Fine Classification of Unconstrained Handwritten Persian/Arabic Numerals by Removing Confusion amongst Similar Classes

Alireza Alaei[1], P. Nagabhushan[2] and Umapada Pal[3]
[1,2]*Department of Studies in Computer Science, University of Mysore, Mysore, 570 006, India*
[3]*Computer Vision and Pattern Recognition Unit, Indian Statistical Institute, Kolkata-108, India*
[1]*alireza20alaei@yahoo.com,*[2]*pnagabhushan@hotmail.com,*[3]*umapada@isical.ac.in*

Abstract

In this paper, we propose two types of feature sets based on modified chain-code direction frequencies in the contour pixels of input image and modified transition features (horizontally and vertically). A multi-level support vector machine (SVM) is proposed as classifier to recognize Persian isolated digits. In first level, we combine similar shaped numerals into a single group and as result; we obtain 7 classes instead of 10 classes. We compute 196-dimension chain-code direction frequencies as features to discriminate 7 classes. In the second level, classes containing more than one numeral because of high resemblance in their shapes are considered. We use modified transition features (horizontally and vertically) for discriminating between two overlapping classes (0 and 1). To separate another overlapping group containing three numerals 2, 3 and 4 we first eliminate common parts of these digits (tail) and then compute chain code features. We employ SVM classifier for the classification and evaluate our scheme on 80,000 handwritten samples of Persian numerals [10]. Using 60,000 samples for training, we tested our scheme on other 20,000 samples and obtained 99.02% accuracy.

Keywords: Persian Numeral Recognition, Chain Code, Handwritten Character Recognition, SVM.

1. Introduction

The literature details many high accuracy recognition systems for isolated handwritten numerals in some languages (English, Chinese, and Japanese) [6]. However, systems for Persian handwritten numeral/character recognition have not evolved compared to the developments that have taken place with other languages. The reasons for this are cursiveness of handwritten in Persian and multiple forms of each character with respect to its position in words.

Persian numerals mostly dominate in Iran and in some of its neighboring countries. Persian script like other scripts has its own symbols representing 10 numerals. Persian and Arabic numerals are almost the same; but there are some differences between handwritten samples of them [15]. In Persian, some digits (e.g. 0, 2, 4, 5 and 6) are written in two styles. Therefore, the number of classes that exists is 15 classes. However, we reduced the classes to 10 classes and these characteristics make the recognition of Persian numerals more complicated than in other languages. Samples of printed and handwritten Persian digits are shown in Fig.1 and Fig. 2 respectively.

0	1	2	3	4	5	6	7	8	9
۰,٥	۱	٢,۲	۳	٤,۴	٥,۵	٦,۶	۷	۸	۹

Fig.1. Printed samples of Persian digits

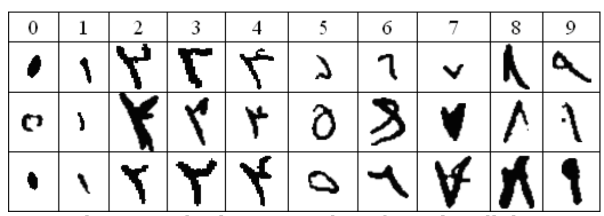


Fig.2. Handwritten samples of Persian digits

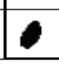
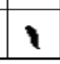
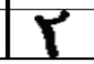
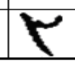
Researchers have used several methods for feature extraction and classification of Persian/Arabic numerals. Many features based on segmentation and shadow code [1], modified contour chain code [2], fractal code [13], profiles [3, 15], moment [5], template [8], structural features (points, primitives) [14], and wavelet [9, 12] have been extracted and used in the literature. For classification, different types of Neural Networks [1, 5, 8, 12, 13], SVM's [2, 3, 9, 15] and Nearest Neighbor [14] have been employed.

From the literature survey of the existing pieces of works on Persian/Arabic numerals recognition, it is evident that not much effort is extended to remove the confusion and ambiguity between similar classes and identify a more efficient feature set as well. To overcome such problems, we introduce a feature set based on transition and propose to find out another

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.181

more effective feature set based on contour chain code in each window map [2, 7]. These types of feature sets, which express the physical shape of input image, provided very good results in [2, 7]. To solve the problem of confusion amongst similar classes (Table 1), at first we consider the ten classes' problem {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} as seven classes {(0, 1), (2, 3, 4), 5, 6, 7, 8, 9} by merging similar shape classes into a single group. We employ SVM for classification purpose, because SVM is reported to be one of the best classifiers for numeral recognition [3, 9, 15]. In the next level, we eliminate common part of similar shape classes from their training and test samples. Then we compute directional frequencies in contour pixels for the group {2, 3, 4} to identify these three digits. For the group {0, 1} we compute transition features as well. Two separate SVMs are used for classification of numerals of these two groups. The scheme of our two-stage recognition system is shown in Fig. 3.

