of Repetition in the Dataset	2	2	2	1	1	1	1	1

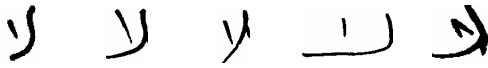


Figure 1. Five shapes of the DC of type 3.

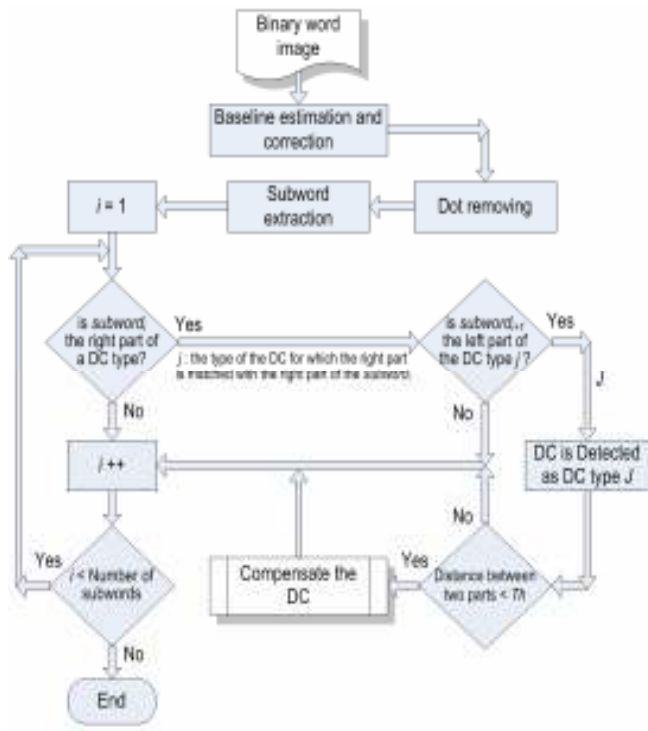


Figure 2. Flowchart of the proposed algorithm.

Before starting the DC detection stage, some processes should be done. In this way, first, the separated parts, called blobs, in an input image must be determined.

2.2. Dot-Removing Stage

Since we want to join the disconnected parts of a subword together, dots and additional small markings called diacritics [7] are not important for us; because they must be written isolated from other parts of a word. Hence, dots and isolated diacritics should be removed to reduce the complexity of the searching process for DC parts. In Farsi and Arabic scripts, dots are written at the top or bottom side of the main body of a word. The main body is usually written on the baseline. On the other hand, most of the disconnected parts of DCs, which must be joined together, are located on or close to the baseline of a word. Therefore, the word baseline estimation plays an important role in the DC detection stage. After the baseline estimation, each blob which satisfies one of the following conditions is labeled as a dot or diacritic and must be deleted. Otherwise, it is considered as a subword. These conditions are:

$$y_{k,max} < y_{Baseline} - Y_1 \quad (1)$$

$$y_{k,min} > y_{Baseline} + Y_2 \quad (2)$$

where $y_{k,min}$ and $y_{k,max}$ are the minimum and maximum vertical coordinates of the k -th blob's foreground

pixels. Their values are set in the experimental result section.

2.3. Baseline Estimation

In some scripts such as English, the word's baseline can be extracted accurately using the horizontal projection of a word image [3]. In Farsi and Arabic words, this approach does not work as good as in English. However, in most studies it has been used because of its simplicity. In [14] the estimation was done more accurately using some important pixels. In our approach, to achieve better estimations, some key areas are found in an input image. Based on the coordinates of these areas and some conditions, the best location for the baseline is estimated. The areas are found by searching for the pieces in the word image which are matched with some specific templates. The matching process is done by calculating the correlation between the template and input image pieces. The center of the matched piece is marked as a candidate baseline pixel where t indicates the index of the template.

2.3.1. Template Selection

To find better key areas, the templates must include some properties as follows:

1. Should be found in most of handwritten words.
2. Should be located on or close to the correct baseline.
3. The number of successful matching occurring close to the correct baseline must be much greater than those that are far from it.

