J. Shanbehzadeh, H. Pezashki, A. Sarrafzadeh, 'Features Extraction from Farsi Hand Written Letters', *Proceedings of Image and Vision Computing New Zealand 2007*, pp. 35–40, Hamilton, New Zealand, December 2007.

Features Extraction from Farsi Hand Written Letters

Jamshid Shanbehzadeh[1], Hamed Pezashki[2], Abdolhossein Sarrafzadeh[3]

[1]Tarbiat Moalem Unv., Teheran, I. R. Iran, [2]Islamic Azad Unv.-Science and Research Branch, Teheran, I. R. Iran
[3]Institute of Information and Mathematical Sciences, Massey Unv., Auckland, New Zealand

Email: Jamshid@saba.tmu.ac.ir, H.A.Sarrafzadeh @massey.ac.nz

Abstract

Recognition of handwritten Farsi letters is complicated because of the similarity between letters and the different styles of writing. This paper proposes a new set of features for handwritten Farsi letters. This set is a combination of two groups of features to distinguish similarity the letters. The first group of three features explains the general structure of a letter including the number of components. These features are employed to find the best match for a letter. The second group includes seventy five features. These features are extracted from partitioning a letter into smaller parts. Such smaller parts are generated by dividing the letters into smaller frames. Features extracted from the frames are suitable to distinguish structurally dissimilar letters. Vector quantization has been employed to test the features and we have tested the new features on 3000 letters. The new algorithm provides 87% accuracy in average for handwritten Farsi letters.

Keywords: Farsi letters, feature extraction, letter recognition, OCR

1 Introduction

Despite the use of electronic documents, the amount of printed and handwritten documents has never decreased. This has posed a lot of difficulty in document storage, retrieval, search and update, however, electronic documents are appropriate for these purposes. Document image analysis and recognition covers the algorithms to transform documents into electronic format suitable for storage, retrieval, search and update. Every language has its own characteristics and this affects the analysis and recognition processes. The important characteristics of Farsi/Arabic words that make text analysis and recognition difficult are character connectivity and different shapes of characters depending on their location within the word.

Document analysis and recognition consists of five steps. The first step obtains an image document from the text using a scanner. The second step is the pre-processing to remove the artifacts from the scanned image. The third step segments the document into basic elements. The basic elements can be sub-words or characters depending on the scheme. In situations where we have an infinite number of words for recognition, they have to be segmented into characters. Otherwise, we can segment the words into sub-words. After segmentation, we have to extract features from the basic elements. The extracted features are the input of the recognition step. This paper presents a part of a large project in which the required pre-processing and segmentation steps have been performed successfully by Marashi and Shanbehzadeh [26] and Rastegarpour and Shanbehzadeh [27]. The focus of this paper is on feature extraction.

In general, features can be divided into two groups; structural and statistical. Structural features are related to the appearance of the text. Circles, periods, and dots of letters are among these features. Statistical features are numerical measurements of text's image such as accumulation of pixels. In 1987, Almuallim and Yamaguchi [25] presented one of the earliest methods for recognizing Arabic manuscript texts. In this method the letters' skeleton and structural features were used to recognize the word. In 1992, a structural method was proposed by Gorain [5]. In 2001, Amin [7] utilized structural information that describe the letters including lines, curves and circles for Arabic letter recognition. In 2005, Al-Taani [8] suggested a structural method to identify Arabic numbers. He described how the numbers were recognized based on primary figures such as curves, lines and forms using his method. Structural features vary in font and personal writing style. Thus, they cannot be easily extracted, but the recognition level can be improved by combining several features. The selected features depend on the language [9].

