

A Recognition-Based Approach to Segmenting Arabic Handwritten Text

Ashraf Elnagar, Rahima Bentrchia

Department of Computer Science, College of Sciences, University of Sharjah, Sharjah, UAE
Email: ashraf@sharjah.ac.ae

Received 22 July 2015; accepted 1 November 2015; published 4 November 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Segmenting Arabic handwritings had been one of the subjects of research in the field of Arabic character recognition for more than 25 years. The majority of reported segmentation techniques share a critical shortcoming, which is over-segmentation. The aim of segmentation is to produce the letters (segments) of a handwritten word. When a resulting letter (segment) is made of more than one piece (stroke) instead of one, this is called over-segmentation. Our objective is to overcome this problem by using an Artificial Neural Networks (ANN) to verify the resulting segment. We propose a set of heuristic-based rules to assemble strokes in order to report the precise segmented letters. Preprocessing phases that include normalization and feature extraction are required as a prerequisite step for the ANN system for recognition and verification. In our previous work [1], we did achieve a segmentation success rate of 86% but without recognition. In this work, our experimental results confirmed a segmentation success rate of no less than 95%.

Keywords

Character Segmentation, Handwritten Recognition Systems, Arabic Handwriting, Neural Networks, Multi-Agents

1. Introduction

Automatic recognition of handwritings is developing well as a result of research contributions in this field. There are two main types of character recognition systems, namely, on-line and off-line systems. On-line systems recognize handwritings input on a tablet or any a similar device by a digital pen or stylus. Off-line systems deal with images of handwritings stored. The temporal information in the first type would positively contribute to the recognition process. Such information is absent in the second type, which makes this problem in particular more challenging.

In this paper, we study the problem of Arabic handwritten characters recognition, [2]. Several factors contribute to the difficulty of recognizing Arabic handwritings, which is, mainly, attributed to the quality of the writing (the poorer the writing, the harder it is to be recognized). This subject is still under research because of its potential array of applications. So many services need to automate human processes for time and accuracy purposes. One application appears in automating the process of reading and recognizing handwritten accounts names and checks amounts in banks. Sorting incoming mails by recognizing the handwritten addresses is another main application required in post-mail offices. One more application can be seen in retrieving the ancient Arabic handwritten manuscripts.

Our recognition system is embedded in the segmentation system that was proposed in [3] as illustrated in Figure 1. After word segmentation, the resulting segments are normalized and a set of twenty features is extracted and fed into an artificial neural network. The recognition outputs range from class 1 to class 46, as clarified in Table 1, where each class represents the location of a single letter in the word or the location of a group of letters that share similar shape characteristics. Some segments may not be recognized because of the poor writing, so they are combined with the following segment and a new set of twenty features is extracted again and passed to the neural network. This process is repeated until the segment is recognized. A set of grouping rules is also applied to specific recognized characters.

The rest of the paper is organized as follows. Section 2 presents related work. Segmentation phase is described in Section 3 while the recognition phase is discussed in Section 4. Section 5 presents the experimental results, followed by the Conclusion and future direction.

2. Related Work

Most of the work on handwriting recognition was done on Latin text. This lack in Arabic handwriting recognition systems is highly related to the difficulty of segmenting words into characters because of the cursive nature of Arabic handwriting. Therefore, Arabic recognition methods can be divided into those which first segment the word to be recognized, and those that recognize the whole word. In this work we are focusing on the segmentation of Arabic text rather than digits' strings such as in [3] [4].

