International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:2, No:7, 2008

Multi-Font Farsi/Arabic Isolated Character Recognition Using Chain Codes

H. Izakian, S. A. Monadjemi, B. Tork Ladani, and K. Zamanifar

***Abstract*—**Nowadays, OCR systems have got several applications and are increasingly employed in daily life. Much research has been done regarding the identification of Latin, Japanese, and Chinese characters. However, very little investigation has been performed regarding Farsi/Arabic characters recognition. Probably the reason is difficulty and complexity of those characters identification compared to the others and limitation of IT activities in Farsi and Arabic speaking countries. In this paper, a technique has been employed to identify isolated Farsi/Arabic characters. A chain code based algorithm along with other significant peculiarities such as number and location of dots and auxiliary parts, and the number of holes existing in the isolated character has been used in this study to identify Farsi/Arabic characters. Experimental results show the relatively high accuracy of the method developed when it is tested on several standard Farsi fonts.

***Keywords*—**Farsi characters, OCR, feature extraction, chain code.

I. Introduction

THE Farsi language has got 32 characters which 28 of them are common with Arabic. These Characters are written in cursive form and location and number of dots play an important role in the identification of characters. The sentences are written from right to left in contrast to the Latin languages. Properties of Farsi/Arabic characters have been given in [1-4] in details. Due to the continuity of Farsi and Arabic texts, it is more difficult to identify these characters than Latin characters, especially when we have dealing with various fonts. Most available OCR systems have been designed for recognition of Latin, Chinese, and Japanese characters [5-9]. Systems developed for identifying Farsi/Arabic characters are not still so efficient and leave much place for further investigations in this field. Fig. 1 shows Farsi characters, while Fig. 2 illustrates a few Farsi sentences where the separation of words but continuity of letters within a word is obvious. Fig. 3 shows some of the Farsi characters in different standard MS-Windows fonts.

ت	پ	ب	آ , ا
'te'	'pe'	'be'	'alef'
ح	چ	ج	ث
'he'	'che'	'jim'	'ce'
ر	ذ	د	خ
're'	'zal'	'dal'	'khe'
ش	س	ژ	ز
'sheen'	'seen'	'jhe'	'ze'
ظ	ط	ض	ص
'za'	'ta'	'zad'	'sad'
ق	ف	غ	ع
'ghaf'	'fe'	'ghein'	'ein'
م	ل	گ	ک
'mim'	'lam'	'gaf'	'kaf'
ی	ه	و	ن
'ye'	'ha'	'vav'	'noon'

Fig. 1 The Farsi alphabets and their pronunciations, from top right to the bottom left

عیب رندان مکن ای زاهد پاکیزه سرشت
که گناه دگران بر تو نخواهند نوشت

من اگر نیکم و گر بد تو برو خود را باش
هر کسی آن درود آخر کار که کشت

Fig. 2 Farsi Poems

Hesam Izakian, is an MSC student at the department of computer engineering, faculty of engineering, university of Isfahan, Iran (e-mail: izakian@comp.ui.ac.ir).

S. Amirhassan Monadjemi, is an assistant professor at the department of computer engineering, faculty of engineering, university of Isfahan, Iran (e-mail: monadjemi@eng.ui.ac.ir).

Behrouz Tork Ladani, is an assistant professor at the department of computer engineering, faculty of engineering, university of Isfahan, Iran (e-mail: ladani@eng.ui.ac.ir).

Kamran Zamanifar, is an assistant professor at the department of computer engineering, faculty of engineering, university of Isfahan, Iran (e-mail: zamanifar@eng.ui.ac.ir).

Characters	MS-Windows standard fonts
ی چ ش ع گ م	Nazanin
ی چ ش ع گ م	Titr
ی چ ش ع گ م	Tahoma

Fig. 3 Some of the Farsi characters written with different fonts

Like other applications of pattern recognition, the must difficult and basic point in OCR systems is extraction of appropriate and robust features. Features should contain information required to distinguish between classes, be insensitive to irrelevant variability in the input, and also be limited in number to permit efficient computation of discrinant functions and to limit the amount of training data required [11]. So far, various methods have been employed to identify Farsi and Arabic characters. In [12], movement on edges has been employed to extract features of characters. In [13], hidden Markov model has been employed to extract features of characters and finally identify them. Some others [15] use morphology to identify characters. In [14], firstly, characters have normalized form the viewpoint of size, and then the umber of pixels in rows and columns have been used as features of characters identify them. In this paper, chain code has been employed as a crucial feature and the number of holes and number and location of auxiliary parts of characters have been employed as added specifications. Meanwhile, this procedure is quite independent regarding size of characters and there is no need to normalize the size of characters. This paper is structured as follows: employed preprocessing will be described in Section II. Then we will explain the procedure of feature extraction in Section III. Identification of characters will be discussed in Section IV and finally the paper will be concluded in Section V.