There are three schemes to extract the feature from the skeleton, contours or pixels of characters. The first scheme is based on features extracted from the skeleton of letters. In 1994, Abuhaiba et al. [16] suggested a collection of graph models for recognition of distinct letters based on the skeleton of the letters. The skeleton was converted into a tree structure and compared with a model using an indicator. In 1996, Amin et al. [15] conducted the recognition based on the letters' skeleton using a graph method. In 1998,

Abuhaiba et al. [16] proposed a system which utilized letters' skeleton for recognition of manuscript text. The method of feature extraction developed by Dehghani et al. [18] is based on the letter's contours. In this scheme, the contours of a word are obtained and the whole word is divided into frames. Then features are extracted from these frames. In 2003, Clocksin and Fernando [19] conducted recognition for Syriac texts. Syriac language is grammatically simpler than Arabic. In this method, the whole image of the distinct letter was employed. In 2005, EL-Hajj et al. [20] suggested a method in which the features were obtained from above and below the line in a frame and were given to a Markov recognizer in which features such as the accumulation of pixels and the letter's concavity were used. In 2005, Asiri and Khorsheed [21] proposed a scheme in which wavelet coefficients were employed. The wavelet coefficients were obtained for each letter and were then passed on to a neural network for recognition [21]. In the following section, we will explain our proposed method which obtains statistical features from the pixels of the text's image.

The dots of letters convey significant information for recognizing Farsi letters. In Farsi language, some letters will have identical shapes if their dots are removed and their difference is only in the number and the location of dots. For instance, the letters "پ" and "ت" would have an identical shape without the dots and their difference is only in the number and the location of their dots. In "پ" there are three dots below the main body but "ت" has two dots which are located above the main body. Previous methods had no emphasis on the dots as they were inherently designed for English letters which have few dots. But the dots play significant role in the recognition of Farsi letters. This paper utilizes important information contained in letters' dots in the recognition phase. The structure of the rest of this paper is as follows. The next section presents the new feature extraction algorithm. Section 3 discusses the experimental results. Conclusions are presented in Section 4.

2 The New Feature Extraction Scheme

The extracted features consist of two parts. The features of the first part distinguish the letters with similar parts and, the features of the second part distinguish the letters with dissimilar main body. In the first part, the features are obtained from the whole letter's image and for each letter information such as the number of dots and their location with respect to the main body, is obtained. In the second part, at first the dots are removed from the letters then, the remaining part is divided into smaller frames (windows) from which the statistical information is obtained to generate features for the main body of the letter.

2.1 Features to Distinguish Similar Letters

In this part we employ the information on letters' dots to generate features. These features are useful in distinguishing letters that are only different in the number and the location of their dots with respect to the main body of the letter. First, we obtain the letter's skeleton and then identify the number of isolated parts of the letters by utilizing connected component analysis. To do this, first the binary picture and then the skeleton of the letter is obtained. The biggest component is the main body of the letter and the remaining parts are the dots. Figure 1 illustrates an example for the letter "ژ". Figure 1.a is the original letter, Figure 1.b shows its skeleton and Figure 1.c is the output of the connected component analysis. As shown in 1.c the letter consists of 4 connected components. The biggest part represented by 1's is the main body and the remaining parts, represented by 2's, 3's and 4's are the dots.

Using this information we consider the following features for letters. The number of components of each letter is considered as the first feature (f1). For example, letter "ژ " consists of four parts. By counting the number of pixels in each part and, by considering the fact that the main body of the letter has the most pixels, the dots and the main body of the letter can be separated. We consider the number of the dots as the second feature (f2). After obtaining the dots, we specify the location of dots relevant to the baseline. The baseline can be found by finding the center of letter. We show the dots located above the line by 1, otherwise by -1. An example is presented in Figure 2.

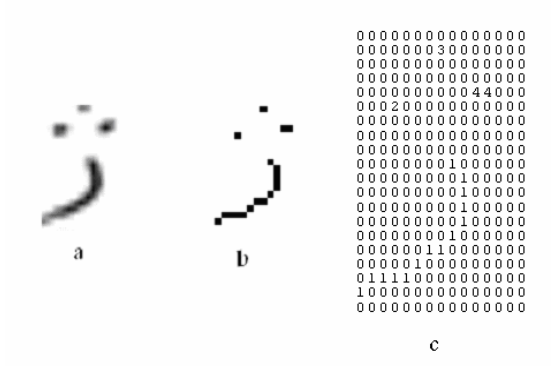

Figure 1:a. letter "ژ " b. skeleton of "ژ " c. separated parts of the letter represented by various numbers.

