Cluster Based Weighted SVM for the Recognition of Farsi Handwritten Digits

Mehdi Salehpour, and Alireza Behrad

***Abstract*—The recognition of handwritten characters and digits is an important and challenging issue in OCR algorithms. This article presents a new method in which cluster based weighted support vector machine is used for the classification and recognition of Farsi handwritten digits that is reasonably robust against rotation and scaling. In the proposed algorithm, after applying the necessary preprocessing on the digits images, the required features are extracted using principle component analysis (PCA) and linear discrimination analysis (LDA) algorithms. The extracted features are then classified using a new classification algorithm called cluster based weighted SVM (CBWSVM). We tested the proposed algorithm with a database containing 7600 handwritten digits with and without rotation and the results showed the recognition rate of 96.5% in digits without rotation and 95.6% in digits with rotation of the 15 degrees. The comparison of the results with those of other methods showed the efficiency of the proposed algorithm.**

***Index Terms*—CBWSVM, Clustering, Handwritten digit recognition, PCA, PCA-LDA .**

I. Introduction

TODAY, high volume of available printed and paper documents are converted to digital image documents by the use of scanners or cameras. Storage, retrieval and efficient management of these image documents are very important in many applications, such as office automation and digital libraries. It is sometimes necessary to retrieve required information from such digital documents. Therefore optical character recognition systems have been developed which are techniques for converting text images into computer readable/editable texts. Identifying the handwritten characters is one of the very interesting and challenging issues in OCR algorithms. Among different applications of OCR, the recognition of handwritten digits has many usages including the recognition of digits in post office and bank applications [1].

Different algorithms and features are used for the identification of digits and characters. In [2],[3] combination of classifiers are used for recognition of handwritten Farsi digits. In [4] normalized Gabor filter bank with zero DC response is used for indexing and the retrieval of character images. However the algorithm is sensitive to selection of Gabor filter parameters including bandwidth and orientation.

In [5] one-dimension geometric moments with Bayes classifier is used for the recognition of Arabic characters; However the method is only tested for the recognition of printed documents and the efficiency of the algorithm has not been demonstrated for the recognition of the handwritten characters.

In [6] the recognition of Persian digits is performed by overlaying different normalized images of one class and counting the number of black pixels in different pixel coordinates. The resultant image representing the number of black pixels in different pixel coordinates is used for the classification of test images. However the method is sensitive to the location of digits in the area and outliers. In addition the rotations of the digits give rise to the generation of different patterns than patterns of database and hence erroneous recognition. Neuro-fuzzy algorithm is another method for the recognition of handwritten characters proposed by [7]. However the method needs the intervention of human experts in some parts of the algorithm.

In this article, a new method for the recognition of handwritten digits is introduced that is reasonably robust against the rotation and scaling. We used two different methods for feature extraction as well as two different methods for classification and the results are compared with each other and those of other methods. Experimental results showed the effectiveness of the proposed features and classification algorithms.

The Structure of the paper is as follows. Following this section, the block scheme of the proposed algorithm is reviewed. Section 3 describes the feature extraction algorithms. In Section 4 the methods for classification are explained and finally the experimental results and conclusions are discussed in section 5 and 6 respectively.

II. Block Scheme of the Proposed Algorithm

Fig. 1 shows the block scheme of the proposed algorithm for training and test stages. The algorithm starts by applying required preprocessing to input image. In preprocessing stage the input image is first converted to binary image and using morphology based algorithms noisy points are removed. We then extract the bounding box and center of mass for the handwritten digit and normalize it to a constant size of 40 × 40. Feature extraction is the next stage of the proposed algorithm. We used two different algorithms for feature extraction including PCA and PCA-LDA algorithms. The extracted features are then used for training and testing the classifier algorithm. We utilized Cluster based weighted

Alireza Behrad is with the Faculty of Engineering, Electrical Engineering Department of Shahed University Tehran, Iran. (phone: (+98 - 21) 51212036; e-mail: behrad@shahed.ac.ir).

Mehdi Salehpour is Msc. Student of Faculty of Engineering, Shahed University Tehran, Iran (e-mail: salehpour@shahed.ac.ir).

978-1-4244-8819-3/10/$26.00 ©2010 IEEE

(fuzzy) SVM for the classification and recognition of handwritten Farsi digits.

