End-Shape Recognition for Arabic Handwritten Text Segmentation

Amani T. Jamal, Nicola Nobile, and Ching Y. Suen

CENPARMI (Centre for Pattern Recognition and Machine Intelligence)
Computer Science and Software Engineering Department, Concordia University
Montreal, Quebec, Canada
{am_jamal,nicola,suen}@cenparmi.concordia.ca

Abstract. Text segmentation is an essential pre-processing stage for many systems such as text recognition and word spotting. However, few methods have been published for Arabic text segmentation. In Arabic handwritten documents, separating text into words is challenging due to the enormous different Arabic handwriting styles. In this paper, we present a new segmentation methodology of an Arabic handwritten text line into words. Our proposed approach of text segmentation utilizes the knowledge of Arabic writing characteristics. This method shows promising results.

Keywords: component, Arabic Handwritten Documents, segmentation, End-Shape recognition.

1 Introduction

Extracting all the word images from a handwritten document is an essential pre-processing step for two reasons [1]. First, for text recognition methods, which can be categorized into letter-based and word-based, there is a need to work on pre-extracted word images. Secondly, for word-spotting or content-based image retrieval techniques, all the word images in the documents are required to be pre-segmented properly. Most of the techniques in handwritten document retrieval and recognition fail if the texts are wrongly segmented into words.

Few methods have been published for Arabic text segmentation. In Arabic handwritten documents, separating text into words is challenging due to the enormous different Arabic handwriting styles. In this paper, we present a new segmentation methodology of an Arabic handwritten text line into words. Our proposed approach of text segmentation utilizes the knowledge of Arabic writing characteristics.

In this Section, we provide some background of the Arabic characteristics and the previous works of text line segmentation into words. In addition, the challenges of Arabic handwritten text segmentation are given. Finally, the proposed approach is summarized with the rational of applying it and our overall methodology is explained. The secondary component removal technique is briefly explained in Section 2. The used metric-based segmentation method is explained in Section 3. The contribution of this paper is described in Section 4. The experiment is explained in Section 5. Finally the conclusion is given in Section 6.

N. El Gayar et al. (Eds.): ANNPR 2014, LNAI 8774, pp. 228–239, 2014.

1.1 Arabic Characteristics

In the Arabic script, there is a major characteristic that differentiates this language from Latin-based ones. Twenty-two letters in the Arabic language must be connected on a baseline within a word. The remaining six letters cannot be connected from the left, which we call non-left-connected (NLC) letters. In this way, NLC letters separate a word into several parts depending on how many of these letters are included in a word. In other words, NLC letters indicate a separation of Part of Arabic Word (PAW). A study shows that NLC letters represent 33% of the text [2]. The Arabic script is considered as semi-cursive [4] since each word may be composed of one or more sub-words or (PAWs). In [5], a sub-word is defined "as being a connected entity of one or several characters belonging to the word". Figure 1 shows one word with two PAWs.

Fig. 1. An Arabic word with two PAWs

1.2 Challenges

Arabic texts have two types of spacing, intra-word gaps (gaps between PAWs within a word) and inter-word gaps (gaps between words). Intra-word gaps in the Arabic language are different from the ones in Latin-based languages. In Latin, intra-word gaps refer to the spaces that arise arbitrarily between any successive letters as a result of handwriting styles. In Arabic, intra-word gaps are the ones between two PAWs, where the word must be disconnected due to NLC letters. This is part of the structure of the language.

Generally, handwritten texts lack the uniform spacing that is normally found in machine-printed texts. In Arabic machine-printed texts, the inter-word gaps are much larger than intra-word gaps. However, in Arabic handwritten documents, the spacing between the two types is mostly the same [3]. This is pointed out in Figure 2 from the CENPARMI cheque database [11]. Since the shape of most of the NLC letters are curved, with the open end to the left, they are usually written with long strokes, which shrink the distance between words. Sometimes, they caused overlapping, or touching between words.

Fig. 2. Intra and inter word gaps in Arabic text

1.3 Related Work

Word segmentation is a critical step towards word spotting and text recognition. There are many word segmentation techniques in the literature [14]. Nevertheless, it is still a challenging problem in handwritten documents. Word segmentation techniques are based mainly on the analysis of the distance between adjacent CCs. The algorithms can be categorized into gap thresholding and metric classification. In the former, the threshold is determined to distinguish between gap types. In the latter, the gaps are classified into either inter or intra word gaps.