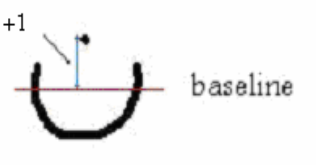

Figure 2: Location of a sample letters' dot.

2.2 Features to Distinguish Dissimilar Letters

Figure 1 shows a sample letter and its main body. Features are extracted from the main body of the letter. To create the feature vector, the main body is divided into vertical frames. The height and width of the frame is constant and is considered as one of the system parameters. Each frame is divided into five cells as proposed in [20], [22]. The height of these cells is fixed (here, it is considered 10 pixels- Figures 3 and 4).

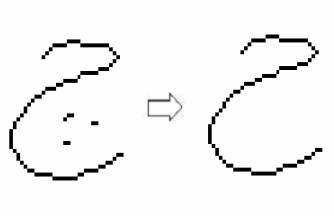

Figure 3: The letter's skeleton and its main body after the dots are removed.

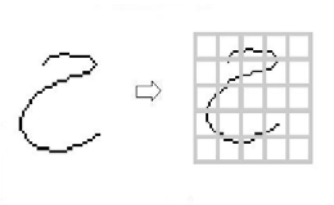

Figure 4: Dividing letters into vertical frames

We generate 15 features for each frame. There are two groups of features: the first group is the distributive features based on the accumulation of the foreground pixels or the black pixels and the second group is the concavity features [20].

2.2.1 Pixels Distribution Features

Suppose H is the height of the frame in each picture, **h** is the height of each cell and **W** is the width of each frame. The number of cells in each frame n_c is shown in (1).

$$n_c = H/h \qquad (1)$$

Here the frame is considered 50 pixels high, the cell is 10 pixels high, and each frame is considered 10 pixels wide. As a result, n_c equals 5. Suppose *r(j)* is the number of black pixels in the j^{th} row of a frame, *n(i)* is the number of black pixels in cell *i* and *b(i)* is the level of cell *i* based on a threshold level. If the *n(i)* of cell *i* is less than the threshold value then we assigns 0 to it, otherwise, we assign 1. This procedure is presented in (2).

If n(i) less than threshold value
then b(i)=0 (2)
else b(i)=1

We consider a boundary for the baseline, and call the lower and upper baseline L and U, respectively. For each frame the following features are extracted. The first feature, f1, is the accumulation of foreground pixels (black) as shown in (3).

$$f_1 = \sum_{i=1}^{n_c} n(i) \qquad (3)$$

Considering the accumulation level of a frame's cell accumulation level of each cell will be *0* or *1*. The second feature (f_2) is the sum of all the accumulation levels of the cells of a frame.

The third feature is the sum of difference of *b(i)* of successive cells of a frame as presented in (4).

$$f_3 = \sum_{i=2}^{n_c} \left| b(i) - b(i-1) \right| \qquad (4)$$

The forth feature shows the difference between the gravitational center of black pixels of frame t and its previous frame which is calculated using (5). The position of gravitational center is determined using (6).

$$f_4 = g(t) - g(t-1) \qquad (5)$$

$$g = \frac{\sum_{j=1}^{H} j.r(j)}{\sum_{j=1}^{H} r(j)} \qquad (6)$$

The vertical position of the gravitational center in each frame is considered as the ninth feature. This feature is normalized by the height of each frame and is determined using (7).

$$f_5 = \frac{g - L}{H} \qquad (7)$$

The sixth feature is similar to the third one but only those cells that are above the lower baseline are considered. In (8), k is the cell containing the lower baseline.

$$f_6 = \sum_{i=K}^{n_c} \left| b(i) - b(i-1) \right| \qquad (8)$$

The seventh feature, f7, indicates an area to which the gravitational center of the black pixels belongs. This area is considered based on the lower and upper baseline. Practically, these two baselines divide the frame into 3 areas. The area above the upper baseline (f7=1), the central area (f7=2) and the area below the lower baseline (f7=3).