II. Pre-processing

Pre-processing covers all those functions carried out prior to the features extraction to produce a cleaned up version of the original image so that it can be used directly and efficiently by the feature extraction components of the OCR [10]. Here the aim of the preprocessing is mainly thinning of the character image to the thickness of one point. In this study, the thinning method presented in [17] has been employed. Fig. 4 shows the specimens of images following a complete thinning procedure.

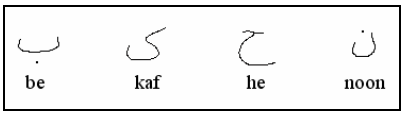

Fig. 4 Some Farsi characters after thinning.

III. Feature Extraction

Here, the number and locations of auxiliary parts, the number of holes and a chain code have been extracted as the features of each character and later have been employed in characters classification. Each of these features and their extraction procedure has been explained below. Possible Auxiliary sections of Farsi characters have shown in Fig. 5, otherwise, the character body will be one-parted (see Fig. 1 too).

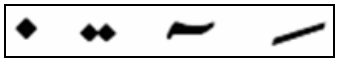

Fig. 5 Possible Auxiliary parts of Farsi characters

A. Number and Locations of Auxiliary Parts

The body of characters (the largest object present in the image) will be considered as the main object in the images and other objects existing will be assumed as auxiliary parts. This feature will be considered as an ordered triple F: (P1, P2, P3). We move from top of the image, row by row, and assign P1 to the first object in the image. P2 and P3 are also allocated to objects located in the lower parts of the character image. Each of the P1, P2 and P3 can have an integer value between 0 and 3. If the intended object is the main body of the character, the value 3 will be assigned to relevant P_i. If the intended object is double dots or, for example the oblique line on 'gaf', or if it bears the sign '~', the value 2 will be assigned to its P_i. If the intended object is a singe dot, 1 will be assigned to its P_i. Fig. 6 shows some samples of images along with their relevant ordered triple codes.

پ	ث	ب	ف	ت
3 2 1	1 2 3	3 1 0	1 3 0	2 3 0
چ	گ	س	ر	ن
3 2 1	2 3 0	3 0 0	3 0 0	1 3 0

Fig. 6 Ordered triple code of some Farsi characters

B. Analysis of the Holes in the Character Images

The next feature we use is the number of holes existing in the character. A character can have a maximum of one hole in Farsi (and also Arabic) scripts. For example the characters 'be', 're', 'dal' have got no holes, whilst the characters 'sad', 'fe', 'ghein' have got one hole. There are different suitable methods for computing the number of holes in the characters. To find out the number of holes of each character we have employed the procedure described in [18]. Placing two features extracted from the images along side each other, the feature vector of each character will be as fallow:

$$F = [P1, P2, P3, Holes] \quad (1)$$

Next, using the feature vector F, the total set of Farsi characters could be separated to ten non-overlapping subsets which are shown in Table I along with their computed feature vectors.

TABLE I
FARSI CHARACTERS PARTITIONED INTO 10 SUBSETS

	F=(P1,P2,P3,Holes)	Character
1	F=(3,0,0,0)	ع ا س د ح ک ل ر ی
2	F=(3,0,0,1)	ه م ص ط و
3	F=(3,1,0,0)	ب ج
4	F=(1,2,3,0)	ژ ث ش
5	F=(3,2,1,0)	پ چ
6	F=(2,3,0,0)	آ گ ت
7	F=(2,3,0,1)	ق
8	F=(1,3,0,0)	غ خ ذ ز ن
9	F=(1,3,0,1)	ف ض
10	F=(3,1,0,1)	ظ

C. The Chain Code Feature

We here introduce a novel chain code feature for Farsi characters classification. In order to obtain the chain code, we just focus on the main part (body) of the character image. From the top of the image, we move row by row to the bottom and consider the first pixel of the body of image which exactly has got one neighbor, as the start point of the chain code. If a character has no starting point, we will consider its chain code as zero (for example, see the character 'ha' in Fig. 1).