There is little research for Arabic handwritten text segmentation. Some works apply to manual segmentation [13]. In [9], an online Arabic segmentation method was proposed. The gap types are classified based on local and global online features. The fusion of multi-classification decisions was used as a post-processing stage to verify the decisions.

J. Alkhateeb et al. proposed a method for Arabic handwritten text segmentation into words based on the distances between PAWs and words [6]. Vertical projection analysis was employed to calculate the distances. The statistical distribution was used to find the optimal threshold. Bayesian criteria of minimum classification error were used to determine the threshold. The technique was applied on a subset of the IFN/ENIT database. The correct segmentation of one-word and three-word images was 80.34% and 66.67% respectively.

In [8], an offline handwritten Arabic text segmentation technique was introduced. First, the CCs of the images were detected based on the baseline. Their bounding boxes were determined. These boxes were extended to include the dots and any small CCs. The distances between adjacent PAWs were obtained. They assumed that the distance between words is larger than the distance between PAWs. Based on that assumption, a threshold approach was used. Two conditional probabilities were determined by manually analyzing more than 200 images. A Bayesian histogram minimum classification error criteria was used to find the optimal distance. They achieved 85% of correct segmentation.

M. Kchaou et al applied scaling space to segment Arabic handwritten documents into words [7]. The techniques that were used for segmentation were scaling space and feature extraction from horizontal and vertical profiles. Two documents written by five writers were used in their experiments. Segmentation errors varied between 29.5% and 3.5%. They believe that the errors arising from different writer styles, coordinating conjunctions and distances between PAWs.

In [18], the segmentation is based on extracting several features from the adjacent clusters. The main and secondary components are merged into clusters. Nine features were extracted. The neural network is used to classify the gaps between the words. Overall performance is about 60% correct segmentation.

Due to the importance of text segmentation, four Handwriting Segmentation Contests were organized: ICDAR 2007, ICDAR 2009, ICFHR 2010, and ICDAR 2013 [15]. Therefore, a benchmarking dataset with an evaluation methodology were created to capture the efficiency of the methods. The total number of participants on these competitions was thirty research groups with different algorithms. In addition, there are plenty of methods for Latin-based languages in comparison to Arabic language that address this problem [14].

1.4 Proposed Approach

The main difference with our segmentation approach from previous methods is utilizing the knowledge of Arabic writing by shape analysis. In [6], [9], and [8], the authors pointed out the importance of using the language specific knowledge for Arabic text segmentation. In addition, in [7], the authors claim that one of the problems of Arabic text segmentation is the inconsistent spacing between words and PAWs. Our approach for segmentation is a two-stage strategy: (1) metric-based segmentation, and (2) recognition-based segmentation.

Utilizing the Knowledge of Arabic Writing. In the Arabic alphabet, twenty-two letters out of twenty-eight have different shapes when they are written at the end of a word as opposed to the beginning or middle. Two non-basic characters have different shapes at the end of a word. Therefore, analyzing these shapes can help to identify the end of a word. In fact, there are just fourteen main shapes that can be used to distinguish the end of a word, since the remaining characters have the same main part but have a different number and/or position of dots. Only NLC letter shapes, which cause the disconnection within a word, are written the same way at the beginning, the middle or the end of a word. Therefore NLC letters cannot identify the end of a word. Consequently, End Shape Letters (ESLs) can be categorized into two classes: endWord and nonEndWord. Figure 3 shows the shape of the letter Noon when it is written at the beginning of the word, the middle and the end, and this letter is part of endWord class.