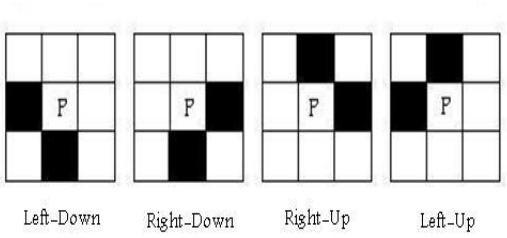

Figure 5: Four forms of concavity for the background pixel P

2.2.2 Local Concavity Features

The concavity features show the local concavity information and the direction of movement in each frame. Each of the concavity features, f8 to f11 are the indicators of white pixels (background) that create the four forms of concavity using 3×3 windows shown in Figure 5. The four concavity features are estimated as follows: Suppose Nlu (in the same form as Ndl, Nrd, Nur) is the number of white pixels which neighbour black pixels in left and up direction (in the same way as right-up, right-down and left-down) in each frame. These four features are defined in each frame as in (9).

$$f_{11} = \frac{Ndl}{H} \; f_{10} = \frac{Nrd}{H} \; f_9 = \frac{Nur}{H} \; f_8 = \frac{Nlu}{H} \quad (9)$$

Using the information gained from two baselines (the upper and lower baseline), we define four more features that indicate the concavity in the central region of the word, i.e. the region limited by two upper and lower baselines. Let **d** be the distance between two baselines (**d**=U-L). Also suppose CNZlu (in the same way as CNZdl, CNZrd, CNZur) is the number of white pixels in the central region such that they neighbour black pixels in left-up direction (in the same way as right- up, right-down and left-down). The four concavity features that depend on the baseline are presented in (10) [20].

$$f_{14} = \frac{CNZrd}{d}, \; f_{15} = \frac{CNZdl}{d} \qquad (10)$$
$$f_{12} = \frac{CNZlu}{d}, \; f_{13} = \frac{CNZur}{d}$$

We will have a vector for each frame with 15 features, 10 of which are independent of the baseline and the rest are estimated based on the location of the baseline. We normalize all the features to achieve a similar bound for comparison and training. Formula 11 is employed for normalization.

$$NewValueforEachFeature = \frac{OldValueOfFeature}{MaximumValueOfFeature} x10 \quad (11)$$

These features are suitable for texts that could be divided into three regions (body, upward moving and downward moving) such as Farsi, Arabic or connected Latin texts. This causes feature extraction not to be dependent on language.

3 Experimental Results

Farsi language consists of 32 letters [23]. Each letter may have up to four forms depending on its location within a word [24]. The database used in the experiment consists of 3000 letters obtained from handwritings by various people. Each letter is normalized to 50*50 pixels. The features are extracted from 60% of samples in the database. After normalization, the feature vectors are classified by Vector Quantization [28]. The experiment is performed on the other 40% of the letters.

Table 1: Recognition rate for each letter.

Letter	Rec. Rate	Letter	Rec. Rate	Letter	Rec. Rate
ا	98.2	ر	87.9	ف	88.6
ب	85.3	ز	87.2	ق	82.1
پ	83.7	ژ	85.6	ک	90.5
ت	84.6	س	82.1	گ	84.6
ث	84.1	ش	81.4	ل	93.6
ج	76.9	ص	84.6	م	95.8
ح	78.5	ض	83.7	ن	85.6
خ	78.2	ط	87.5	و	96.3
چ	77.4	ظ	87.4	ه	97.8
د	89.7	ع	85.3	ی	90.5
ذ	84.8	غ	84.6		

Table 2: Recognition rate for each feature.

