Classification of Personal Arabic Handwritten Documents

SALAMA BROOK AND ZAHER Al AGHBARI
Department of Computer Science
University of Sharjah, UAE
zaher@sharjah.ac.ae

Abstract: - This paper presents a novel holistic technique for classifying Arabic handwritten text documents. The classification of Arabic handwritten documents is performed in several steps. First, the Arabic handwritten document images are segmented into words, and then each word is segmented into its connected parts. Second, several structural and statistical features are extracted from these connected parts and then combined to represent a word with one consolidated feature vector. Finally, a generalized feedforward neural network is used to learn and classify the different styles/fonts into word classes, which are used to retrieve Arabic handwritten text documents. The extraction of structural and statistical features from the individual connected parts as compared to the extraction of these features from the whole word improved the performance of the system.

Key-Words: - Data mining of Arabic text, Word recognition, Arabic handwriting, Segmentation of Arabic handwritten documents, Feature extraction, Classification, and Retrieval of Arabic handwritten documents

1 Introduction

Recently, automatic reading of handwritten text has become an important issue. This comes together with the increase use of pen-based interface devices. For example, pen-based personal digital assistants (PDA), replaced the whole keyboard with a pen by which all commands and data entries can be performed. Using this pen, data can be input by the user in the form of handwritten notes. To read the handwritten text offline, first the document is divided into its smallest units (characters/words). This first step is called segmentation and it is an essential step in the recognition process. Next these segmented units are uniquely represented by their features. Then, pattern recognition techniques are used to convert these small units into their equivalent ASCII text. The recognition phase is an intermediate step between the input device (pen or tablet) and the storage device. The importance of such systems is due to the increasing number of applications that utilize the Arabic handwritten text documents such as archiving documents and automatic reading of checks.

Arabic language is a widely used language as more than 1 billion people use Arabic in either their daily activities or religion-related activities. Arabic characters are used to transcribe several languages such as Arabic, Farsi (Persian), and Urdu languages. Although recognition techniques for other languages such as Latin, Chinese, and Indian achieved high rates of recognition, these techniques cannot be directly applied to Arabic handwritten text due to the following characteristics of the Arabic text: (1) the cursive nature even in machine printed form, (2) letter shape is context sensitive, and (3) writing style variability from person to person. Furthermore, offline reading of Arabic text cannot utilize the essential temporal information of the text. Recognition of Arabic handwritten text has been considered by some researchers; however, recognizing handwritten words from images has only been successful in specific domains with limitations [1][2], especially for Arabic text.

Further, handwritten Arabic text recognition devices have low performance due to the characteristics of the Arabic language that are not found in the Latin languages such as:

- Arabic has 28 letters and each letter can assume 2 to 4 different forms depending on its position within the word. For example, the letter “م “, reads as meem, has an isolated form which is “م” , initial form at the beginning of the word “مـ”, in the middle of the word “ـمـ” or at the end of the word “ـم”.

- Arabic is a cursive type language which is written from right to left and the segmentation must follow this.

- Words may have one or more connected parts. This adds another difficulty to the recognition process. For example, the word “مركبة“ , reads as markabah, which means vehicle in Arabic, consists of two connected parts.

- Some characters may form a new ligature shape, which is a vertical stacking of two or more characters. For example, the first two letters: “ل “ and “م “ in the word “لحم”, reads as laham, which means meat, are very difficult to separate.

- Some of the scripts have loops in their structure. For example, the letters “ض“ and “ط “ .

- Some characters have dots on the top, in the middle, and at the bottom such as the letters “ت“, “ج“ and “ب “, respectively.

- Some characters have diacritics, such as the letters “'طَ“, “ ’طُ“ and “ طِ “, respectively.