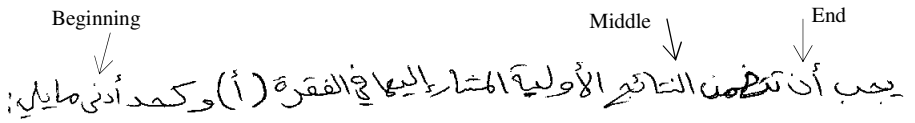

Fig. 3. Letter Noon in different positions

1.5 Our Methodology

Our methodology is composed of two stages as mentioned earlier. The first stage is called metric-based segmentation. The second stage is named ESL-based segmentation. The input of our system is a binarized text line. A method that was proposed by M. Al-Khayat et al. [16] for text line segmentation was used. First, the Connected Components (CCs) of a text line were extracted. CCs by definition, consist of connected black pixels. Normally, a PAW is composed of several CCs: a main component, diacritics, and/or directional markings. Therefore, the first main step in segmenting an Arabic handwritten word is detecting and labeling its CCs. CC analysis is the most efficient approach since the Arabic script consists of several overlapping CCs. The 8-connectivity method was used. Second, the secondary components were removed, which was explained in detail in Section 2. Then, metric-based and ESL-based segmentation were applied. The ESL-based proposed method was provided in Section 4. The overall methodology is given as a block diagram in Figure 4.

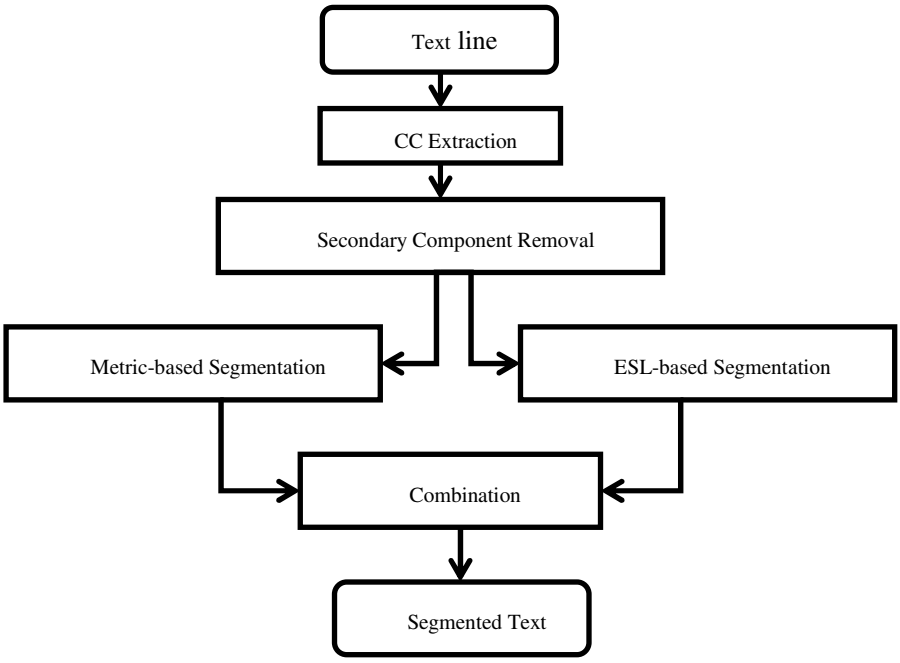

Fig. 4. Overall Methodology

2 Removal of Secondary Components

In this paper, the secondary components are removed to improve the performance of metric-based segmentation and to reduce the number of classes of Final Shape Letter recognition system. However, many algorithms also remove the secondary components to facilitate skew correction and baseline estimation. Some methods also detect the secondary components to extract more features for recognition or spotting systems. We used secondary component removal using morphological reconstruction [12].

Mathematical morphology is an essential tool in image processing that is used to process images based on its shape information. Reconstruction is a morphological transformation that involves two images. The mask image constrains the transformation. The marker image is the starting point for the transformation. Using the morphological reconstruction method that is based on a thin horizontal line facilitates main component extraction. This line is defined below the middle of the image. The reconstruction method is used to ascertain that only the main components are analyzed. We process only binary images. The word images are the masks. The marker is a generated binary image with the same size as the mask image and a horizontal line that is located below the middle of the image.

3 Metric-Based Segmentation

In this stage, the distance between adjacent components was computed using a gap metric. This method is somewhat like a writer dependent technique since the threshold was estimated based on a given text line. In fact, since spaces between words are part of a writing style, this writer dependent technique provides better result [14]. Thus, a global threshold across all documents is an inadequate solution.

3.1 Distance Computation

After extracting the main components that are ordered from left-to-right, a bounding box for each component was calculated. Then all the overlapped bounding boxes were merged. The minimum horizontal distances between pairs of adjacent bounding boxes were measured. After that, all gap metrics of the text line were sorted.