Feature	Recognition Rate	Feature	Recognition Rate
f_1	80.5	f_9	56.1
f_2	76.1	f_{10}	31.6
f_3	50.5	f_{11}	70
f_4	71.6	f_{12}	60.5
f_5	62.7	f_{13}	67.2
f_6	73.3	f_{14}	54.4
f_7	69.4	f_{15}	36.1
f_8	77.2		

The performance of Vector Quantization depends on the length of the feature vector and the number of code vector for each letter. Each feature has an independent effect on the recognition performance. Table 1 presents the accuracy for each letter based on a codebook of size four for each letter. The recognition rate based on individual components of the feature vector is shown in Table 2. We performed the experiments based on those features that had 70% accuracy. These features are f1, f2, f4, f6, f8 and f11. We achieved 85.59% accuracy with these five features. As a result, depending on the factor that is more important to us, e.g. recognition rate or the time, we choose one of these cases.

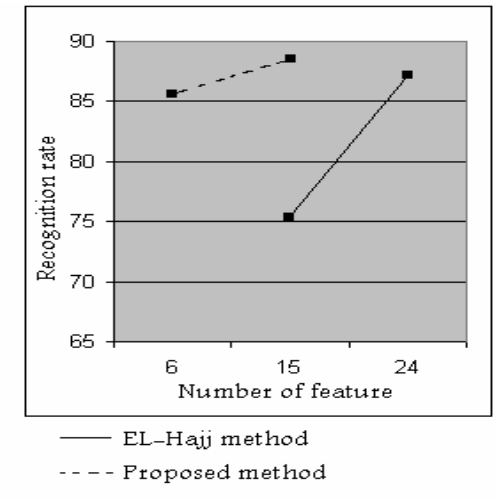

Figure 6: Comparison of Schemes

4 Conclusion

In 2005, EL-Hajj [20] suggested a method in which each letter was divided into several frames and for each frame 2 groups of features were obtained. The total number of these features was from 15 to 25. A comparison of the new features introduced in this paper to those in [20] is presented in Figure 6. It can be seen that the new algorithm is capable of providing better performance with fewer features. This would results in better performance in terms of accuracy and speed.

References

[1] M. M. Haji, "Farsi Handwritten Word Recognition Using Continous Hidden Markov Models and Structural Features, MSc Thesis, Comp. Eng. Dept., Shiraz Unv. Shiraz , Iran, 2005.

[2] M.S. Khorsheed, "Off-Line Arabic Character Recognition A Review," *Pattern Analysis and Applications*, vol. 5, pp. 31-45, 2002.

[3] K. Saeed, M. Tabedzki, "Intelligent Feature Extract System for Cursive-Script Recognition", MSc Thesis, The University of Finance and Management in Bialystok, Poland, 2005.

[4] A. Amin, "Off-Line Arabic Character Recognition: The State Of The Art", *Pattern Recognition Society*, Elsevier Science, vol. 31, No. 5, pp. 517-530, 1998.

[5] H. Goraine, M. Usher, and S. Al-Emami, "Off-Line Arabic Character Recognition," *Computer*, vol. 25, pp. 71-74, 1992.

[6] L. M. Lorigo and V. Govindaraju,"Offline Arabic Handwriting Recognition: A Survey", *IEEE Trans. On Pattern Analysis and Machine Intelligence*, Vol. 28, NO. 5, pp. 712- 724, 2006.

[7] A. Amin, "Recognition Of Hand-Printed Characters Based On Structural Description and Inductive Logic Pogramming", *Sixth International Conference on Document Analysis and Recognition*, Seattle, Washington, USA, 2001 pp. 333-337.

[8] A. T. Al-Taani, "An Efficient Feature Extraction Algorithm for the Recognition of Handwritten Arabic Digits", *I. J. Comp. Intelligence* 2 (2), pp. 107-11, 2005.

[9] M. Z. Khedher, G. A. Abandah, and A. M. Al-Khawaldeh, "Optimizing Feature Selection for Recognizing Handwritten Arabic Characters", *Trans. on Engineering, Computing and Technology*, vol. 4 Feb, 2005.