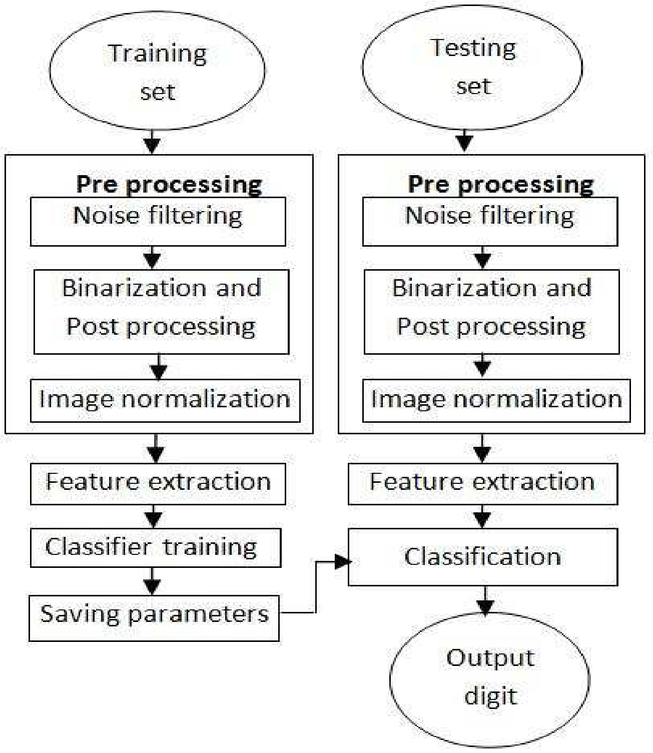

Fig. 1. Block scheme for the proposed algorithm

III. Feature Extraction

We used two different methods for feature extraction including PCA and PCA-LDA algorithms which are explained in the following sections.

A. Feature extraction using PCA method

In this method, all the images in the database are represented as one dimensional vectors X_i after applying required preprocessing. Then covariance matrix (C) is calculated for all training images in database of handwritten Farsi digits using the following equations [8]:

$$\bar{X} = \frac{\sum_{i=1}^{N} X_i}{N} \qquad (1)$$

$$\hat{X}_i = X_i - \bar{X} \qquad (2)$$

$$W = [\hat{X}_1, \hat{X}_2, \ldots, \hat{X}_N] \qquad (3)$$

$$C = \frac{1}{N}\sum_{i=1}^{N} \hat{X}_i \hat{X}_i^T = \frac{1}{N} W W^T \qquad (4)$$

where $\bar{X}$ and C are mean vectors and covariance matrix for digit images in the database and N is the total number of database image.

Then the principle component analysis (PCA) is applied to the covariance matrix C and M largest principle components are used for feature extraction as follows:

$$Y_i = (X_i - \bar{X})^T D \qquad (5)$$

where Y_i are the extracted features and D is the matrix of M principle vectors.

B. Feature extraction using PCA-LDA method

The feature extraction using PCA-LDA algorithm is similar to PCA method. In this method, the covariance matrix is calculated like PCA algorithm; However M_2 largest principle components are used for feature extraction where M_2>M. We then reduce the size of features using the LDA algorithm. The algorithm for size reduction and the extraction of features using the LDA algorithm has the following steps:

- The average of vectors for each digit is calculated using the output of PCA algorithm as follow:

$$\bar{Y}^k = \frac{\sum_{i=1}^{N_k} Y_i^k}{N_k} \qquad k=1,2,\ldots,10 \qquad (6)$$

where N_k is the number of images for digit k in the database, Y_i^k are the output of PCA algorithm for class k and $\bar{Y}^k$ are the mean vector value for class (digit) k.

- The scattering matrix for each class (digit) is calculated using the following equations:

$$\hat{Y}_i^k = Y_i^k - \bar{Y}^k \qquad k=1,2,\ldots,10 \qquad (7)$$

$$S^k = \sum_{i=1}^{N_k} (\hat{Y}_i^k \cdot \hat{Y}_i^{kT}) \qquad k=1,2, ...,10 \qquad (8)$$

$$S_W = \sum_{k=1}^{10} S^k \qquad (9)$$

where S^k are scattering matrix for each class.

- The intra class scattering matrix S_B and matrix A are calculated:

$$S_B = \frac{N}{10}\sum_{k=1}^{10} (\bar{Y}^k - \bar{X})(\bar{Y}^k - \bar{X})^T \qquad (10)$$

$$A = S_w^{-1} S_B \qquad (11)$$

- Eigen value and eigen vectors for matrix A are calculated and M largest eigen values are used to map Y_i vectors to obtain feature vectors [9].

IV. Classification Algorithm

SVM algorithm classifies the input data by mapping the input samples to higher dimensions in which the separation between classes is performed more efficiently according to cover's theorem [3].