In this paper, a retrieval technique of Arabic handwritten text is presented. As mentioned above, classification of Arabic words is a very challenging step in the Arabic handwritten text recognition process. For pen-based devices, such as tablet PCs and PDAs, notes are written using the device’s word processor software, saved instantly as an image, then later it will be classified and indexed by our proposed technique. Our classification and retrieval technique is based on segmentation of Arabic handwritten documents into lines, then words, and then each word is segmented into its connected parts. Several features are extracted from these connected parts and then combined to represent the corresponding word with one consolidated feature vector. Then, a generalized feedforward neural network is used to learn and classify the different styles/fonts into word classes, which are used to index and retrieve Arabic handwritten documents.

Existing text recognition systems can be classified into two major classes. The first class of text recognition systems segments a word into its individual characters and then extracts features from these characters, such as [14] and [18], however this approach has not attained high accuracy in performance especially for Arabic text due to the difficulty of the character segmentation phase. The second class of text recognition systems considers the word as the smallest unit and thus extracts global features from the word unit such as [7], [17] and [19], however the global features of a word usually lacks the peculiarities of characters and thus reduce the ability to distinguish between words. In this paper, we segment the word into its connected parts, which is a process that is more accurate than character segmentation. Features extract form these connected parts have more distinguishing power as compared with the features extracted from the word unit and at the same time we avoid the error-prone segmentation of the a word into its characters. Our experiments confirm the feasibility of our approach to recognize and classify Arabic handwritten text.

In Section 2, we survey the related work to Arabic handwritten recognition and classification. Then, we explain our segmentation method in Section 3. The feature extraction step is presented in Section 4. Then, the classification method for Arabic handwritten words is presented in Section 5. In Section 6, we present our experiments. Finally, we conclude the paper in Section 7.

2 Related Work

Variations in handwriting style have presented a challenge to the process of automation of transcribing handwritten text documents. Thus, handwritten text documents are transcribed by hand [3][4], which is tiring, time-consuming, and unreliable task. Reducing such manual work can be achieved by building an automatic word recognition system that segments the handwritten text document into its words. Then, these words are grouped based on their features’ similarities into clusters, where each cluster contains a certain word [5][6]. Indices are constructed on these words (clusters) to facilitate easy access to handwritten documents. Automatic approaches of general-domain handwriting recognition are difficult [1], however recognizing handwriting from images has only been successful in specific-domains with limitations [2]. Furthermore, traditional handwriting recognition approaches require very high accuracy in the feature extraction and recognition phases [7].

Any recognition system must have two main stages [8]. The first stage is *feature extraction*, which extracts measurements from the input text data. The problem of extracting features from the input data is achieved by selecting information that is most relevant to the classification of words and able to discriminate between words. The second stage is *classification*, which determines the class to which the input word belongs.

The feature extraction method used in character recognition systems is probably the most important phase in achieving good recognition rate [9]. There were several feature extraction approaches: statistical and structural. Statistical features are derived from the statistical distribution of the pixels in the document image [10]. On the other hand, structural features describe the geometrical and topological characteristics of the text patterns [11].

Although there have been several approaches to Arabic handwritten text segmentation, segmentation has not achieved a reasonable level of performance [13][14]. One of the reasons that off-line segmentation of Arabic handwritten text has not achieved an acceptable level of performance is that essential temporal information is lost. Therefore, in [15], the authors tried to restore the lost temporal information by finding a connection between the offline and online handwriting. The other reasons for low performance of segmentation of Arabic handwritten text are due to the characteristics of the Arabic language that are not found in the Latin languages, which we discussed in the previous section.

A neural network, or any machine learning technique, is used to classify the extracted features. These neural network techniques are robust to differences in handwriting style and can accommodate new word shapes [12]. To build an application for handwritten text, all phases of the automated handwritten recognition system should be designed carefully and precisely because of the variability and complexity of the problem [8].