[10] S. Mozaffari, K. Faez, H. Rashidy Kanan, "Feature Comparison between Fractal Codes and Wavelet Transform in Handwritten Alphanumeric Recognition Using SVM Classifier", *Proceedings of the 17th IEEE Int. Conf. on Pattern Recognition*, Cambridge, UK, 2004.

[11] S. Mozaffari, K. Faez, M. Ziaratban, "Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters. *Proc. of the Eight IEEE International Conference on Document Analysis and Recognition*, Seoul, Korea, 2005.

[12] H. Al-Yousefi and S.S. Udpa, "Recognition of Arabic Characters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, 853-857, 1992.

[13] J. Cowell, F. Hussain, "Extracting Features from Arabic Characters", *The International Conference on Computer Graphics And Imaging (CGIM2001)*, Hawaii, 2001

[14] I.S.I. Abuhaiba, S.A. Mahmoud, and R.J. Green, "Recognition of Handwritten Cursive Arabic Characters", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 664-672, 1994.

[15] A. Amin, H. Al-Sadoun, and S. Fischer, "Hand-Printed Arabic Character Recognition System Using an Artificial Network", *Pattern Recognition*, vol. 29, pp. 663-675, 1996.

[16] I.S.I. Abuhaiba, M.J.J. Holt, and S. Datta, "Recognition of Off-Line Cursive Handwriting," *Computer Vision and Image Understanding*, vol. 71, pp. 19-38, 1998.

[17] A. Dehghani, F. Shabani, and P. Nava, "Off-Line Recognition of Isolated Persian Handwritten Characters Using Multiple Hidden Markov Models", *Proc. Int'l Conf. Information Technology: Coding and Computing*, pp. 506-510, 2001

[18] M. Dehghan , K. Faez , M. Ahmadi , M. Shridhar, ”Unconstrained Farsi handwritten word recognition using fuzzy vector quantization and hidden Markov models“, *Pattern Recognition Letters*, vol. 22, pp. 209-214, 2001.

[19] W.F. Clocksin and P.P.J. Fernando, “Towards Automatic Transcription of Syriac Handwriting”, *International Conference on Image Analysis and Processing*, Mantova, Italy, September 2003, pp. 664-669.

[20] R. El-Hajj , L. Likforman-Sulem, C. Mokbel, “Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling”, *Proceedings of the 8th International Conference on Document Analysis and Recognition (ICDAR’05)*, Seoul, Korea, 2005

[21] A. Asiri, and M. S. Khorsheed, ”Automatic Processing of Handwritten Arabic Forms Using Neural Networks”, *Trans. On Eng., Computing & Technology*, vol. 7, pp.313-317, 2005.

[22] A. Ahmadi, S. Omatu, M. Yoshioka, “Off-line Persian Handwritten Recognition Using Hidden Marrkov Models”, *Proceedings of the Annual Conference of the Institute of Systems, Control and Information Engineers*, Japan, 2002, pp. 231-232.

[23] M. Salmani Jelodar, M.J. Fadaeieslam, N. Mozayani, M. Fazeli, “A Persian OCR System using Morphological Operators”, *The Second World Enformatika Conference, WEC'05*, February 2005, Istanbul, Turkey, pp. 137-140.

[24] P. Burrow, “Arabic Handwriting Recognition”, Master of Science, School of Informatics University of Edinburgh, 2004.

[25] H. Almuallim and S. Yamaguchi, “A Method of Recognition of Arabic Cursive Handwriting”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 9, pp. 715-722, 1987.

[26] N. Marashi, J. Shanbehzadeh, ”Comparison of Local and Global Thresholding for Binarization of Cheque Images”, *Progress in Pattern Recognition Conference*, UK, 2007, pp. 171-178.

[27] M. Rastegarpour, J. Shanbehzadeh, “Off-Line Hand-Written Farsi/Arabic Word Segmentation into Subword under Overlapped or Connected Conditions”, *Progress in Pattern Recognition Conference*, UK, 2007, pp. 186-194.

[28] R.M. Gray, “Vector Quantization”, *IEEE Acoust. Speech Signal Processing Mag.*, pp. 4-29, 1984.