86 small pieces close to the correct baseline are extracted from 1000 different subwords of set_b of the IFN/ENIT dataset as the primary templates. A set of 3380 subwords (different from the set used for template extraction) are utilized to determine which of these templates are suitable for the baseline estimation. For each subword in this set, the i -th template, T_i , is checked to be matched with any pieces of the subword. The number of successful matchings in this set for T_i is calculated as N_i . Let $P_{i,k}$ be the piece in the k -th subword which is matched with T_i . $d_{i,k}$ is defined as the vertical distance between the correct baseline of the k -th subword and the center of $P_{i,k}$. Now, for the template T_i , the value of d_i can be computed from:

$$d_i = \text{mean}(d_{i,k}) \quad (3)$$

d_i shows the average vertical distance of the centers of the pieces which are matched with T_i from the correct baseline. A fitness function based on the above conditions is defined as follows:

$$\text{fitness} = \frac{N_i}{(k + d_i)^c} \quad (4)$$

where k is a small constant value that prevents the denominator from being zero and c controls the importance of d_i against N_i . In our application, d_i is more important than N_i . Consequently, the value of c should be selected greater than 1. The value of the *fitness* is large when the number of successful matching is large and the average distance of the location of these matchings from the baseline is small. The values of k and c are experimentally set to 1 and 4.

Nine templates for which the value of the *fitness* is greater than that of the others are selected for the baseline estimation stage. They are illustrated in Figure 3. The white and black pixels are foreground and background pixels respectively. The most important advantage of the proposed baseline estimation algorithm rather than the horizontal projection-based approaches is that the baseline estimation and correction for each subword of a word can be performed separately. While in the horizontal projection-based methods, a number of subwords are needed to estimate an accurate baseline. Another prominence of our algorithm is its greater accuracy in estimating the baseline for word or subword for which there is no certain maximum value in the horizontal

projection. For example, in Figure 4, the horizontal projection profile of the fourth subword (from right) is a smooth profile and contains no certain peak value. The detected key area in this subword indicates the correct baseline of the subword. The horizontal projection and centers of the detected key areas for each subword of the instance word “دروازه” is shown in Figure 4. The instance word includes six subwords and one dot. Subwords are indexed from right to left. Figure 5 shows the results of the word’s baseline correction obtained by using a horizontal projection-based approach (red line) and the proposed method (blue line). In Figure 6, the baselines are estimated and corrected for each subword. It is clearly observed that our proposed method presents more accurate baseline estimation and correction results not only for the whole word as shown in Figure 5, but also for each subword as shown in Figure 6, rather than the projection-based approach.



Figure 3. Nine templates for finding key areas.

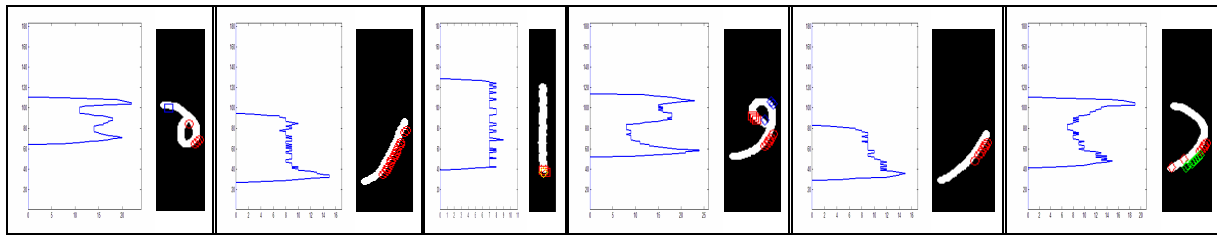


Figure 4. Horizontal projection (left images) and centres of detected key areas in each subword (right images).



Figure 5. Baseline estimation results obtained from the whole word image.

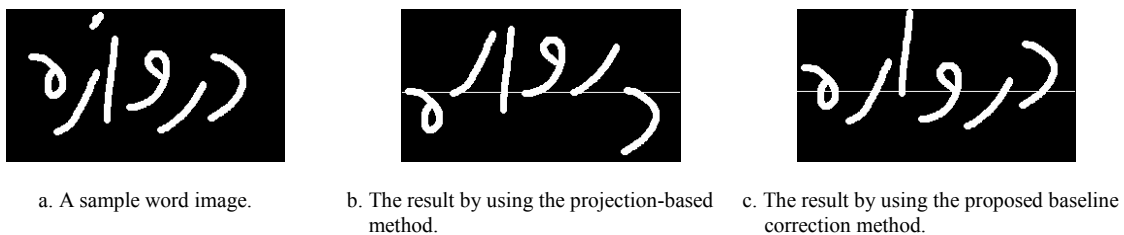


Figure 6. Baseline correction for each subword.