3 Arabic Handwriting Segmentation

Segmentation of Arabic handwriting is very challenging because of the fact that words in Arabic may have several connected parts (sub-words), which are separated by spaces. Moreover, Arabic words have the characteristics of cursive nature, the variability of letter shapes, and overlapping between neighboring words or connected parts. The technique used in this paper is based on our observation of the histograms of the lines of text in which inter-word spaces (between words) are normally larger than intra-word spaces (between connected parts). In addition, our technique takes into account the peculiar characteristics of the Arabic handwriting. Thus, segmentation is performed in three steps:

(a) Locating the lines of text.
(b) Locating words in each line of text.
(c) Locating connected parts in each word.

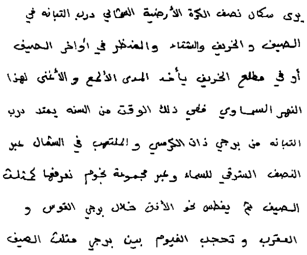
Our segmentation technique reads in a text image and then it segments the text image into lines. Each line of text is segmented into words and then each word is segmented into its connected parts. A connected part is one or more letters connected together and they do not contain a space. For example, the word مخالطة consists of two connected parts; “مخا” and “لطة”.

3.1 Divide whole image to line images

To divide a document image into line images, a pre-processed document image is projected horizontally (see Figures 1 and 2) to create a horizontal histogram that represents the text density in the whole image. Then, the peaks of the horizontal histogram are detected, where the peaks represent the baselines. The average between successive peaks' indices is computed and the resulting value marks the border of a line. This method is independent of sweeping direction. That is the histogram is swept from left to right or from right to left. The *image to line segmentation* algorithm is as follows:

```
Algorithm: image to lines
Input:  document binary image
Output: linet images

  1. Project a document binary image horizonally
     to create a binary histogram.
  2. detect the peaks (baselines) of the
     histogram.
```



يرى سكان نصف الكرة الأرضية الشمالي درب التبانه في
الصيف و الخريف والشتاء والمنظر في أواخر الصيف
أو في مطلع الخريف يأخذ المدى الألمع و الأغنى لهذا
النهر السماوي ففي ذلك الوقت من السنه يمتد درب
التبانه من برجي ذات الكرسي و الملتهب في الشمال عبر
النصف الشرقي للسماء وعبر مجموعة نجوم نعرفها كمثلث
الصيف ثم يغطس نحو الأفق خلال برجي القوس و
العقرب و تحجب الغيوم بين برجي مثلث الصيف

Figure 1: Arabic text document

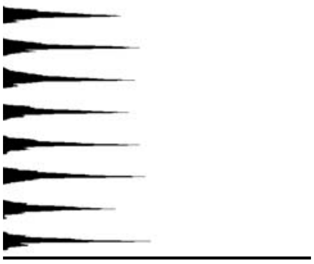


Figure 2: Horizontal projection of the document

```
  3. compute the middle points between every
     two successive peaks and mark these middle
     points as the line borders.
```

3.2 Divide a line image to word images

In this step, a line image is divided into word images by locating the inter-word spaces (columns with white pixels), which separate the words. Therefore, a line image is vertically projected to create a vertical histogram representing the word density in the line image (see Figure 3). The resulted histogram will have some zero-value columns. These zero-value columns draw the boundaries between words or connected parts. The *word segmentation* algorithm is as follows:

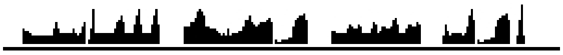


Figure 3: vertical projection of a line of text

```
Algorithm: Line to Words
Input: line binary image
Output: words' images

1. Project the line image vertically to create a binary histogram.
2. Detect the zero-value columns (ranges) and store their beginning and ending indices.
3. Compute and store the widths of these zero-value ranges.
4. Compute a threshold width, τw, by computing the average between the largest and smallest range. Note that the line's beginning and ending range of spaces are removed from the list of ranges since a text can end at the middle of a line (as in the end of a paragraph) or start after an indentation (as in the start of a paragraph).
5. Compare τw to the detected ranges in the line image to detect the boundary of word images:
    a. If τw is greater than the width of the detected range, then this range is an intra-word spacing.
    b. Otherwise the range is an inter-word spacing and thus the word boundary is declared
```