Table 1. Similar samples of Persian digits

Numerals	0	1	2	3	4
Similar Shapes					

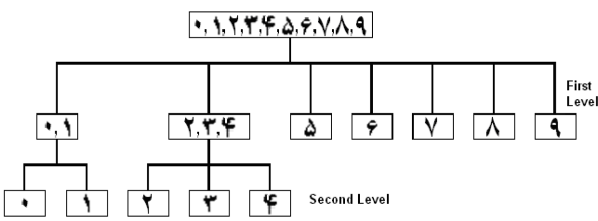


Fig.3. Two levels of proposed Persian numerals classification system

The organization of rest of the paper is as follows: In Section 2 we illustrate feature extraction techniques and Section 3 describes classification. Experimental results and comparisons are described in Section 4. Finally, we present conclusion and future work.

2. Feature extraction

Directional chain code information of the contour points of the input image can be used as features for different purposes like character segmentation, recognition [2, 7], etc. In our system, we compute features based on chain-code directional frequencies of contour pixels of the images as follows: First, we find the bounding box of each input image. Then for better result and independency of features to size and position (invariant to scale and translation), we convert each image (located in bounding box) to a normal size of 49×49 pixels (Fig. 4). This normalized value is decided from the experiment. We extract the contour of the normalized image (Fig. 5). The image contour is scanned horizontally by keeping a window map of size 7×7 on the image from the top left most point to down right most point (49 non-overlapped blocks). For each block the chain code frequencies for all 8 directions are computed (Fig. 6). Instead of expressing the features in terms of 8 directions, we propose to simplify the features into 4 sets corresponding to 4 directions (Fig. 7): i) Horizontal direction code (direction 0 and 4), ii) Vertical direction code (direction 2 and 6), iii) Diagonal direction code (direction 1 and 5) and iv) Off diagonal direction code (direction 3 and 7). Thus, in each block, we get four values (features) representing the frequencies of these four directions. As result, in each image and we compute $49 \times 4 = 196$ features.

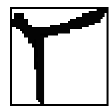


Fig.4. Bounding box of a normalized image

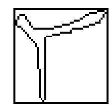


Fig.5. Digit '2' in Persian and its contour

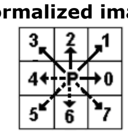


Fig.6. Point P and its 8-direction codes

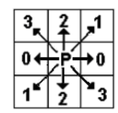


Fig.7. Four directions obtained from 8 directions

In the first level of classification in the proposed system, we use chain-code directional frequencies of contour pixels of the images, as described earlier.

In the second level of classification, to separate combined classes a 0-1 transition feature set (vertical and horizontal) [11] in addition to chain-code directional frequencies is proposed.

To extract vertical transitions we scan the image column-wise (top to bottom) starting from left and check for the occurrence of transition from background to foreground and vise versa. We count it and this process is repeated for the entire image. For horizontal transitions we scan the image row-wise (left to right) from top to bottom and the remaining procedure is same as above. The values of horizontal and vertical transitions after normalization (dividing them with the area of the bounding box) constitute transition feature set. In Persian digit 1, value of horizontally transition feature will be more when compared to 0 and value of vertically transition for 0 (zero) will be more in most of the time when compared to 1 (Fig. 8).

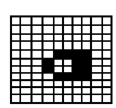
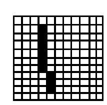
 V=14,H=12  V=4, H=18

Fig.8. Digits 0 and 1 in Persian and their vertical and horizontal transition features before normalization. V and H are number of vertical and horizontal transitions

For combined class {2, 3, 4}, which contains numerals with very similar shapes (Table 1), a new technique is proposed to separate and solve the problem of confusion among them. By looking at the shapes of these digits, it is found that lower parts (tails) of these digits are almost the same and only upper parts (heads) of these digits differ. By this idea, the common part (tail) of each digit is automatically removed (3rd column of Fig. 9). The procedure of removing the tail is as follow:

Step 1: Connect 2 components if there is only 1 pixel distance between them (bridge connectivity).

Step 2: Use component analysis and remove small components that have less than 9 pixels area.

Step 3: Find background profiles from right and left. (2nd column of Fig. 9).