3.2 Threshold Estimation

After identifying the largest space (determined based on empirical study) between the sorted values, the threshold was determined. The threshold is the minimum value of the largest group of gap metrics. If the spaces between the gap metrics are almost the same, the threshold is calculated to be the mean of the gap metrics. Finally, the text line is segmented into words based on this threshold. The algorithm is given below:

Algorithm for Word Threshold Estimation

For each text line
Calculate the bounding boxes for each CC (Bc_i), i =1….. L
Calculate the distance between Bc_i and Bc_{i+1}
Find all gap metrics G_j
Find spaces between G_j that is denoted by S_i
If a large space is found
    The minimum value in the largest group is determined as a Threshold (T).
    $T \subset G_i$
Else
    The mean of the gap metrics is the threshold
    $T = mean(G_i)$

4 ESL-Based Segmentation

In this stage, the main idea is to recognize the ESL that helps to specify the word segment. ESL can be isolated or part of a PAW. However, the end-shape needs to be detected first before recognition can begin. Each step is described in the following sub-sections. Our method is depicted in Figure 5.

4.1 ESL Detection

At this stage, the main purpose is to detect the isolated letter or the last letter of a PAW. The last part will be extracted based on the height, width, and the baseline position.

4.2 ESL Recognition

At this stage, either the end-shape of the main component or the isolated letter is sent to an ESL recognizer. We created an ESL database and classifier to identify the end of a word. This recognizer classifies the end-shapes of main components and of an isolated letter.

The ESL database contains the shape of letters at the final position (only in its isolated form). The endWord set contains eleven classes and the nonEndWord set is composed of three classes. We used the CENPARMI Arabic isolated letter database [19]. To test the ESL recognition system before applying it to the documents, a testing model was generated using the testing set of the CENPARMI Arabic isolated letter database. We applied the method that was used by M. W. Sagheer [24]. This ESL recognizer consists of the following three phases: (1) Pre-processing, (2) Feature extraction, and (3) Recognition.

Since our concern is the main component of the letters, we removed all the secondary components that comprise less than half the area of the largest component. Then, the bounding box of the main component is calculated in order to eliminate all the white spaces around it. The image was normalized to two different sizes, 64 x 64 and 128 x 128 pixels by using aspect ratio adaptive normalization strategy [22]. Two different sizes of the image were used for different feature extraction processes. In addition, the image was skeletonized to standardize the representation of the images and facilitate feature extraction. The Zhang and Suen thinning algorithm [23] was applied. We extracted gradient features and structural features. Several experiments were conducted with different features to find the best combination of these features that produce the best results as shown in Table 1.

Gradient Features Extraction. In our gradient feature extraction phase, each image of size 128 x 128 pixels was converted into a grayscale image. Robert's filter masks were applied on the images.

Let IM(x, y) be an input image; the horizontal gradient component (g_x) and vertical gradient component (g_y) were calculated as follows:

g_x = IM(x+1, y+1) - IM(x, y)

g_y= IM(x+1, y) - IM(x, y+1)

• The gradient strength and direction of each pixel IM(x,y) were calculated as follows:

$$\text{Strength: } s(x, y) = \sqrt{gx^2 + gy^2} \quad (1)$$

$$\text{Direction: } \theta(x, y) = \tan^{-1}(g_y / g_x) \quad (2)$$

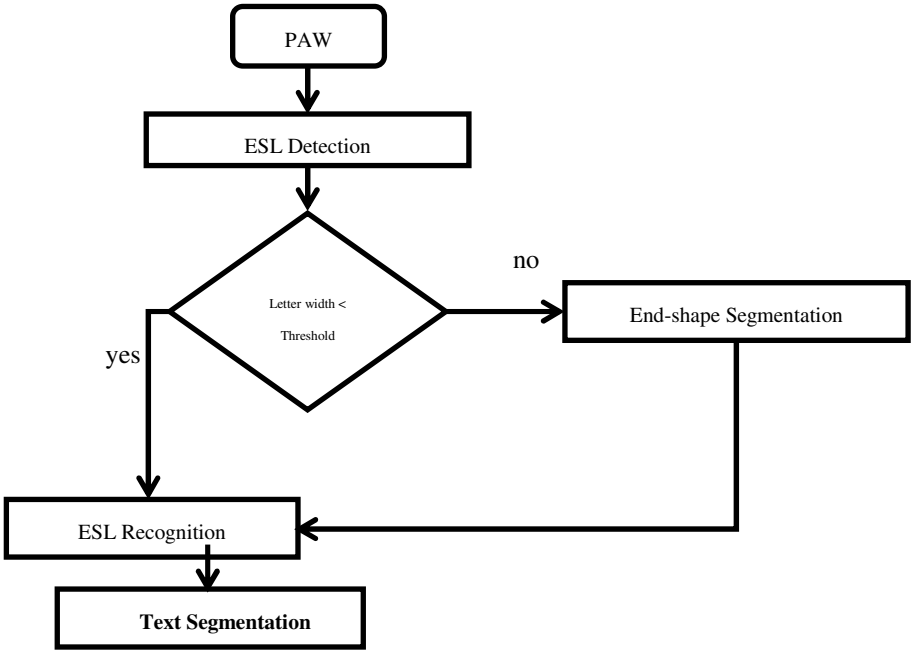

Fig. 5. ESL-based segmentation method

After calculating the gradient strength and direction for each pixel, the following steps were taken in order to calculate the feature vector:

1. The direction of a vector (g_x, g_y) in the range of [π,-π]. These gradient directions were quantized to 32 intervals of π/16 each.
2. The gradient image was divided into 81 blocks, with 9 vertical blocks and 9 horizontal blocks. For each block, the gradient strength was accumulated in 32 directions. By applying this step, the total size of the feature set in the feature vector is (9 x 9 x 32) = 2592.
3. To reduce the size of a feature vector, a 5 x 5 Gaussian filter was applied by down sampling the number of blocks from 9 x 9 to 5 x 5. The number of directions was reduced from 32 to 16 by down sampling the weight vector [1 4 6 4 1]. The size of the feature vector is 400 (5 horizontal blocks x 5 vertical blocks x 16 directions).
4. A variable transformation ($y = x^{0.4}$) was applied on all features to make the distribution of features Gaussian-like.

Structural Features Extraction. In addition to the gradient feature, other structural features were extracted. The additional features are: the number of black pixels, horizontal and vertical histograms, end and intersection points, holes, and structure of the top part of the image. However, the horizontal and vertical features were removed

since they provide lower performance. Moreover, the upper profile features were used to capture the outline shape of the top part [17]. To extract the upper profile feature the following steps were followed:

- Each image was converted into a two-dimensional array.
- For each column, the distance was measured from the top of the image to the closest black pixel.

Feature Vector. After extracting the gradient and structural features from each image, all the features were merged to make a feature vector of size of 468 (400 gradient features, 64 upper profile, 4 structural features). Then, this feature vector was passed to the classification phase.

Recognition. A Support Vector Machine (SVM) is a technique in the field of statistical learning. SVMs have shown to provide good results for both offline and online cursive handwriting recognition [21]. We used an open source library for the implementation of SVM called LibSVM [20]. The input of LibSVM is a feature matrix and the output is the classification result probabilities. LibSVM uses a Radial Basis Function (RBF) kernel for mapping a nonlinear sample into a higher sample space. RBF is given by:

$$K(X_i, X_j) = \exp(-\gamma \| X_i - X_j \|2) , \gamma > 0 \quad (3)$$

For the K-class problem, K2 SVMs are trained by a pairwise approach. The probability is estimated for test sample x that belongs to class i. The probabilities are obtained from a one-against-one class probability. The two optimal parameters γ and C were chosen by using v-fold cross validation. A training model was generated for the whole collection of images with their class labels.