3.3 Divide a word into connected parts

Similar to segmenting a line into words, in this step a word image is divided into connected parts by locating the intra-word spaces (columns with white pixels), which separate the connected parts. Therefore, a word image is vertically projected to create a vertical histogram representing the word density in the word image. The word to connected parts segmentation algorithm is as follows:

```
Algorithm: word to connected parts
Input: word binary image
Output: connected part images

1. Project a word image vertically to create a binary histogram.
2. Detect the zero-value columns (ranges).
3. Divide the word image into connected parts at the zero-value ranges.
```

Figures 4 is an example of the connected parts segmentation and it shows the results of segmenting a word of into its connected parts.

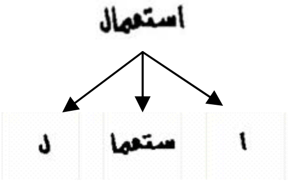


Figure 4: Segmentation of a word into its connected parts.

4 Feature Extraction

Our technique extracts several features from each connected part of a word. Then, these feature vectors are combined to represent the corresponding word with one consolidated feature vector. The images of the connected part extracted from Arabic text documents are of varying font, size and style. An effective representation of the connected part images will have to take care of these variations for successful searching and retrieval. Thus, we extracted two categories of features: *structural* (such as connected part upper/lower profile and projection profile) and *statistical* (such as punctuation count and ration between punctuation and main connect part).

- Structural Features
    - Projection profile
    - Upper profile
    - Lower profile
- Statistical Features
    - Punctuation count
    - Ratio between punctuation and main connect part

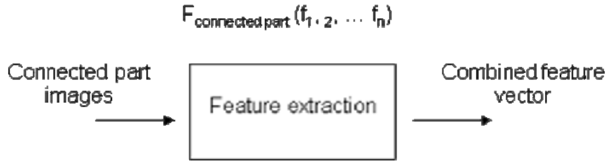


Figure 5: extraction of the connected part feature vector

The images of the connected part, which are binary images, are inputted into the Feature Extraction process, feature vectors that contain the structural and statistical features of the individual connected parts are produced, and then these feature vectors are combined into one consolidated feature vector that represents the corresponding word as shown in Figure 5.

4.1 Structural Features

The Projection profile captures the distribution of ink along one of the two dimensions in a connected-part image, while the upper and lower profiles capture part of the outlining shape of a connected part. To reduce the number of extracted feature values of this feature, we quantized the projection histrogram of a connected part by grouping neighboring columns and computing their average. So, each connected part is represented by a fixed number of groups. In our experiment, we used the group size, *g*, of 10 to represent each connected part.

4.1.1 Projection Profile

The projection profile feature captures the distribution of ink along one of the two dimensions in a word image. A vertical projection profile is computed by summing the intensity values in each connected part image column separately as follows:

$$Pp(I,c) = \sum_{h=1}^{H} 255 - I(r,c) \qquad (1)$$

```
Algorithm: Projection Profile
Input: connected part binary image
Output: feature vector, F, representing projection profile

1. read the image into a two-dimensional array.
2. Divide the width into g groups of columns.
3. for each group compute:
       $Pp(I,c) = \sum_{h=1}^{H} 255 - I(r,c)$
   a. Get the ratio of the number of black to white pixels.
   b. Store the values of step (a) in vector F.
```

4.1.2 Upper & Lower Profiles

The upper and lower profiles capture part of the outlining shape of a connected part. Upper (or, Lower) connected part profile is computed by measure the distance (pixel count) of each group from the top (or, bottom) of the bounding box of the connected part to the closest ink pixel in that group.

```
Algorithm: Upper (or, Lower) Profile
Input: connected part binary image
Output: feature vector, F, representing upper (or, lower) profile

1. Read the image into a two-dimensional array.
2. Divide the width into g groups of columns.
3. for each group
   a. compute the distance from the top (or, bottom) of the bounding box of the
      connected part to the closest ink pixel in that group by counting the number of
      white pixels.
   b. Get the ratio of distance to the number of black pixels of each group .
   c. Store the values of step b in vector F.
```