The reason is that the projection-based method performs the estimation blindly. In our method, the baseline is estimated only based on the detected key areas which often occur close to the baseline.

2.4. DC Compensation

In the compensation phase, the algorithm connects the right and left parts of a DC together with respect to J (the type of the detected DC). The compensation is

done only when the distance between DC parts is small. In the following section, a discussion about the suitable value for the allowable distance between two parts of a DC will be given. The center of the best matched piece of the image in the left (right) part of a DC, which is matched with the templates of the left (right) part of the DC of type J , is selected and marked as cp_L (cp_R). cp_R and cp_L are the pixels, respectively, in the right and left part of a DC, which must be connected together.

2.4.1. Compensation Methods

To join these pixels together, three approaches can be used as follows:

Method 1: One of these two parts is shifted by $(\Delta x, \Delta y)$ where Δx and Δy are the differences between the coordinates of cp_R and cp_L . In our algorithm, the left part is shifted. Consequently we have:

$$\Delta x = x_{cp_R} - x_{cp_L} \tag{5}$$

$$\Delta y = y_{cp_R} - y_{cp_L} \tag{6}$$

The new coordinates of the left part pixels are calculated by:

$$x_{L,new} = x_{L,old} + \Delta x \tag{7}$$

$$y_{L,new} = y_{L,old} + \Delta y \tag{8}$$

Method 2: cp_R and cp_L are connected together with a straight line.

Method 3: cp_R and cp_L are connected together with a curved line. The curvature can be drawn by any curve fitting algorithms such as nonlinear, spline, Gaussian, interpolant and polynomial fitting [7].

Figure 7 shows the results of the above methods. The original image of the subword "لنجا" of the word "النجاح" is depicted in Figure 7(a). A DC of type 2 has occurred because of the writing style of its writer. This DC segmented the subword into two parts, "لن" and "جا". The connections resulted by three above compensation methods are shown in Figure 7 (b-d). In *Method 3*, the curve fitting process is done by using a cubic polynomial-based fitting approach. In the polynomial algorithms, a curved line is fitted to points by a series of polynomial terms as:

$$f(x) = \sum_{i=0}^n p_i x^i \tag{9}$$

where $(x, f(x))$ is the coordinate of each pixel on the fitted line and p_i is the i -th polynomial coefficient. The cubic polynomial fitting is a polynomial algorithm of degree three, hence $n=3$. This method is selected among other fitting algorithms because of its good and smooth fitting results with low complexities.

To have a better fitted curve, a large number of the fitting points should exist. We have only two points.

Some additional pixels are added and utilized to improve the accuracy and smoothness of the curve fitting results. Thus, the fitting points consist of cp_R , cp_L and five additional pixels, q_i ($i=1$ to 5).

To determine these additional points, the skeleton of the right part of the DC is traced from the cp_R to the right side. After crossing each six foreground pixels, the current pixel is marked as an additional point. This procedure is repeated five times. The curved line should be fitted to these seven pixels ($q_5, q_4, q_3, q_2, q_1, cp_R$ and cp_L). The reason behind selecting the additional points in the right part is that the writing direction in Farsi and Arabic languages is from right to left. Thus, we should follow the writing path from right to left. The polynomial coefficients for the instance image shown in Figure 7(d) are calculated and given in Table 2. The fitted curve is illustrated in Figure 8. The fitted curve in Figure 8(b) is an accurate estimation of the writing path from the right part of the subword to the left part.

The straight and curved lines resulted respectively by *Method 2* and *Method 3* are expanded to obtain a wider lines. The expansion is carried out by:

$$EL = L \oplus SE \tag{10}$$

where L , EL and SE are respectively the resulted line, the expanded line and the structure element of the morphological dilation operation \oplus .