Binary SVM classifier is a classifier with two classes ω_1, ω_2 The SVM classifier in this case creates a hyper plane with the following equation to separate these two classes [9]:

$$G(\mathbf{x}) = \omega^T \mathbf{x} + \omega_0 \qquad (12)$$

The classification for the input vector x_i is carried out using the following equation:

$$\begin{cases} \omega^T \mathbf{x}_i + \omega_0 > 0 \Rightarrow \mathbf{x}_i \in \omega_1 \\ \omega^T \mathbf{x}_i + \omega_0 < 0 \Rightarrow \mathbf{x}_i \in \omega_2 \end{cases} \qquad (13)$$

In this article, we utilized weighted or fuzzy SVM algorithm based on K-means clustering. The advantage of this algorithm over SVM classifier is that in the fuzzy mode, the input data is not exclusively assigned to a specified class and its membership function defines its dependency to different classes.

SVM is a binary classifier. In order to use it for the classification of several classes there are two approaches, one against all and one against one [9]

A. *One against all*

In this method, the K binary SVM classifiers are used for classification where K is the number of classes and the separator line for each class separate it from all other classes. Fig. 2 illustrates one against all method. For classification in this method each sample is given to all SVM classifiers and the winner class is selected.

B. *One against one*

In this method, H different binary SVM classifiers are used and each SVM separates class i from class j where i and j are different class indexes [9]. Fig. 3 shows one against one method.

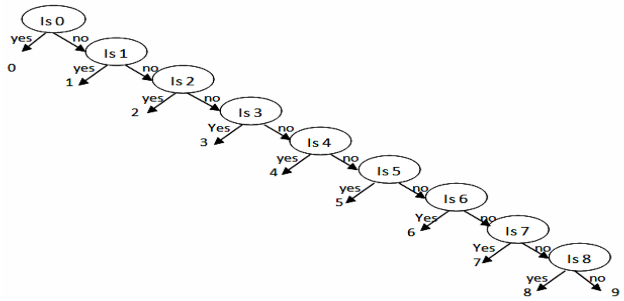

Fig. 2. Ten-class SVM classifier using one against all method

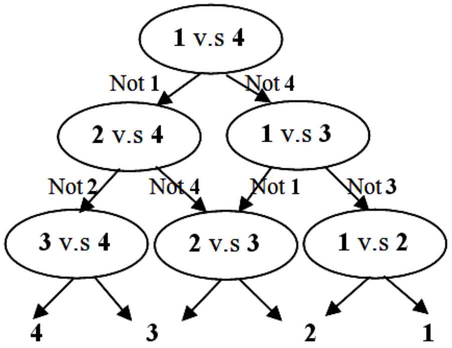

Fig. 3. 4-class SVM classifier using one against one method

C. *Training Process*

Block diagram for training process is given in Fig. 4. For training purpose we first cluster training data to N clusters using k-means clustering algorithm. Then the statistical properties of each cluster including mean vector and covariance matrix are calculated and hyper plane weights.

Then SVM training process is carried out in each cluster individually by using the data of different classes in that cluster. We used weighted SVM algorithm with quadratic kernel function and one against one method for classification in each cluster. Then the average distance between the data of one class in each cluster and the calculated hyper ns of that class is calculated. The output of training process includes parameters such as mean and covariance of feature vectors in each cluster, the average distance of different classes in the clusters to different hyper planes and the weights of hyper planes. These parameters are used in the test process.

D. *Classification*

Classification or test process also uses one against one structure. Fig. 5 shows the block scheme for the classification algorithm. For classification, the cluster weights for input data are calculated first. Cluster weights are Mahalanobis distance between input data and N different clusters which are calculated using the following equation:

$$w_n = \frac{1}{\sqrt{2\pi}\,|\Sigma|^{\frac{1}{2}}} \exp(-(\mathbf{x}_t - \overline{\mathbf{x}}_n)^T \Sigma_n^{-1} (\mathbf{x}_t - \overline{\mathbf{x}}_n)) \tag{14}$$

where Σ_n and $\overline{x}_n$ are covariance matrix and mean vector for n^{th} cluster respectively and w_n is the weight for cluster n.