4.2 Statistical Features

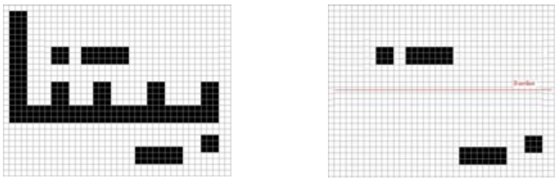
The Punctuation count feature distinguishes words by their punctuations, while the Ratio between punctuation and the main connected part feature captures the differences between punctuations.

4.2.1 Punctuation Count

This feature determines the number of punctuations above and below the baseline of the connected part. First, the algorithm finds the main, usually the biggest, connected part and removes it (see Figure 6a and 6b). The main connected part is usually positioned in the center line (baseline) and punctuations are above or below it. Then, the algorithm counts the number of punctuations above the baseline by projecting all connected parts above the baseline vertically and counting the number of peaks which will be equal to the number of puctuations above the baseline. Similarly, all the connected parts below the baseline are projected vertically and the number of peaks are counted which are equal to the number of punctuations below the baseline.

```
Algorithm: Punctuation Count
Input: connected part binary image
Output: feature vector, F, representing punctuation count above the baseline and punctuation count below the baseline

1. Remove the main connected part.
2. Determine the baseline.
3. Split the image at the baseline into two images.
4. For each of the two images resulted from the previous step:
    a. Project the image vertically.
    b. Count number of peaks in the vertical projection.
```



(a) (b).

Figure 6: (a) original word, and (b) after removing the main connected part.

4.2.2 Ratio between punctuations and main connect part

Connected part punctuation/skeleton ratio feature finds the ratio between each punctuation and the main connected part. The purpose is to distinguish connected parts that have the same main part or skeleton but different punctuations, such as كتب and كبت.

```
Algorithm: Punctuation Ratio
Input: connected part binary image
Output: feature vector, F, representing ratios between punctuations and main connected part

1. Calculate the width of main connected part.
2. Calculate the widths of each of the remaining connected parts (punctuations).
3. for each punctuation,
    a. Compute the ratio between its width and the width of the main connected part.
```

5 Classification by Neural Network

For each segmented word image, the output of the feature extraction phase is a number of feature vectors that is equal to the number of connected parts. An Arabic word may contain one or more connected parts. In the domain of our dataset, the number of connected parts in a single word is between one and five. The combined feature vectors of the connected parts, F_w's, of a word is the input to the neural network and the class to which the word belongs is the output.

The neural network is trained using a training dataset of feature vectors extracted from the segmented words where each segmented word consists of several connected parts. Although, Arabic words have different numbers of connected parts, our technique uses a fixed-size input feature vector for the neural network. The size of the input feature vector F_w of a word is the sum of the sizes of individual connected part's feature vectors, F_{cp}. For example, the word مخالطة consists of two connected parts, “مخا” and “لطة”; therefore, this word is represented by the feature vector $F_{\text{مخالطة}} = (F_{\text{مخا}}, F_{\text{لطة}},0,0,0)$.

The size of each connected part's feature vector is the sum of its extracted features ($F_1 \rightarrow 10 + F_2 \rightarrow 10 + F_3 \rightarrow 10 + F_4 \rightarrow 2 + F_5 \rightarrow 10 = 42$). Thus, Each F_{cp} consists of 42 values. As shown in Figure 7, the input to the neural network is the feature vector $F_w = F_{cp1}+ F_{cp2}+,\ldots,+F_{cp5}$. For example, the word مخالطة will be input to the neural network as $F_{\text{مخالطة}}$ ($F_{1\ =\ 42\ values}$, $F_{2=42\ values}$, 0, 0, 0). The output of the neural network will be the class that represents the input word. Note, that $F_{\text{مخالطة}}$ contains the values of two connected parts and then zeros are padded in place of the other three connected parts.