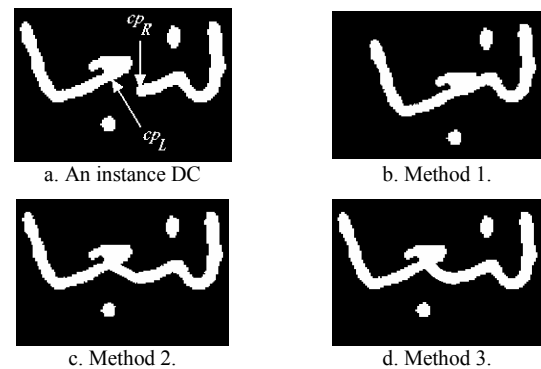


Figure 7. DC compensation results by using three proposed methods.

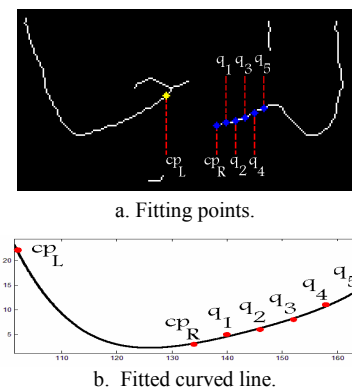


Figure 8. Curve fitting process in Method 3.

Table 2. Calculated polynomial coefficients.

P ₀	P ₁	P ₂	P ₃
577.2	-13.03	0.08304	-0.0001702

SE is a disk with the diameter value equal to ps . ps is the pen size in the word image. To have the expanded curved line similar to the other lines of the word, the pen size should be estimated correctly.

2.4.2. Pen Size Estimation

To estimate the pen size, the run lengths of the foreground pixels in each column of a word image is calculated. After each run, the value of the cell of the array S , corresponding to the run length value, is increased by one. The procedure continues until all columns are processed. The pen size is determined from the Array S by:

$$i = \arg \max_m \{S(m)\}, m=1,2,\dots,30 \quad (11)$$

$$ps = S(i) \quad (12)$$

3. Experimental Results

In the experiments, the first 2700 images of set a of IFN/ENIT dataset were used. This subset included 11871 subwords. Totally 576 DCs were observed in 542 images. This means 20.07% of the images and 4.85% of subwords contained undesirable DCs. It demonstrates the importance of compensating these faults.

3.1. Setting the Values of Y_1 and Y_2

An experiment was performed to determine the proper values of Y_1 and Y_2 in the dot removing stage. Figure 9 shows the elimination rate of dots and subwords. In this figure, it can be observed that by selecting very small values for Y_1 and Y_2 , however most of dots are removed, but many subwords are eliminated incorrectly. On the contrary, for very large values of Y_1 and Y_2 , the rate of dot removing will strongly reduce. The value of Y_1 and Y_2 should be selected so that the maximum dot removing rate and minimum subword removing rate could be obtained. However, our priority was the retaining subwords rather than eliminating the dots. From this figure, the value of Y_1 and Y_2 were chosen as 14 and 22 pixels, respectively.

3.2. Setting the Admissible Distance Between DC Parts

In the second experiment, the admissible distance between two parts of a DC (in the DC compensation phase) was investigated. By choosing a small value for the allowable distance, many correctly detected DCs were erroneously rejected. On the contrary, selecting a large value for it, many incorrectly detected DCs were

allowed to be compensated. From Figure 10, the value of the admissible distance between the two parts of a DC was considered as 10 pixels.