Table 1. Experimental Results with Different Features

Feature vector	**Gradient feature**	Horizontal projection	Vertical projection	**Upper profile**	**Structural features**	Feature vector	Recognition result
Fv1	x					416	89.97%
Fv2	x	x				480	89.66%
Fv3	x		x			480	89.51%
Fv4	x				x	420	90.27%
Fv5	x			x		464	90.73%
Fv6	x			x	x	468	**90.88%**

5 Experiments

We performed experiments using the IFN/EINT [10] database. It was developed by the National School of Engineers of Tunis (ENIT), and the Institute of Communication Technology (IFN) in Germany. The database contains 937 Tunisian town/village names written by 411 writers. In fact, 448 names contain two to three words. We did our experiments on the names that contain two to three words, since our gap threshold estimation is based on the distance within the images. In addition, since the CENPARMI database is composed of only the isolated shapes of the letters (which is a subset of the final shape letters) and there does not exist an available database that has all the shapes of the letters in their final positions (connected), therefore, we applied our experiments on those city names that can be segmented using the CENPARMI isolated letters. This set contains a total of 440 names. We used set-a for training and subsets from set-b, set-c, set-d, and set-e for testing. Table 2 shows the word segmentation results of metric-based, ESL-based and the result of the combined methods. In Section 1.3, the results of related work is reported.

Table 2. Word segmentation result

Set	Metric-based	ESL-based	ESL + Metric
Set-b	67.34%	82.00%	86.16%
Set-c	68.29%	84.15%	93.48%
Set-d	83.12%	98.61%	99.02%
Set-e	71.05%	86.84%	93.88%

6 Conclusion

Arabic handwriting recognition and spotting depend on accurate segmentation. In this paper, we introduced a new segmentation approach for Arabic handwritten text. Our experiments show promising results. Most of the errors are caused by the confusion between classes. An example of such an error is illustrated in Figure 6. Our future work will include creating a database of final shape letters that are connected to the PAWs, improving the segmentation method, and extract more features to improve the FSL recognition system.

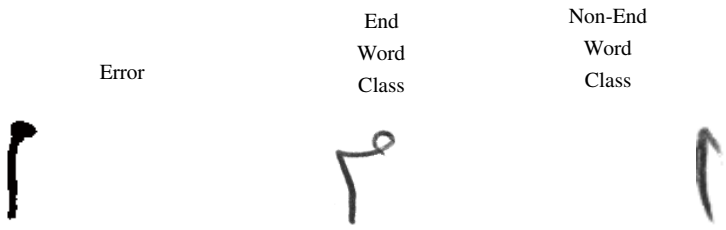

Fig. 6. Error Analysis

Acknowledgment. This work was supported by King Abdulaziz University (KAU), Jeddah, Saudi Arabia and Ministry of Higher Education in Saudi Arabia.