In this system, the generalized feedforward of MLP neural network [16] is used to classify the input feature vectors. Generalized feedforward networks are a generalization of the MLP such that connections can jump over one or more layers. The user simply specify the number of layers, and the system constructs a MLP in which each layer feeds forward to all subsequent layers. In theory, a MLP can solve any problem that a generalized feedfoward network can solve. In practice, however, generalized feedforward networks often solve the problem much more efficiently.

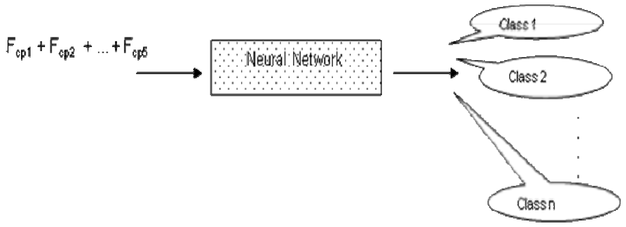


Figure 7: Classification of word's feature vectors

Our neural network contains three layers, namely, input, output and hidden layers. In the training dataset, every word is written in 10 different styles by 10 different human subjects and thus there are 10 feature vectors for every word. The neural network is trained with 5 different styles of every word in the training dataset. As a result of the training phase, classes are generated where each class contains feature vectors of the same word. In other words, a class contains similar feature vectors. A representative feature vector represents each class. New words will be classified based on their similarity to the representative feature vectors. The input layer of the neural network consists of 210 nodes (size of the feature vector of a word) and the output layer consists of 1100 nodes (bigger than the number of words in the used database domain). Each output node represents a class (a word).

5 Experiments

The experiments were performed on a personal Arabic handwritten text documents. We asked 10 different human subjects to write sample documents in their own handwriting styles. The documents have been chosen from an Arabic book entitled “A journey in the galaxies' history”"رحلة في تاريخ المجرات", then these documents were scanned, pre-processed, segmented into lines, words and connected parts, and then the feature vectors were extracted.

We noticed that for personal writing (everyday writing by individuals) or writing by using pen-based devices, users tend to leave larger spaces between words (inter-word spaces) than those between the connected parts of a word (intra-word spaces). As seen from the result of our experiments, our segmentation algorithm is robust to different styles of personal Arabic handwriting (see Tables 1). The computed accuracy evaluation is based on the following equation:

$$Accuracy = \frac{S_{cw}}{T_w} \qquad (2)$$

Where S_{cw} is the number of correctly segmented lines, words, or connected parts and T_w is the total number of lines, words, or connected parts, respectively, in the documents.

Table 1: Segmentation results of each document images to line images, line images to word images, and word images to connected parts images

Document No.	Average Accuracy		
	Document to lines	Lines to words	Word to connected parts
Doc1	100%	100%	100%
Doc2	100%	90%	92%
Doc3	100%	100%	100%
Doc4	100%	100%	100%
Doc5	100%	100%	100%
Doc6	100%	100%	98.6%
Doc7	100%	100%	100%
Doc8	100%	100%	100%
Doc9	100%	100%	100%
Doc10	100%	100%	95.3%

Table 1 shows the results of segmentation. The average segmentation accuracies shown in Table 1 are computed over all 10 different writing styles. As seen from Table 1, the results are very encouraging for personal Arabic handwriting. For the *document to lines* segmentation, the problem is easy since the lines are well separated and thus our algorithm can easily detect the line borders. For the *line to words* segmentation in Doc2 (see Table 1), some words overlap with each other, or touch each other, in some writing styles, which is reduces the ability to distinguish between intra-word and inter-word spacing. For the *word to connected parts* segmentation, the characters of a word are disconnected in some places where they should not be disconnected due to the handwriting style of that individual (see Figure 8). Such situations may affect the overall result significantly and result in over-segmentation. On the other hand, some connected parts of a word may overlap which results in under-segmentation. Such over-segmentation and under-segmentation are clearly noticed in documents 2, 6 and 10.