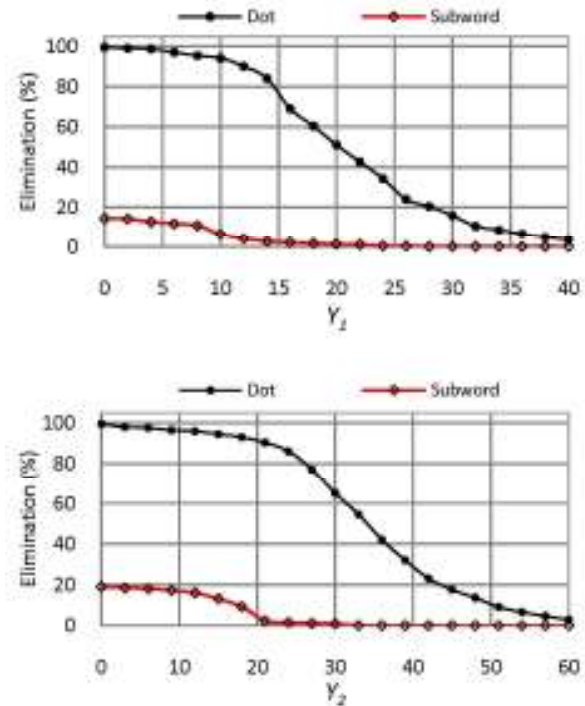


Figure 9. Sensitivity of the dot removing algorithm to Y_1 and Y_2 . 'Subword' and 'Dot' correspond to elimination of subwords and dots, respectively.

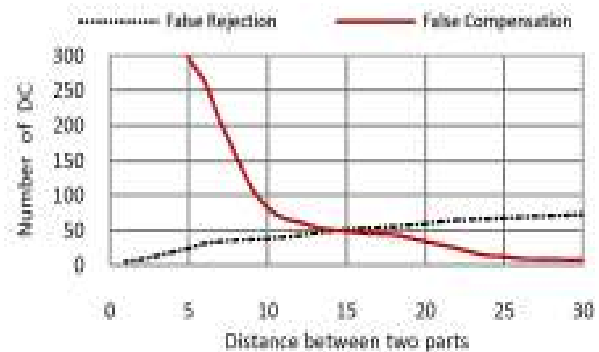


Figure 10. Number of false rejected DC and Number of false compensated DC versus the allowable distance between two parts of a DC.

3.3. DC Detection and Compensation Results

In the DC detection phase, results showed that the proposed method correctly detected 543 out of 576 DCs. For 82 DCs of correctly detected ones, the algorithm rejected their compensations; because of long distance between their left and right parts. The rest (461 DCs) was compensated truly.

Furthermore, the algorithm detected 74 pairs of consecutive subwords erroneously as DCs. 38 out of these incorrect DCs were joined together and caused mistakes. To improve the performance, a recognition-based compensation method was implemented. In this algorithm, after detecting a DC, two isolated subwords before joining, and the combined subword after that

were recognized. For each of these three subwords, the minimum distance from the closest trained subword was calculated.

The compensation was done only when:

$$Dist_{comb} < \frac{Dist_R + Dist_L}{2} \quad (13)$$

where $Dist_{comb}$ is the Mahalanobis distance of the combined subword (after the DC compensation) from the closest train sample. $Dist_R$ and $Dist_L$ are respectively the distances of the right and the left subwords (before the DC compensation) from their corresponding closest train samples. Doing that, 29 out of 38 incorrect compensations were not accepted and only 9 out of them remained.

Thanks to the better results of support vector machines (SVM) as a classifier in various applications, we used it to obtain the better results. In the correction-based compensation phase, using the SVM method, the compensation was done if:

$$Conf_{comb} > \frac{Conf_R + Conf_L}{2} \quad (14)$$

where $Conf_{comb}$ is the SVM confidence value of the combined subword (after the DC compensation). $Conf_R$ and $Conf_L$ are respectively the confidence values of the right and the left subwords (before the DC compensation). By using SVM as the recognizer, only 7 out of 38 incorrect compensations were remained.

3.4. Subword Recognition Improvement

An approach based on the M-band packet wavelet transform proposed in [3] was used to recognize the words. In this approach, the extracted wavelet coefficients were rotation and scale invariant. A set of energy features was computed and extracted from each sub-band of these coefficients. The Mahalanobis classifier [14] was used in [3] for classification. The support vector machines and Mahalanobis are used as classifiers in our experiments. To recognize by the SVM classifier, the LibSVM [6] was used. LibSVM is a library for support vector machines.

Broumandnia *et al.* showed that their proposed approach presented better results than other methods such as approaches based on Fourier-wavelet and Zernike moments in the recognition of Farsi handwritten words. In our experiments, the 3-band wavelet transform and feature vectors of size 96 were used the same as in [3]. Other detail parameters were chosen the same as in [3]. More details about this word recognition algorithm can be found in [3]. Wavelet features were extracted from subwords.