Step 4: Scan the input image from bottom and eliminate the foreground (tail) row-by-row if the distance between the left and right profiles is less than 1.5*average stroke width (stroke width of a component is the statistical mode of black runs obtained by scanning a component vertically as well as horizontally). 4th column from left of Fig. 9 shows the common part that eliminated from the image respectively.

The parameters of 9 pixels for the area and 1.5*average for the width are fixed using some experimental and statistical analysis. From our statistical study on 80000 samples of Persian numerals [10], 1121 samples had more than 2 components and out of these 1121 samples, 630 samples had at least one component less than 9 pixels. To remove these components we use 9 pixels as threshold. By removing those components that disturb proper normalization (They are far away from main components and look like noise), we get right normalized samples for input samples (Fig.10).

After removing common part from the input image, we normalize and found the contour of the modified image. (5th column of Fig. 9 from left). Then we find the features based on chain-code directional frequency, which we describe at the beginning of this section. For each image, we compute 196 features and then classify them. From Fig. 9, it may be noted that by removing common parts and normalizing the remaining of the numerals, features that are more informative can be obtained.

To test reliability of proposed algorithm for common part elimination, we randomly chose 900 samples from classes 2, 3 and 4. The respective algorithm was executed to remove common part and the result was examined manually. The concluding result showed well performance on more than 98% of the samples and on the rest, the algorithm has not completely removed common part.

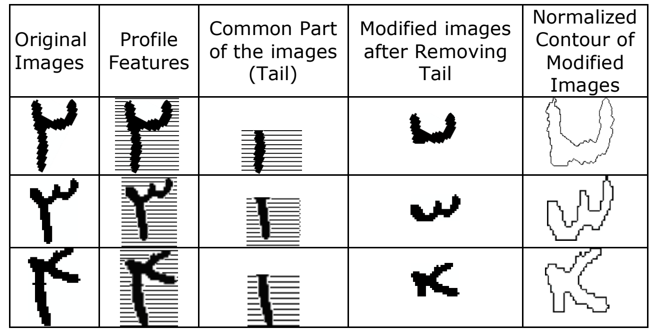


Fig.9. Preparation of images of classes 2, 3 & 4 for feature extraction

Fig. 10. Bounding box: (a) before and (b) after removing disturbance component

3. Classification

SVM has been considered in the literature as one of the powerful classifiers for character and numeral recognition [2, 3, 9, 15]. We use SVM for recognition purpose. The SVM has been defined for two-class problem and it looks for the optimal hyper-plane, which maximized the distance margin between the nearest examples of both classes, named Support Vector. The linear SVM can be extended to a non-linear classifier by using kernel functions like polynomial/Gaussian kernels. We have tested linear, Gaussian and polynomial kernels during our experiments and we received the best result using Gaussian kernel. Details of SVM can be found elsewhere [4].

The proposed architecture of two levels SVM based classifier is included seven one-against-all SVMs for the first level of recognition. The input feature set is the directional features (196 features). In the second level of our classifier, we consider two separate one-against-all SVMs to separate the combined {0, 1} class. The feature set is two 0-1 transition features. For the recognition of combined {2, 3, 4} class we also utilize another three one-against-all SVMs and the feature set is directional features which extract from the modified input images (without tails). All the SVMs train with the respected training feature sets and the results explore by using separate test feature sets.

4. Results and comparative analysis

For experimental results, we considered 6,000 samples per class for training and 2,000 samples per class for testing from a standard Persian numeral dataset [10]. Further, we utilized the dataset, which was used in [3]. Experimental results are described in next subsections.

4.1. Performance of the proposed system

Using 60,000 samples for training, we tested our scheme on other 20,000 samples and obtained 99.02%, which showed an improvement of 0.31% when compared with the work presented in [2]. From the experiment, we got an accuracy of 99.99% when the training and testing samples were the same. Experimental result with dataset in [3] showed correct recognition rate of 99.04%.

4.2. Confusion pairs

Initially by considering 10-class problem, we observed confusion between classes in the recognition phase. The major confusions were amongst classes 2, 3 and 4, which were shown in Table 2. This happened because 2, 3 and 4 look very similar in shape. Some other confusion occurred among classes 0 and 1 too. This was because of very small length of some samples in class 1 and somehow long length of samples from class 0. From the Table 2 it may be noted that out of 2000 samples of numeral three 59 (2.95%) and 24 (1.2%) samples misrecognized as numeral 2 and 4 respectively. Based on this result we grouped similar shape confusion numerals for their special treatment. By using two levels, SVM based classifier; we obtained accuracy of 99.52% for 7 classes of Persian numerals in the first level of classification. Related confusion matrix is shown in Table 3. The Table 4 and 5 show the confusion matrices of the similar shape groups at their 2nd level of classification. By combining the Tables 3, 4 and 5 we obtained Table 6, which shows overall confusion matrix for ten classes of Persian numerals. By comparing confusion matrices of one level classifier (Table 2) and cascaded two levels classifier (Table 6), it may be noted that the number of confusions amongst classes 0 and 1 reduced from 44(32+12) to 19(10+9) and number of confusions amongst classes 2, 3 and 4 decreased from 117 to 80. There were still a large number of confusions (46) between classes 2 and 3. The confusions were because of missing of indent in some samples of class 3.