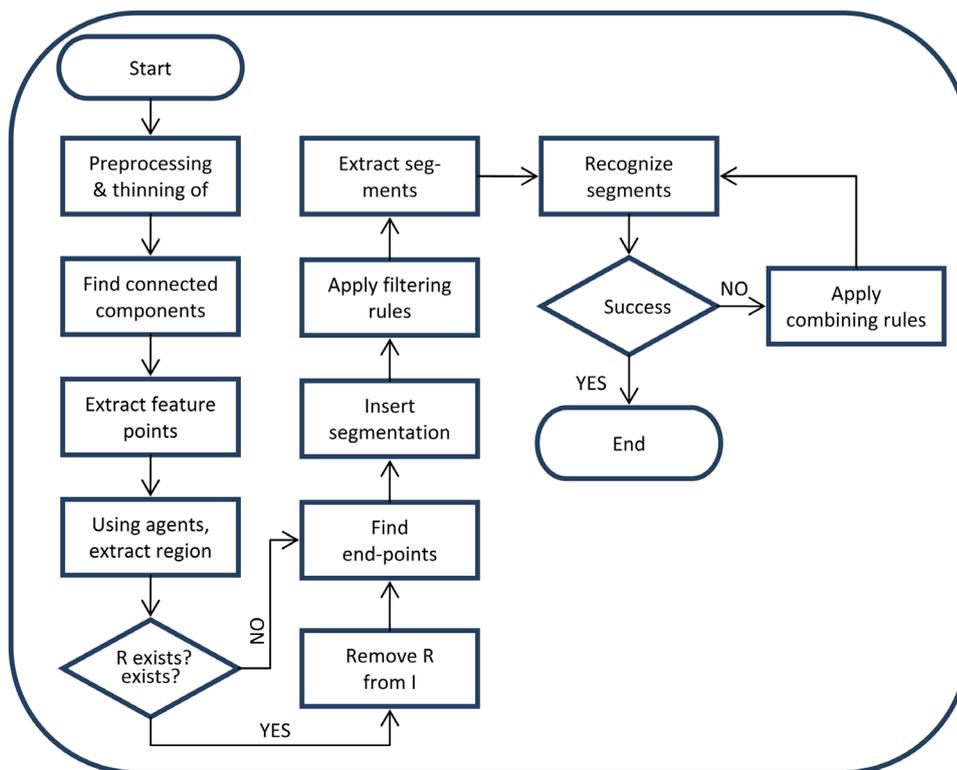


Figure 1. Flow-chart of the major steps of the main system.

Table 1. The output classes of the proposed recognition system.

Letter	Class #	Letter	Class #	Letter	Class #
ف	31	س	17	ا	1
ق	32	ش	18	أ	2
ك	33	ص	19	ب	3
ح	34	ض	20	ت	4
ط	35	ذ	21	ث	5
ظ	36	ر	22	ج	6
ل	37	ز	23	ح	7
ه	38	س	24	خ	8
م	39	ع	25	د	9
ه	40	غ	26	ج	10
م	41	ف	27	ل	11
م	42	ع	28	ر	12
ن	43	ق	29	ي	13
و	44	ه	30	ي	14
	45	ش			15
	46	س			16

Earlier surveys discussed both Arabic printed and handwritten texts, [1] [4]-[6]. One of the segmentation based methods for automatic recognition of printed Farsi, Arabic, and Urdu texts was proposed by Parhami and Taraghi in 1981, [7]. In this approach, sub-words were segmented and recognized according to geometrical features such as concavities, loops, and connectivity. No performance results were reported for this algorithm but smaller type fonts may not be recognized perfectly.

In 1986, Amin and Masini proposed a system for segmentation and recognition that used horizontal and vertical projections and shape-based primitives [8]. On 100 multi-font words, it achieved a character recognition rate of 85% and a word recognition rate of 95%.

Gillies *et al.* constructed a recognition system for Arabic text, [9], where words were over-segmented using two different splitting methods, then the resulting segments are ordered and combined in groups and sent to a trained neural network which recognized whole characters from the grouping options. These were passed to a Viterbi search to predict the word. Using a testing set of 138 page images, digitized to 200×200 , the system achieved a recognition rate of 93%. However, this rate degraded to 89% when the same set was used with a size of 100×200 .

In 2002, Hamami and Berkani developed a structural approach to handle many fonts and it included rules to prevent over-segmentation [10].

Hidden Markov Model (HMM) is used also to recognize words by using words features. In 2001, Dehghanet. Al split words into overlapping vertical segments [11], then, they extracted column features and passed them to HMM.

Al-Qahtani and Khorsheed presented a system based on Hidden Markov Model Toolkit in 2004, [12] [13]. One system did not require the segmentation stage and recognized Arabic scripts using HTK. The second system decomposed the text into line images and divided each line image into smaller overlapped frames. Then it extracted statistical features from each frame and passed them to HTK.