Each pixel of image has got eight neighbors; to each neighbor we assign one value between 1 and 8 as Fig. 7 shows.

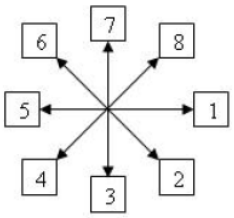

Fig. 7 Neighbors of a pixel and value assign to it

After finding the start point of the chain code in a given character image, we move to the next neighbor pixel which also be a part of the body. Again in the cases of having two or more neighbor pixels with the above condition use the directional priority as shown in Fig. 7. While passing one pixel to its neighboring pixel, we insert the number related to that neighbor in the chain code of that image. After obtaining the chain code for all characters, we realize that the chain code for different characters has different length and length of each chain code depends on the size of the desired character. In more ever, length of the chain codes is usually high; therefore one should normalize this chain code as its length will be fixed and limited, (e.g. 10 digits).

D. Chain Code Normalization

In order to normalize the obtained chain code, we transform it to a two dimensional matrix where in the first row, the value of the chain code, and in the second row, frequency of occurrence of that value are written. For example, if the chain code of a given character is:

$$77773111222225833533 33 \quad (2)$$

then it can be converted into the following form of a 2 × 9 matrix:

$$\begin{matrix} 7 & 3 & 1 & 2 & 5 & 8 & 3 & 5 & 3 \\ 4 & 1 & 3 & 5 & 1 & 1 & 2 & 1 & 4 \end{matrix} \quad (3)$$

now, we are going to omit non useful and redundant details in the chain code. In order to do that, firstly we omit all values which their frequency is 1. For instance, in the above example, the chain code will be reduced to:

$$\begin{matrix} 7 & 1 & 2 & 3 \\ 4 & 3 & 5 & 6 \end{matrix} \quad (4)$$

the process of omitting the less-frequent digits can be continued. For instance in our test, the frequencies less than or equal to five were deleted. Again in the resulted chain code the frequency of each remained digit is summed. Then to transform the chain code matrix to a normalized chain code with length of 10, the relative frequency of each digit is computed using:

$$F_i^n = \frac{F_i}{\sum F_i} \times 10 \quad (5)$$

Where F_i^n is the normalized frequency and F_i is the frequency of each digit in the chain code respectively. In the above example we will obtain:

$$\begin{matrix} 7 & 1 & 2 & 3 \\ 2.22 & 1.66 & 2.77 & 3.33 \end{matrix} \quad (6)$$

then the normalized frequency would be rounded and concatenated to generate the length=10 chain code:

$$7711222333 \quad (7)$$

In some exceptions when the length of the chain code is less than 10, the first or last digits would repeat.

Fig. 8 illustrates some Farsi characters and their corresponding chain codes.

	3 4 4 5 5 5 5 5 6 8
	3 3 3 3 4 5 5 6 7 7
	4 2 8 4 4 3 2 2 1 1

Fig. 8 Chain codes for some example Farsi characters

Following the computation of the chain code, the feature vector will be as follow:

$$F = [P1, P2, P3, Holes, Chain\_Code] \quad (8)$$

IV. Experimental Results

To evaluate the extracted feature vector we need some classification experiments and a classifier. Although there are some advanced classifiers that might be used in OCR systems, e.g. artificial neural networks[21], and support vector machines[22], to decrease the role of the classifier and keep the importance of the feature extraction stage, a simple 1-nearest neighbor classifier[22] was employed in this study.

Matlab was used for code development. In the arranged experiment, each time we trained our classifier with a single font, and measured its recognition performance using another font as the test set. Table II shows the experimental results, where after six tests, average recognition accuracy of 97.4 % is obtained. Even in two cases, the correct recognition rate of 100% is achieved.

TABLE II
Results Obtained from Implementation of the Employed Method

Reference set font	Test set font	Recognition accuracy
Nazanin	Titr	100 %
Nazanin	Tahoma	96.9 %
Titr	Nazanin	100 %
Titr	Tahoma	96.9 %
Tahoma	Nazanin	93.8 %
Tahoma	Titr	96.9 %
Average		***97.4%***

V. Conclusion

A chain code-based approach for identification of Farsi /Arabic characters was introduced in this paper. After thinning the characters, a feature vector consist of different attributes of each Farsi characters, was extracted. The most efficient part of the character feature vector is the novel chain code which was computed using a neighborhood function and a series of normalization. Experimental results using a few standard Farsi fonts shows that the accuracy of the proposed method is 97.4% correct characters identification in average. In the future work, we will try to complete our work for cursive and handwritten Farsi/Arabic character recognition.