4.3. Erroneous samples

Some misclassified samples are shown in Table 7. This table clearly shows that a part of the errors has resulted because of very poor quality samples and different writing styles. In our experiment, we found that in some samples, there were two, three and sometimes more broken parts, which made recognition task more difficult.

4.4. Comparison of results

To compare the performance of our method we noted the performances of most of the works available for Persian numerals. See Table 8 for comparison details. It may be noted from this Table that all the existing works except [2] were evaluated on smaller datasets. The highest accuracy (99.57) was obtained from the work due to Soltanzadeh et al. [3] but they

Table 2. Confusion matrix of 10-class numeral

	0	1	2	3	4	5	6	7	8	9
0	1956	32	0	0	0	9	0	3	0	0
1	12	1986	0	0	1	0	0	0	0	1
2	0	1	1981	10	4	0	1	1	0	2
3	0	1	59	1914	24	2	0	0	0	0
4	0	0	7	13	1979	1	0	0	0	0
5	4	0	0	0	5	1987	0	0	3	1
6	2	8	1	0	2	2	1974	0	0	11
7	0	4	3	1	0	0	0	1992	0	0
8	0	1	0	0	0	0	0	0	1997	2
9	3	9	0	0	0	3	8	0	1	1976

Table 3. Confusion matrix of the result for the first level of classification (7 classes)

	0,1	2,3,4	5	6	7	8	9
0, 1	3986	1	9	0	3	0	1
2,3,4	2	5991	3	1	1	0	2
5	4	5	1987	0	0	3	1
6	10	3	2	1974	0	0	11
7	4	4	0	0	1992	0	0
8	1	0	0	0	0	1997	2
9	12	0	3	8	0	1	1976

Table 4. Confusion matrix of the second level of classification for the classes of 0 and 1

	0	1
0	1978	10
1	9	1989

Table 5. Confusion matrix of the second level of classification amongst classes of 2, 3 and 4

	2	3	4
2	1976	13	6
3	33	1945	19
4	4	5	1990

Table 6. Confusion matrix of the result after second level of classification (10 classes)

	0	1	2	3	4	5	6	7	8	9
0	1978	10	0	0	0	9	0	3	0	0
1	9	1989	0	0	1	0	0	0	0	1
2	0	1	1976	13	6	0	1	1	0	2
3	0	1	33	1945	19	2	0	0	0	0
4	0	0	4	5	1990	1	0	0	0	0
5	4	0	0	0	5	1987	0	0	3	1
6	2	8	1	0	2	2	1974	0	0	11
7	0	4	3	1	0	0	0	1992	0	0
8	0	1	0	0	0	0	0	0	1997	2
9	3	9	0	0	0	3	8	0	1	1976

have used 257 features. We considered at most 196 features (in 2nd level only 2 features for discriminating between classes 0 and 1 were used) in the proposed system and we obtained 99.02% accuracy with the very large dataset [10].

Table 7. Some errors of the proposed system

Class	0	1	2	3	4	5	6	8	9

Samples / Recognized class / Correct class

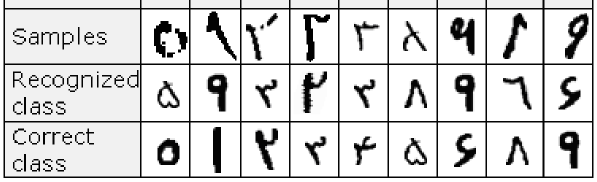


Table 8. Comparison of different algorithms

Algorithms	Dataset size		Accuracy (%)	
	Train	Test	Train	Test
Shirali-shahreza et al.[1]	2600	1300	-	97.80
A. Alaei et al. [2]	60000	20000	99.99	98.71
Soltanzadeh &Rahmati[3]	4974	3939	-	99.57
Dehghan & Faez [5]	6000	4000	-	97.01
Ziaratban et al. [8]	6000	4000	100	97.65
Mowlaei & Faez [9]	2240	1600	100	92.44
Mowlaei et al. [12]	2240	1600	99.29	91.88
Mozaffari et al. [13]	2240	1600	98.00	91.37
Mozaffari et al. [14]	2240	1600	100	94.44
Sadri et al. [15]	7390	3035	-	94.14
Proposed system	60000	20000	**99.99**	**99.02**