Two segmentation free recognition methods appeared in 1995 by Al-Badr and Haralick. In the first system, [14], the whole word was recognized by detecting a set of shape primitives which matched to a constrained set of symbol models. The recognition rate was 99.7% for synthetically degraded symbols and 94.1% for scanned

symbols. For isolated words, the system achieved 99.4% for noise-free words, 95.6% for synthetically degraded words, and 73% for scanned words. The second system was developed to recognize machine printed Arabic words without prior segmentation. The idea was based on shape primitives that were detected with mathematical morphology operations, [15]. The recognition rate was 99.4% for noise-free texts and 73% for scanned texts.

Khorsheed and Clocksin proposed in 1999 another holistic system where features were extracted from a word's skeleton for recognition without prior segmentation [16].

In 2000, Amin introduced another holistic approach where global features such as loops and peaks were extracted from the input word [17], and passed to the C4.5 machine learning system to generate a decision tree for classifying the word. The success rate of the system was 92% using 1000 Arabic words with different fonts.

Another method was presented by Pechwitz and Maergner [18], where the recognition system was based on a semi-continuous 1-dimensional HMM. From each input word, features were collected using sliding window approach. The recognition results achieved 89%, using the IFN/ENIT database of Arabic handwritten words for testing.

In this work, a recognition-based segmentation method for Arabic handwriting is developed. The method used a multi-agent approach to segment words, [1], and relied on recognition to verify the validity of the candidate segmentation points. Comparing the previous methods of segmentation approaches and our approach, this segmentation method is not only resolved the shortcomings of the previous related methods but also achieved better results by avoiding under segmentation. This depended on the high performance of the agents and the right decision to select artificial neural network with grouping rules which improved detecting the candidate segmentation points.

3. Segmentation Phase

Our segmentation system, which we proposed in [3], was basically based on a multi-agent approach to identify the segmentation points.

Initially, the image of Arabic handwritten text was binarized and cleaned from noise. Then, the text was segmented into lines and each line was segmented into words. The resulting words were thinned and the main connected components in each word were determined and passed to agents that extracted three types of feature points before starting their work.

The identification of initial cutting points strongly depends on seven agents. Six agents are major, which are: loop agent, letter Seen agent, under-baseline-cavities agent, above-baseline-right-cavity agent, above-baseline-left-cavity agent, and above-baseline-narrow-left-cavity agent. The other agent which is the baseline agent is a minor one since it was used by major agents to facilitate their task. First, the agents detected regions that look like some of Arabic characters, these regions were subtracted from the whole word and the remaining parts were left for further processing. Next, all end points features were extracted from the remaining regions and an initial cutting point was inserted between every two successive end points'. Finally, a set of filtering rules was applied to remove the extra segmentation points. The experiments reported very good results where the success rate was 86%.

4. Recognition Phase

This phase is very important in our segmentation system, [3]. Since segmenting words into characters is a challenging task, especially for Arabic handwritings, a verification tool is needed to measure the segmentation performance. Artificial neural network was selected to decide if the resulting segment is a letter or a stroke and then needs further processing.

Generally, artificial neural networks are very common in pattern recognition field. Our decision to use ANN as a recognition model was based on the excellent features that it possesses compared to other recognition tools. A well-trained neural network can perform complex functions and solve challenging problems that are difficult for conventional computers or human beings since it is based on learning what it sees. In addition, neural networks can be modified easily and retrained when the requirements of the problem are changed. Finally, its integration property allows several recognition tools to work properly and cooperate with neural networks. This feature may increase the efficiency of the problem solution. The following sections describe the main steps in our approach.

4.1. Features Extraction

In this step, each segment image is converted into numerical features which describe the segment. The feature extraction methods used in character segmentation systems are probably the most important factor in achieving a good segmentation/recognition rate. After segmenting the word, its output segments are normalized into 250×250 images. Then, twenty structural features are extracted from them. Fifteen Fourier descriptors are extracted from the segments contour and normalized to remove character variations in shift, size, and rotation, [19] [20]. The other five features include number of loop, number of black points to total number of points' ratio, the existence of connection to the right and left of the segment [21], and height to width ratio.