References

1. Huang, C., Srihari, S.: Word Segmentation of Off-line Handwritten Documents. In: Proceedings of the Document Recognition and Retrieval (DRR) XV, IST/SPIE Annual Symposium, San Jose, CA, USA, vol. 6815 (2008)
2. Olivier, G., Miled, H., Romeo, K., Lecourtier, Y.: Segmentation and Coding of Arabic Handwritten Words. In: Proceedings of 13th International Conference of Pattern Recognition (ICPR 1996), vol. 3, pp. 264–268 (1996)
3. Amin, A.: Recognition of Printed Arabic Text based on Global Features and Decision Tree Learning Techniques. Pattern Recognition 33(8), 1309–1323 (2000)
4. Miled, H., Amara, N.E.B.: Planar Markov Modeling for Arabic Writing Recognition: Advancement State. In: Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR 2001), Seattle, USA, pp. 69–73 (2001)
5. Westall, J.M., Narasimha, M.S.: Vertex Directed Segmentation of Handwritten Numerals. Pattern Recognition 26(10), 1,473–1,186 (1993)
6. AlKhateeb, J.H., Ren, J., Ipson, S.S., Jiang, J.: Knowledge-based Baseline Detection and Optimal Thresholding for Words Segmentation in Efficient Pre-processing of handwritten Arabic text. In: Proceedings of 5th International Conference on Information Technology: New Generations (ITNG 2008), Las Vegas, Nevada, pp. 1158–1159 (2008)
7. Kchaou, M., Kanoun, S., Ogier, J.: Segmentation and Word Spotting Methods for Printed and Handwritten Arabic Texts: A Comparative Study. In: Proceedings of 13th International Conference on Frontiers in Handwriting Recognition (ICFHR 2012), Bari, Italy, pp. 274–279 (2012)
8. AlKhateeb, J.H., Jiang, J., Ren, J., Ipson, S.: Component-based Segmentation of Words from Handwritten Arabic text. In: Proceedings of World Academy of Science, Engineering and Technology (WASET), Vienna, Austria, vol. 31 (2008) ISSN: 1307- 6884
9. Elanwar, R.I., Rashwan, M., Mashali, S.: Arabic Online Word Extraction from Handwritten Text Using SVM-RBF Classifiers Decision Fusion. In: Proceedings of the 4th WSEAS International Conference on Nanotechnology, Cambridge, UK, pp. 68–73 (2012)
10. Pechwitz, M., Maddouri, S.S., M"argner, V., Ellouze, N., Amiri, H.: IFN/ENIT- Database of Handwritten Arabic Words. In: 7th Colloque International Francophone sur l'Ecrit et le Document (CIFED), Hammamet, Tunis, vol. 2, pp. 127–136 (2002)
11. Al-Ohali, Y., Cheriet, M., Suen, C.Y.: Databases for Recognition of Handwritten Arabic Cheques. Pattern Recognition 36(1), 111–121 (2003)
12. Jamal, A.T., Suen, C.Y.: Removal of Secondary Components of Arabic Handwritten Words Using Morphological Reconstruction. In: Proceedings of the 2nd International Conference on Information Technology (ICIT), Dubai, UAE (2014)
13. Manmatha, R., Rath, T.M.: Indexing Handwritten Historical Documents - Recent Progress. In: Proceedings of the Symposium on Document Image Understanding Technology, Greenbelt, USA, pp. 77–85 (2003)
14. Louloudis, G., Gatos, B., Pratikakis, I., Halatsis, C.: Text Line and Word Segmentation of Handwritten Documents. Pattern Recognition 42(12), 3169–3183 (2009)
15. Nikolaos, S., Gatos, B., Louloudis, G., Pal, U., Alaei, A.: ICDAR 2013 Handwriting Segmentation Contest. In: Proceedings of 12th International Conference on Document Analysis and Recognition (ICDAR), Washington DC, USA, pp. 1402–1406 (2013)

16. Al-Khayat, M., Lam, L., Suen, C.Y., Yin, F., Liu, C.-L.: Arabic Handwritten Text Line Extraction by Applying an Adaptive Mask to Morphological Dilation. In: Proceedings of the 10th IAPR International Workshop on Document Analysis Systems (DAS), Gold Coast, Australia, pp. 100–104 (2012)
17. Aghbari, Z., Brook, S.: HAH manuscripts: A Holistic Paradigm for Classifying and Retrieving Historical Arabic Handwritten Documents: Expert Systems with Applications. An International Journal 36(8), 10942–10951 (2009)
18. Srihari, S., Srinivasan, H., Babu, P., Bhole, C.: Handwritten Arabic Word Spotting Using the CEDARABIC Document Analysis System. In: Proceedings of Symposium on Document Image Understanding Technology (SDIUT), pp. 123–132. College Park, USA (2005)
19. Alamri, H., Sadri, J., Suen, C.Y., Nobile, N.: A Novel Comprehensive Database for Arabic Off-line Handwriting Recognition. In: Proceedings of 11th International Conference on Frontiers in Handwriting Recognition (ICFHR 2008), Montreal, Canada, pp. 664–669 (2008)
20. Chang, C., Lin, C.: LIBSVM: A Library for Support Vector Machines (2001), http://www.csie.ntu.edu.tw/cjlin/libsvm
21. Gatos, B., Pratikakis, I., Kesidis, A.L., Perantonis, S.J.: Efficient Off-Line Cursive Handwriting Word Recognition. In: The Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006), La Baule, France, pp. 121–125 (2006)
22. Liu, C.-L., Nakashima, K., Sako, H., Fujisawa, H.: Handwritten Digit Recognition: Investigation of Normalization and Feature Extraction Techniques. Pattern Recognition 37(2), 265–279 (2004)
23. Zhang, T.Y., Suen, C.Y.: A Fast Parallel Algorithm for Thinning Digital Patterns. Communications of the ACM 27(3), 236–239 (1984)
24. Sagheer, M.W.: Novel Word Recognition and Word Spotting Systems for Offline Urdu Handwriting. Master Thesis, Concordia University, Montreal, Canada (2010)