The classifiers was trained with the rest of the images of set *a* of IFN/ENIT (6537-2700=3837 images and 16640 subwords). The performance improvements achieved by using three compensation methods are reported in Table 3 and Table 4,

respectively, by using the Mahalanobis and the SVM classifiers.

In the subword recognition phase, before the DC compensation, all 576 disconnected subwords were recognized erroneously. After the DC compensation by using the cubic polynomial algorithm, 348 out of 576 disconnected subwords were correctly recognized using the SVM classifier.

Table 3. Amounts of performance improvement achieved by using three compensation methods and the Mahalanobis classifier.

Compensation method	Method 1	Method 2	Method 3
# of true recognitions after the compensation of 461 detected DCs	248	312	337
Amounts of improvement in subword recognition rate	$\frac{248-9}{11871} = 2.01\%$	$\frac{312-9}{11871} = 2.55\%$	$\frac{337-9}{11871} = 2.76\%$

Table 4. Amounts of performance improvement achieved by using three compensation methods and the SVM classifier

Compensation method	Method 1	Method 2	Method 3
# of true recognitions after the compensation of 461 detected DCs	256	321	348
Amounts of improvement in subword recognition rate	$\frac{256-7}{11871} = 2.10\%$	$\frac{321-7}{11871} = 2.65\%$	$\frac{348-7}{11871} = 2.87\%$

80.03% of all 576 DCs were compensated. 38 pairs of subwords were incorrectly connected together. By implementing the recognition-based compensation, using the Mahalanobis and the SVM classifiers, 29 and 31 out of these 38 false compensations were corrected, respectively. With respect to only 7 remaining false compensations, 2.87% improvement in subword recognition rate was achieved by using the proposed cubic polynomial-based compensation method and the SVM classifier.

4. Conclusions

In this paper, a novel subject was introduced in the domain of the word preprocessing. About 20% of the images in the dataset contained at least one undesirable discontinuity. It demonstrates the importance of the correcting these faults. Furthermore, a new algorithm for baseline estimation and correction was proposed. Based on the corrected baseline, dots and isolated diacritics were removed. Detection of gaps was done by using a template matching-based algorithm. Experimental results showed that the compensation of gaps by the cubic polynomial curve fitting-based approach was performed better than other methods.

Acknowledgements

This research was supported by Iranian Telecommunication Research Center (ITRC).