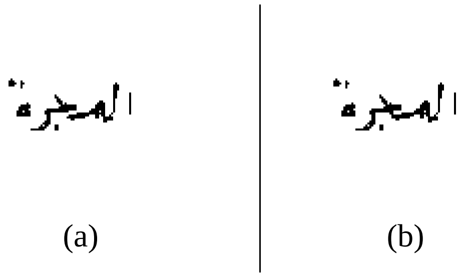


Figure 8: (a) Word contains discontinuities, (b) Same word without discontinuities

We divided the collected dataset into two subsets; the first subset contains five different writing styles, which are written by 5 different human subjects, and the second subset contains the other 5 writing styles. The documents in the first subset were used to train the network and the documents of the second subset were used to test our technique. The documents of the second subset were classified by the neural network and the accuracy was computed according to Equation 1. The Average accuracies shown in Table 2 were computed over the all the words in each writing style, where each writing style consists of 50 different words

We compared the classification accuracy of our technique in which we represent a word by the combined feature vector of the feature vectors its individual connected parts with the common holistic technique that represents a word by a feature vector without dividing the word into smaller units. As shown in Table 2, the average accuracies of classification using our technique are higher that those of the holistic technique for all the tested writing styles. We can conclude that our technique improved the classification of Arabic handwritten words and it is robust to different styles of personal Arabic handwriting. We compared the classification accuracy of our technique in which we represent a word by the combined feature vector of the feature vectors its individual connected parts with the common holistic technique that represents a word by a feature vector without dividing the word into smaller units. As shown in Table 2, the average accuracies of classification using our technique are higher that those of the holistic technique for all the tested writing styles. We can conclude that our technique improved the classification of Arabic handwritten words and it is robust to different styles of personal Arabic handwriting.

Table 2 : Accuracy of the tested 5 writing styles

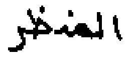
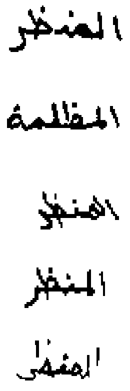
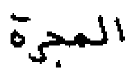
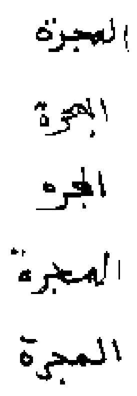
Style sample	**Average Accuracy of classification with connected parts**	**Average Accuracy of classification with whole words**
Writing Style 6	98%	90 %
Writing Style 7	100 %	90 %
Writing Style 8	95 %	92 %
Writing Style 9	96%	88 %
Writing Style 10	91%	80 %

Table 3 shows the output of two input words, where one input word, المجرة , has perfect match from all five writing styles and the other input word, المنظر, has one mismatch from a similar word. The left column shows the input word to the system, needless to say that the input is the feature vector that represents this word but we include the image of the word in this table for clarification. The middle column shows the matching words to the input word. The right column shows the index value of each output word.

5 Conclusion

In this paper, we presented a technique to classify Arabic handwritten documents. The technique utilizes the density distribution of Arabic handwritten text to find the boundaries between lines, words and connected parts. Then several features that capture the peculiarities of Arabic handwriting, such as the dots and diacritics, were extracted from these connected parts to represent their corresponding words. As seen from the result of our experiments, our technique is robust to different styles of Arabic handwriting. Although the proposed system is applied to Arabic handwritten documents, it can be adapted to other languages' handwriting auch as Latin.