A different number of Fourier descriptors are tested and the final set includes 15 descriptors. The selection of these features was based on their ability of describing the general shape of any closed curve such as characters by a set of Fourier coefficients. Suppose that a character consists of a sequence of points (x_i, y_i) , where $i = 1, 2, \dots, N$ and N is the number of points in character's boundary. Each of these points can be represented as a complex number $a(n) = x(n) + iy(n)$. The discrete Fourier transformation $u(n)$ represents the coordinate sequence $a(n)$ as follows:

$$u(n) = \sum_{k=0}^{N-1} a(k) e^{\frac{i2\pi kn}{N}}$$

The first 15 coefficients (descriptors) are selected as our features. This is referred to that the general properties of the character shape are kept in the first (low) coefficients. Because characters varied in size, location and maybe rotation angle, Fourier descriptors can be manipulated to be character rotation, scale, and shift invariant. To make Fourier descriptors rotation and shift invariant, only their absolute values are used, and to make them scale invariant, the coefficient are normalized by dividing them by the first coefficient $a(1)$, [19].

4.2. Reconstruction and Recognition

Artificial neural networks are computational models which take their inspiration from the models and theories of the human brain. The most popular neural network is the multilayer feed-forward network where neurons are grouped as layers and connections between neurons in consecutive layers are permitted. The inputs are fed from the input layer and outputs are at the output layer.

In this work, after images normalization, a vector of 20 features is extracted from each segment image and classified using a feed forward neural network trained by back-propagation learning algorithm [2]. The structure of this ANN, shown in **Figure 2**, consists of four layers: one input layer of 20 neurons, two hidden layers of 100 neurons and one output layer of 46 neurons. The neurons in the hidden layers and the output layer are working

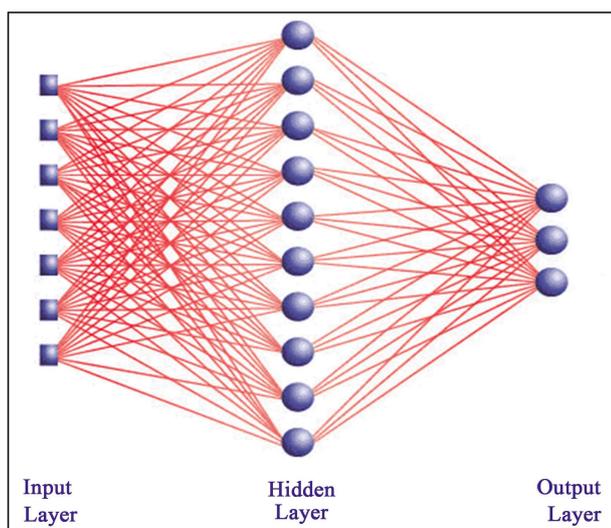


Figure 2. General structure of the ANN in which the hidden layer consists of 2 layers.

using tan-sigmoid and linear algorithms, respectively, and the network is trained using *traincgf* function. The final selection of the ANN's structure and the used algorithms was determined after trying so many other structures and testing several algorithms. The ANNs which are trained using "*traincgf*" give better results compared to those that use other training algorithms. Moreover, *traincgf* has smaller storage requirements and faster convergence in some recognition problems.

The 46 outputs represent the classes that each segment may belong to, and each class includes letters that have similar shape (body) in a specific location in the word; in the beginning of the word, in the middle, in the end, or isolated. The list of output classes appears in **Table 1**.

The proposed ANN is trained using 2000 characters; more than 40 characters from each class, written by different people. Then, testing was accomplished by selecting examples from each class and passing them to the ANN. A total of 250 characters were used as testing examples. The obtained recognition rate exceeds 87%.

4.3. Restoration and Grouping

This step is required when the word is over-segmented and additional segmentation points were determined. As a result, pseudo characters that passed to the neural network are not correctly recognized. To remedy this situation, the extra segmentation points are removed and the adjacent segments are combined and passed again to the neural network, [22]. This process is repeated until the candidate character is recognized.