References

[1] M. M. Altuwaijri and M. A.Bayoumi, “Arabic text recognition using neural networks”, IEEE International Symposium on Circuits and Systems, pp;415-418, 1994.
[2] B.M.F.Bushofa and M.Spann, “Segmentation and recognition of Arabic characters by structural classification”,Image and Vision Computing, , 15,pp:167-179, 1997.
[3] B. Al-Badr and S. A. Mahmoud, “Survey and bibliography of Arabic optical text recognition”, Signal Processing, 41, pp:49-77, 1995.
[4] L. Zheng, Abbas H. Hassin and X.Tang, “A new algorithm for machine printed Arabic character segmentation”, Pattern Recognition Characters, 25(15),pp:1723-1729, 2004.
[5] J. Mantas, “An Overview of Character Recognition Methodologies”, Pattern Recognition 19, pp. 425-430, 1986.
[6] R. M. Bozinovic and S. N. Shihari, “Off Line Cursive Script Word Recognition”, IEEE Trans.Pattern Anal. Mach. Intell. PAMI 11, pp. 68-83, 1989.
[7] R. Casey and G. Nagy, “Automatic Reading Machine”, IEE Trans. Comput. 17, pp. 492-503, 1968.
[8] K. Y. Wang, R. C. Casey and F. M. Wahl, “Document Analysis System”, IBM J.Res.Dev . 26, pp. 647-656, 1982.
[9] S. Mori, C. Y. Suen and K. Yamamoto, “Historical Review of OCR Research and Development”, Proc. IEEE 80, pp. 1029-1058, 1992.
[10] N. B. Amor, N. E. BenAmara, “Multifont Arabic Characters Recognition Using Hough Transform and HMM/ANN Classification”, journal of multimedia, VOL. 1, NO. 2, MAY 2006.
[11] Lippmann “Pattern Classification using Neural Networks.” IEEE Communications Magazine, 1989.
[12] Kavianifar M. and Amin A. “Preprocessing and structural feature extraction for a multi-fonts Arabic/Persian OCR”, Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR '99 (Cat. No.PR00318). Soc, Los Alamitos, CA, USA. pp: 213-16, 1999.
[13] S. Alma'adeed, C. Higgens, D. Elliman, R. Kasturi, D. Laurendeau, and C. Suen, “Recognition of off-line handwritten Arabic words using hidden Markov model approach” , Proceedings 16th International Conference on Pattern Recognition.IEEE Comput. Soc, Los Alamitos, CA, USA. Vol.3: 481-4, 2002.
[14] J. Cowell, F. Hussain, M. H. Hamza, and M. Sarfraz, “Extracting features from Arabic characters”, Proceedings of the IASTED erence Computer Graphics and Imaging. ACTA Press, Anaheim, CA, USA, pp: 201-6, 2001.
[15] T. Sari, and M. Sellami, “MOrpho-LEXical analysis for correcting OCR-generated Arabic words” , Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition. IEEE Comput. Soc, Los Alamitos, CA, USA. pp: 461-6, 2002.
[16] S. Hoque, K. Sirlantzis, M. C. Fairhurst, “A New Chain-code Quantization Approach Enabling High Performance Handwriting Recognition based on Multi-Classifier Schemes”, Proceedings of the Seventh International Conference on Document Analysis and Recognition , ICDAR 2003.
[17] T. Y. Zhang, C. Y. Suen, “A fast parallel algorithm for thinning digital patterns”, ACM 1984.
[18] Pratt, K. William, “Digital Image Processing”, New York, John Wiley & Sons, Inc., p. 633, 1991.
[19] P. Zingaretti, M. Casparroni and L. Vecci, “Fast chain coding of region boundarie”, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (4), 407-415, 1998.
[20] Christopher J.C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition”, Data Mining and Knowledge Discovery, Vol.2, No.2, 1998.
[21] L. Almeida, “Multilayer Perceptrons”, Handbook of Neural Computation, IOP Publishing Ltd and Oxford University Press, pp. C1.2: 1-C1.2: 30, 1997.
[22] T. Mitchell, “Machine Learning”, McGraw Hill, New York, 1997.