Table 3 : The system output of two words “Al Manthar, المنظر” and “Al Majarah, المجرة”

Input word	NN output
	
	

References:

[1] Tomai C. I., Zhang B. and Govindaraju V. “Transcript Mapping for Historic Handwritten Document Images”. In: Proc. Of the 8th Int’l Workshop on Frontiers in Handwriting Recognition 2002, pp. 413-418, August 6-8, 2002.

[2] Rath. T., Lavrenko, V. and Manmatha, R., "Retrieving Historical Manuscripts using Shape" CIIR Technical Report, 2003.

[3] Manmatha, R.and Rath, T.M., "Indexing Handwritten Historical Documents - Recent Progress" in Indexing Handwritten Historical Documents - Recent Progress | the Proc. of the Symposium on Document Image Understanding (SDIUT-03), pp. 77-85, 2003.

[4] Rath T. M. and Manmatha R. “Word Image Matching Using Dynamic Time Warping”**.** In: Proc. of the Conf. on Computer Vision and Pattern Recognition (CVPR), Madison, WI, vol. 2, pp. 521-527, June 18-20, 2003.

[5] Kane, S., Lehman, A. and Partridge, E., "Indexing George Washington's Handwritten Manuscripts" to appear in CIIR Technical Report, 2001.

[6] Rath T. M., Kane S., Lehman A., Partridge E. and Manmatha R.: “Indexing for a Digital Library of George Washington's Manuscripts - A Study of Word Matching Techniques”. CIIR Technical Report MM-36, 2002.

[7] Madhvanath S. and Govindaraju V. “The Role of HolisticParadigms in Handwritten Word Recognition”. Trans. on Pattern Analysis and Machine Intelligence 23:2 ,149-164, 2001.

[8] Hasan Al-Rashaideh , “Preprocessing phase for Arabic Word Handwritten Recognition”, 2006.

[9] O. D. Trier, A. K. Jain, and T. Taxt, “Feature Extraction Methods For Character Recognition – A Survey,” *Pattern Recognition*, 29 (4) (1996), pp. 641–662.

[10] Bazzi I, Schwartz R, Makhoul J, “An Omnifont open-vocabulary orc system for English and Arabic”, IEEE Trans. On Pattern Analysis and Machine Intelligence, 1999, 21(6), pp. 495-504.

[11] Khorsheed MS, Clocksin WF, “Structural features of cursive Arabic script”, Proceedings of 10th British Machine Vision Conference, Nottingham, UK, 1990, pp. 422-431.

[12] Alazim HA, “A hybrid fuzzy-neural approach to the recognition of Arabic script”, The 5th International Conference and Exhibition on Multi-Lingual Computing, Cambridge, UK, 1996.

[13] Al-Badr B. and Mahmoud S., "Survey and bibliography of Arabic optical text

recognition", Signal Processing, 41:49--77, 1995.

[14] Khorsheed M. S. “Off-Line Arabic Character Recognition”, A Review. Pattern Analysis and Applications. Vol1.2, No.1, pp 31-45, 2002.

[15] Abuhaiba I.S.I. and Ahmed P., "Restoring of temporal information in off-line handwriting", Patt. Recog., vol. 26, N°7, pp: 1009-1017, 1993.

[16] Laurene Fausett,. "Fundamentals of Neural Networks : Architectures, Algorithms and Applications", Prentice-Hall, 1994, ISBN 0-13-334186-0.

[17] Ataer E. and Duygulu P., "Retrieval of Ottoman Documents", International Workshop on Multimedia Information Retrieval (MIR'06), USA, Oct. 2006.

[18] Omar A.J, Samer A.K, Bashar A.G., Mohamed F., Hani K., “A new Algorithm for Arabic Optical Recognition”, WSEAS Trans. in Information Science and Applications, vol. 3, no. 4, 2006.

[19] Joe A., Trenton P., Yfantis E., Dean C. “Methods and Techniques in Handwritten Form Recognition”, WSEAS Trans. in Information Science and Applications, vol. 3, no. 3, 2006.