A preprocessing step was applied first to remove segmentation points that yield to segments with width less than a threshold. This process eliminates most of the strokes which wrongly found in letters such as "ainc" and "haa". The following examples depict this case in **Figure 3**. As clarified in the figure, the small segment of letter "ع" in the word "الشرايع", and of the letter "و" in the word "صحراوي", and of the letter "ء" in the word "الرجاء" are eliminated and combined to their related segments to form complete letters.

Because word segmentation is a precise process, a set of rules is used to cooperate with the embedded recognition system in order to keep the correct segments (characters) and combine the wrong ones in a correct way.

The grouping rules are based on the recognition results of segments. As observed in the segmentation stage, a letter is segmented in the worst case into three segments and this happened in letters that belong to the classes: 15, 16, 17, 18, and in some types of the handwritten letters of classes 7, 10, 19, 20, 21, and 22. In addition, characters of classes 2, 3, 6, 11, 12, 13, 23, 24, 29, 32, 43, and 44, shown in **Table 1**, are segmented into two segments in the most types of handwritings. **Figure 4** clarifies these cases of triple (three segments) and double (two segments) segmentation.

The main objective of the cooperation between the recognition model and the grouping rules is to handle the over segmented letters introduced in **Figure 4**. The following diagram, shown in **Figure 5**, demonstrates this cooperation strategy.

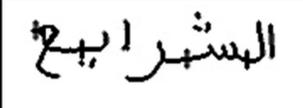
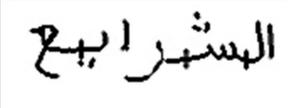
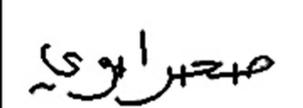
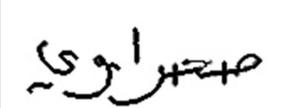
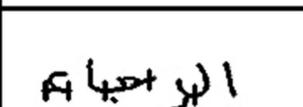
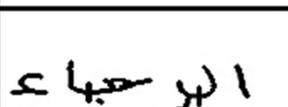
Before Elimination	After Elimination
	
	
	

Figure 3. The word before and after extra segmentation points.

Class no.	Double Segmentation	Triple Segmentation
3		
6		
13		
24		
29		
10		
18		
21		
22		

Figure 4. The double and triple segmentation of some characters.