References

- [1] Al-A'ali M. and Ahmad J., "Optical Character Recognition System for Arabic Text Using Cursive Multi-Directional Approach," *Journal of Computer Science*, vol. 3, no. 7, pp. 549-555, 2007.
- [2] Alohalı Y., Cheriet M., and Suen Y., "Databases for Recognition of Handwritten Arabic Cheques," *Pattern Recognition*, vol. 36, no. 1, pp. 111-121, 2003.
- [3] Broumandnia A., Shanbehzadeh J., Varnoosfaderani M., "Persian/Arabic Handwritten Word Recognition Using M-band Packet Wavelet Transform," *Image and Vision Computing*, vol. 26, no. 6, pp. 829-842, 2008.
- [4] Chen K. and Wang F., "Segmentation of Single- or Multiple-Touching Handwritten Numeral String Using Background and Foreground Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1304-1317, 2000.
- [5] Cheung A., Bennamoun M., and Bergmann W., "An Arabic Optical Character Recognition System Using Recognition-Based Segmentation," *Pattern Recognition*, vol. 34, no. 2, pp. 215-233, 2001.
- [6] Chih-Chung C. and Chih-Jen L., *LIBSVM: A Library for Support Vector Machines*, Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>, 2007.
- [7] Daniel C. and Wood F., *Fitting Equations to Data*, John Wiley & Sons, New York, 1980.
- [8] Dehghan M., Faez K., Ahmadi M., and Shridhar M., "Handwritten Farsi (Arabic) Word Recognition: A Holistic Approach Using Discrete HMM," *Pattern Recognition*, vol. 34, no. 5, pp. 1057-1065, 2001.
- [9] Ebrahimi A. and Kabir E., "A Two Step Method for the Recognition of Printed Subwords," *Iranian Journal of Electrical and Computer Engineering*, vol. 2, no. 2, pp. 57-62, 2005.
- [10] Elnagar A. and Alhadj R., "Segmentation of Connected Handwritten Numeral Strings," *Pattern Recognition*, vol. 36, pp. 625-634, 2003.
- [11] Farah N., Khadir T., and Sellami M., "Artificial Neural Network Fusion: Application to Arabic Words Recognition," in *Proceedings of European Symposium on Artificial Neural Networks*, Bruges, Belgium, pp. 151-156, 2005.
- [12] Hu J. and Yan H., "A Model-Based Segmentation Method for Handwritten Numeral Strings," *Computer Vision and Image Understanding*, vol. 70, no. 3, pp. 383-403, 1998.
- [13] Kim K., Kim H., and Suen Y., "Segmentation-Based Recognition of Handwritten Touching Pairs of Digits Using Structural Features," *Pattern Recognition Letters*, vol. 23, pp. 13-24, 2002.
- [14] Krzanowski W., *Principles of Multivariate Analysis*, Oxford University Press, 1988.
- [15] Lorigo M. and Govindaraju V., "Off-line Arabic Handwriting Recognition: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp. 712-724, 2006.
- [16] Menhaj B. and Adab M., "Simultaneous Segmentation and Recognition of Farsi/Latin Printedtexts with MLP," in *Proceedings of International Joint Conference on Neural Networks*, pp. 1534-1539, 2002.
- [17] Pal U., Belaid A., and Choisy C., "Touching Numeral Segmentation Using Water Reservoir Concept," *Pattern Recognition Letters*, vol. 24, pp. 261-272, 2003.
- [18] Pechwitz M. and Märgner V., "Baseline Estimation for Arabic Handwritten Words," in *Proceedings of International Workshop on Frontiers in Handwriting Recognition*, pp. 479-484, 2002.
- [19] Pechwitz, M., Maddouri S., Maergner V., Ellouze N., and Amiri H., "IFN/ENIT-Database of Handwritten Arabic Words," in *Proceedings of CIPED'02*, pp. 129-136, 2002.
- [20] Sadri J., Suen C., and Bui T., "A Genetic Framework Using Contextual Knowledge for Segmentation and Recognition of Handwritten Numeral Strings," *Pattern Recognition*, vol. 40, no. 3, pp. 898-919, 2007.
- [21] Souici-Meslati L. and Sellami M., "A hybrid approach for Arabic literal amounts recognition," *the Arabian Journal for Science and Engineering*, vol. 29, no. 2B, pp. 177-194, 2004.
- [22] Zahour A., Taconet B., Mercy P., and Ramdane S., "Arabic Hand-Written Text-Line Extraction," *ICDAR'01*, pp.281-285, 2001.
- [23] Zhao S., Chi Z., Shi P., and Yan H., "Two-Stage Segmentation of Unconstrained Handwritten Chinese Characters," *Pattern Recognition*, vol. 36, no. 1, pp. 145-156, 2003.
- [24] Ziaratban M., Faez K., and Ezoji M., "Use of Legal Amount to Confirm or Correct the Courtesy Amount on Farsi Bank Checks," in *proceeding of International Conference on Document Analysis and Recognition*, pp. 1123-1127, 2007.



Majid Ziaratban received BSc and MSc degree in electronic engineering from Guilan University in 2002 and Amirkabir University of Technology in 2005, respectively. Currently, he is a PhD student at the Electrical Engineering Department of Amirkabir University of Technology, Tehran, Iran. His research interests include Farsi and Arabic document analysis and recognition, computer vision, and pattern recognition.



Karim Faez received his BS degree in electrical engineering from Tehran Polytechnic University as the first rank in June 1973, and his MS and PhD degrees in computer science from University of California at Los Angeles (UCLA) in 1977 and 1980, respectively. Prof. Faez was with Iran Telecommunication Research Center (1981-1983) before joining Amirkabir University of Technology in Iran. He was the founder of the Computer Engineering Department of Amirkabir University in 1989 and he has served as the first chairman during 1989-1992. Professor Faez was the chairman of planning committee for Computer Engineering and Computer Science of Ministry of Science, research and Technology (during 1988-1996). His research interests are in pattern recognition, image processing, neural networks, signal processing, farsi handwritten recognition, earthquake signal processing, fault tolerant system design, computer networks, and hardware design. He is a member of IEEE and ACM.