5. Conclusion

In this paper, two efficient feature extraction techniques and a cascaded SVM based classifier are proposed. From experimental results, it is evident that our system resulted good performance (99.02%). We noted that most of misclassified samples were from classes of 2, 3 and 4, which are similar in shapes. The recognition of such similar numerals is sometimes difficult even for human being. It is obvious that to improve the performance of proposed system further, we need to investigate more on sources of errors. In future, we plan to use some preprocessing techniques like slant detection/correction which may remove some of confusions amongst similar classes. Moreover, we are going to use some dimensionality reduction techniques as well as applying rejection criteria, which may lead us to better results.

Acknowledgment

Authors would like to thank Mr. S. Chanda for his help in this work.

References

[1] M. H. Shirali-Shahreza, K. Faez and A. Khotanzad, "Recognition of Hand-written Persian/Arabic Numerals by Shadow Coding and an Edited Probabilistic Neural Network", Proceedings of International Conference on Image Processing, Vol. 3, 1995, pp. 436-439.

[2] Alireza Alaei, Umapada Pal and P. Nagabhushan, "Using Modified Contour Features and SVM Based Classifier for the Recognition of Persian/Arabic Handwritten Numerals", Proceedings of 7th International Conference on Advances in Pattern Recognition, 2009, pp.391-394.

[3] H. Soltanzadeh and M. Rahmati, "Recognition of Persian handwritten digits using image profiles of multiple orientations", *Pattern Recognition Letters* 25, 2004, pp.1569–1576.

[4] V. Vapnik, "The Nature of Statistical Learning Theory", *Springer Verlang*, 1995.

[5] M. Dehghan and K. Faez, "Farsi Handwritten Character Recognition With Moment Invariants", Proceedings of 13th International Conference on Digital Signal Processing, Vol. 2, 1997, pp. 507-510.

[6] S. N. Srihari and G. Ball, "An Assessment of Arabic Handwriting Recognition Technology", CEDAR Technical Report TR-03-07, 2007.

[7] F. Kimura, T. Wakabayashi, S.Tsuruoka and Y. Miyake, "Improvement of handwritten Japanese character recognition using weighted direction code histogram", *Pattern recognition*, vol.30, no.8, 1997, pp.1329-1337.

[8] M. Ziaratban, K. Faez and F. Faradji, "Language-Based Feature Extraction Using Template-Matching in Farsi/Arabic Handwritten Numeral Recognition", Proceedings of 9th ICDAR, Vol.1, 2007, pp. 297-301.

[9] A. Mowlaei and K. Faez, "Recognition of Isolated Handwritten Persian/Arabic Characters and Numerals Using Support Vector Machines", Proceedings of XIII Workshop on Neural Networks for Signal Processing, 2003, pp. 547-554.

[10] H. Khosravi, E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on the variety of handwriting styles", *Pattern Recognition Letters* Vol.28, Issue 10, 2007, pp. 1133-1141.

[11]P.D. Gader, M.Moharned and I-H. Chiang, "Handwritten Word Recognition with Character and Inter-Character Neural Networks", IEEE Transactions on Systems, Man. and Cybernetics, Vol. 27, Issue 1, 1997, pp. 158-164.

[12] A. Mowlaei, K. Faez and A. Haghighat, "Feature Extraction with Wavelet Transform for Recognition of Isolated Handwritten Farsi/Arabic Characters and Numerals", D.S. P. Vol. 2, 2002, pp. 923- 926.

[13] S. Mozaffari, K. Faez and H. Rashidy Kanan, "Recognition of Isolated Handwritten Farsi/Arabic Alphanumeric Using Fractal Codes", Image Analysis and Interpretation I6th Southwest Symposium, 2004, pp. 104-108.

[14] S. Mozaffari, K. Faez and M. Ziaratban, "Structural Decomposition and Statistical Description of Farsi/ Arabic Handwritten Numeric Characters", Proceedings of the 8th ICDAR, Vol. 1, 2005, pp. 237- 241.

[15] J. Sadri, C. Y. Suen and T. D. Bui, "Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits", Proceedings of the 2nd Conference on Machine Vision and Image Processing & Applications, Vol. 1, 2003, pp. 300-307.