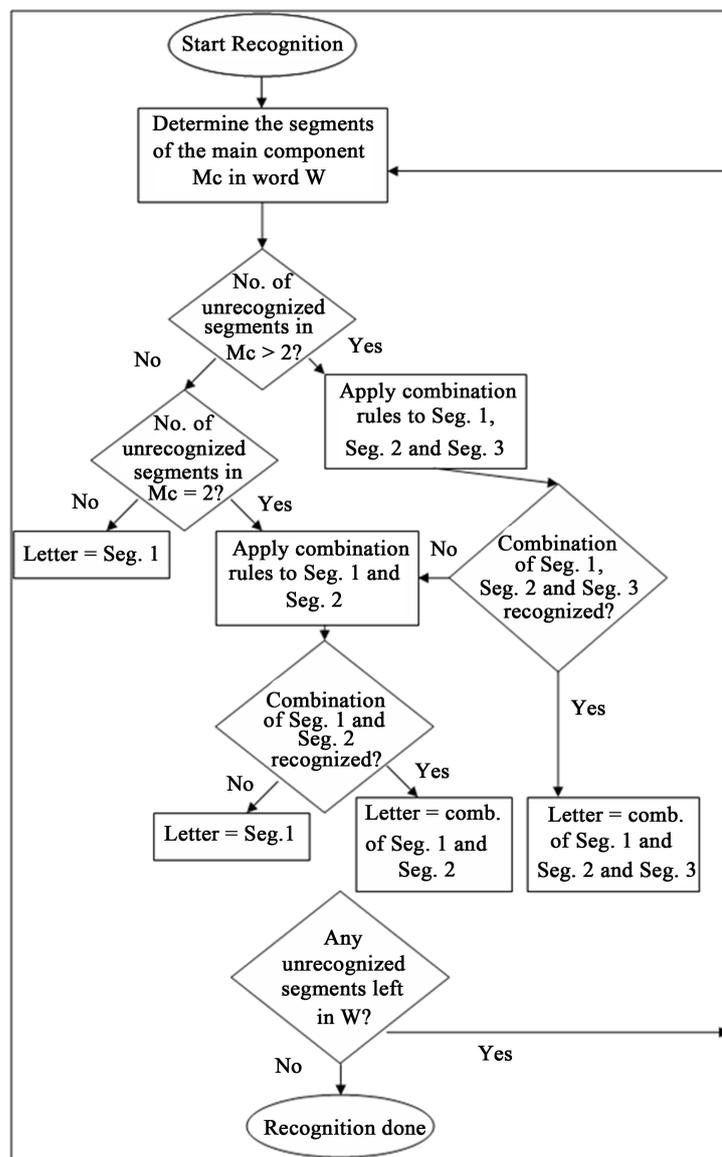


Figure 5. The grouping rules used with the recognition model.

First, the number of resulting segments was determined. Grouping rules were applicable only when the number of segments is equal to or greater than two. Initially, the first two segments or three segments, if any, are passed to the neural network and recognized separately. If the first segment was recognized as a letter which can be a part of any other letter that may over-segmented into three parts, this first segment is combined with the second segment and the third segment and passed again to the neural network. If this grouping was well recognized, then the final character will be the grouping form of the three segments. Otherwise, only the first two segments are combined and passed to the neural network. If the first segment can be a part of any other letter that may over-segmented into two parts, then The final character will be the one with the higher recognition result of the first segment alone and its grouping with the second segment. Finally, this process of grouping and recognition repeated starting from the next unrecognized segment until all resulting segments are recognized as letters or combined into recognized letters. **Figure 6** illustrates an over-segmented letter “ث”, where the second segment was classified as class 5 and the grouping form of the two segments was classified as class 6.

Figure 7 also shows a triple-segmented letter “ص”, where the first segment was classified as class 43 and the combined form of the first, second, and the third segments was returned as class 22.

5. Experimental Results

More than 600 of over-segmented words were tested using the proposed recognition system aided by the above grouping rules. The obtained results are very encouraging, as illustrated in **Table 2**. The percentage of correct segmentation increased from 86% before recognition and grouping rules application to 95% after recognition and grouping. On the other hand, the over-segmentation percentage decreased from 14% to 5% whereas the under-segmentation percentage remained constant.

The majority of the over-segmented characters are combined and recognized correctly. **Figure 8** depicts segmented words before and after recognition and grouping rules application.

As demonstrated above, both double and triple segmented letters are combined correctly. The letter “ش” which appears in the words “الشوامخ” and “بوعطوش” was segmented into three parts and each part is a candidate letter. However, the grouping and recognition processes yield to one strong candidate letter instead of the three parts. A different example of double segmented letters can be observed in letters “ث” and “ي” that belong to words “مارث” and “صحراوي”, respectively. Similarly, the combined segments had higher recognition rate, compared to each segment separately. The same case can be observed in the rest of examples.

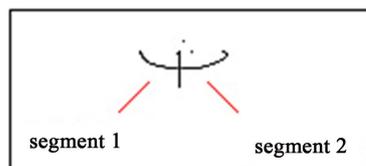


Figure 6. Applied grouping rules to double-segmented letter.

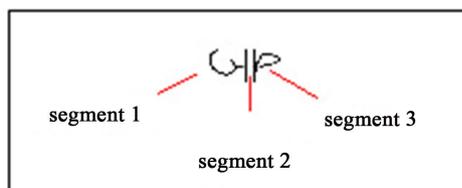


Figure 7. Applied grouping rules to triple-segmented letter.

Table 2. The segmentation results using the selected sample.

	No recognition	After recognition & grouping
% of correct segmentation	86%	95%
% of over segmentation	≈14%	≈5%
% of missing/wrong segmentation	0.3%	0.3%

Before Applying Combination Rules	After Applying Combination Rules
بوعطووش	بوعطووش
البفا بفين	البفا بفين
البراد عبة	البراد عبة
الشوامخ	الشوامخ
الدهخا نبيه	الدهخا نبيه
شماخ	شماخ
قمر طاج	قمر طاج
مارث	مارث
حزوة	حزوة
بوعثمان	بوعثمان

Figure 8. Over-segmented words before and after applying grouping rules.

Because the grouping rules are strongly based on the recognition result of each segment, those which are misrecognized may not help in handling the over-segmented letters. This case appears in the letters of classes: 3, 6, 29, 32, 43, and 44, where the left segment of the over-segmented letter is wrongly recognized as letter alif “ا”, which belongs to class 1. One example is shown in Figure 9, in the letter “ف” of the word “تكريف” and in the letter “ن” in the word “بوعثمان”.

The under-segmentation problem may also occur because of letter misrecognition. Adjacent letters that form shapes similar to those of classes 15, 16, 17, 18, 19, 20, 21 and 22 may wrongly be combined although they are correctly classified before grouping. This scenario is very clear in Figure 9, in the word “اسيدي”, where the adjacent letters “ي” and part of the letter “س” were recognized and classified as letter “س”, and the remaining part of “س” was classified as letter “ل”. Similar case was also detected in the words “مدنين” and “حاسي”, with different classification of the combined segments. In the word “جورمين”, the two letters “ي” and “م” were combined and classified as letter “ص”.

However, these results still fair because the combined segments form a body shape similar to that of an existing alphabetical letter. Moreover, the recognizer is not an interpreter to search for the meaning of the word based on the recognition results of its letters. Therefore, the obtaining outcomes are acceptable since no recent work could solve these situations.

6. Conclusion

In this paper, an effective segmentation method for Arabic handwriting was developed. The method used a multi-agent approach to segment words and relied on recognition to verify the validity of the candidate segmenta-

Over-Segmentation	Wrong OR Under-Segmentation
تكر جيف	للهدي
بوعثمان	جو مين
صبا لبح	صبا نين
تتبعها ل	حا صبي

Figure 9. The problems of character misrecognition.

tion points. The use of an artificial neural network along with grouping rules lead to a good treatment of the over-segmentation problem in Arabic handwritings. Furthermore, it achieved better results, when compared to similar works, by reducing the effect of under segmentation. This is attributed to the decision agent, which makes the proper decisions to identify the candidate segmentation points. The resulting segments are passed to the recognizer, which will invoke and apply the grouping-rules agent on the unrecognized segments before passing it to the recognizer again. The experimental results (~95%) were very satisfactory and promising. Our future direction will focus on improving this approach and including other styles of Arabic handwritings. On the improvement front, currently we are studying the use of SVM and HMM recent and relevant techniques.

References

- [1] Elnagar, A. and Bentrucia, R. (2012) A Multi-Agent Approach to Arabic Handwritten Text Segmentation. *Journal of Intelligent Learning Systems and Applications*, **4**, 207-215. <http://dx.doi.org/10.4236/jilsa.2012.43021>
- [2] Cheriet, M., Kharma, N. and Lui, C.-L. and Suen, Y. (2007) Character Recognition Systems: A Guide for Students and Practitioners. John Wiley & Sons, Inc., Hoboken.
- [3] Elnagar, A. and Alhaji, R. (2003) Segmentation of Connected Handwritten Numeral Strings. *Pattern Recognition*, **36**, 625-634. [http://dx.doi.org/10.1016/S0031-3203\(02\)00097-3](http://dx.doi.org/10.1016/S0031-3203(02)00097-3)
- [4] Vellasques, A.E. Oliveira, L.S., Britto Jr., A.S., Koerich, A.L. and Sabourin, R. (2008) Filtering Segmentation Cuts for Digit String Recognition. *Pattern Recognition*, **41**, 3044-3052. <http://dx.doi.org/10.1016/j.patcog.2008.03.019>
- [5] Al-Badr, B. and Mahmoud, S.A. (1995) Survey and Bibliography of Arabic Optical Text Recognition. *Signal Processing*, **41**, 49-77. [http://dx.doi.org/10.1016/0165-1684\(94\)00090-M](http://dx.doi.org/10.1016/0165-1684(94)00090-M)
- [6] Amin, A. (1998) Offline Arabic Character Recognition: The State of the Art. *Pattern Recognition*, **31**, 517-530. [http://dx.doi.org/10.1016/S0031-3203\(97\)00084-8](http://dx.doi.org/10.1016/S0031-3203(97)00084-8)
- [7] Eldin, A.S. and Nough, A.S. (1998) Arabic Character Recognition: A Survey. *SPIE Proceedings: Optical Pattern Recognition IX*, **3386**, 331-340.
- [8] Khorsheed, M.S. (2002) Off-Line Arabic Character Recognition: A Review. *Pattern Analysis and Applications*, **5**, 31-45. <http://dx.doi.org/10.1007/s100440200004>
- [9] Parhami, B. and Taraghi, M. (1981) Automatic Recognition of Printed Farsi Texts. *Pattern Recognition*, **14**, 395-403. [http://dx.doi.org/10.1016/0031-3203\(81\)90084-4](http://dx.doi.org/10.1016/0031-3203(81)90084-4)
- [10] Amin, A. and Masini, G. (1986) Machine Recognition of Multi-Font Printed Arabic Texts. *Proc. of the 8th IEEE International Joint Conference on Pattern Recognition*, 392-395.
- [11] Gillies, A., Erlandson, E., Trenkle, J. and Schlosser, S. (1999) Arabic Text Recognition System. *Proceedings of the Symposium on Document Image Understanding Technology*, Annapolis, 14-16 April 1999, 253-260.

-
- [12] Hamami, L. and Berkani, D. (2002) Recognition System for Printed Multi-Font and Multi-Size Arabic Characters. *The Arabian Journal for Science and Engineering*, **27**, 57-72.
- [13] Dehghan, M., Faez, K., Ahmadi, M. and Shridhar, M. (2001) Handwritten Farsi (Arabic) Word Recognition: A Holistic Approach Using Discrete HMM. *Pattern Recognition*, **34**, 1057-1065. [http://dx.doi.org/10.1016/S0031-3203\(00\)00051-0](http://dx.doi.org/10.1016/S0031-3203(00)00051-0)
- [14] Al-Qahtani, S.A. and Khorsheed, M.S. (2004) An Omni-Font HTK-Based Arabic Recognition System. *Proceedings of the 8th IASTED International Conference on Artificial Intelligence and Soft Computing*, Marbella, 1-3 September 2004.
- [15] Al-Qahtani, S.A. and Khorsheed, M.S. (2004) A HTK-Based System to Recognize Arabic Script. *Proceedings of the 4th IASTED International Conference on Visualization, Imaging, and Image Processing*, Marbella, 6-8 September 2004.
- [16] Al-Badr, B. and Haralick, R. (1998) A Segmentation-Free Approach to Text Recognition with Application to Arabic Text. *International Journal on Document Analysis and Recognition*, **1**, 147-166. <http://dx.doi.org/10.1007/s100320050014>
- [17] Al-Badr, B. and Haralick, R. (1995) Segmentation-Free Word Recognition with Application to Arabic. *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, Montreal, 14-16 August 1995, 355-359. <http://dx.doi.org/10.1109/ICDAR.1995.599012>
- [18] Khorsheed, M.S. and Clocksin, W.F. (1999) Structural Features of Cursive Arabic Script. *British Machine Vision Conference*, Nottingham, 13-16 September 1999, 422-431. <http://dx.doi.org/10.5244/c.13.42>
- [19] Amin, A. (2000) Recognition of Printed Arabic Text Based on Global Features and Decision Tree Learning Techniques. *Pattern Recognition*, **33**, 1309-1323. [http://dx.doi.org/10.1016/S0031-3203\(99\)00114-4](http://dx.doi.org/10.1016/S0031-3203(99)00114-4)
- [20] Pechwitz, M. and Maergner, V. (2003) HMM-Based Approach for Handwritten Arabic Word Recognition Using the IFN/ENIT-Database. *Proceedings of the 7th International Conference on Document Analysis and Recognition, ICDAR*, Edinburgh, 6 August 2003, 890-894. <http://dx.doi.org/10.1109/icdar.2003.1227788>
- [21] Gonzalez, R.C. and Wintz, P. (1987) *Digital Image Processing*. 2nd Edition, Addison-Wesley, Boston.
- [22] Teuber, J. (1991) *Digital Image Processing*. Prentice Hall International Series in Acoustics, Speech and Signal Processing, Prentice Hall, Upper Saddle River.