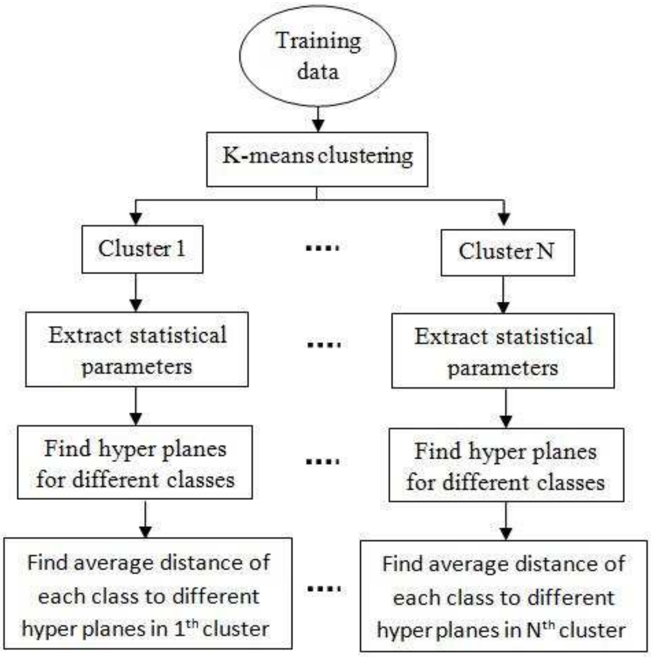
Training data
K-means clustering
Cluster 1
....
Cluster N
Extract statistical parameters
....
Extract statistical parameters
Find hyper planes for different classes
....
Find hyper planes for different classes
Find average distance of each class to different hyper planes in 1th cluster
....
Find average distance of each class to different hyper planes in Nth cluster

Fig. 4. Block diagram of training process using cluster based weighted SVM

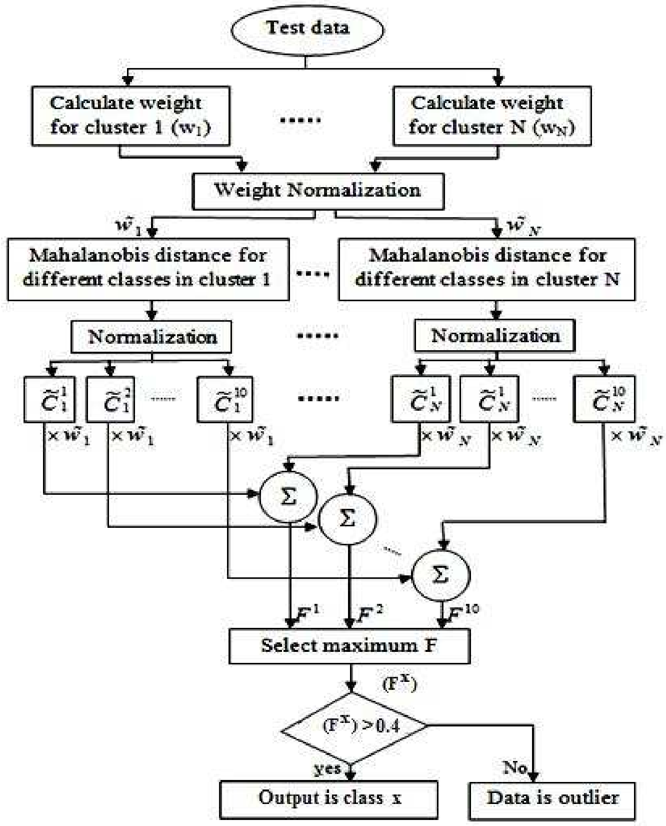

Fig. 5. Block diagram of test process

We then normalize w_n weights to a value between 0 and 1 as follow:

$$\tilde{w}_n = \frac{w_n}{\sum_{k=1}^{N} w_k} \qquad (15)$$

where $\tilde{w}_n$ is the normalized cluster weight for cluster n. We then calculate the Mahalanobis distance between input data and 10 different classes in each cluster. To calculate Mahalanobis distance between input data and class m in a cluster, the Mahalanobis distance between input data and all hyper planes related to class m are calculated in that cluster. Then sum of all calculated distance for class m is called Mahalanobis distance between input data and class m in the cluster n. The calculated Mahalanobis distances between input image and 10 different classes in a cluster is further normalized to a value between 0 and 1 and represented by $\tilde{C}_n^m$ where n and m are cluster and class indexes respectively.

We then calculate membership function of input image for each class using the following equation:

$$F^m = \sum_{n=1}^{N} \tilde{w}_n \tilde{C}_n^m \qquad m=1, 2,..., 10 \qquad (16)$$

where F^m is the membership function of input data for class m. Among 10 resultant membership functions, the maximum value is selected and compared to a threshold value obtained experimentally. If it is less than the threshold value, it is known as outlier data. Otherwise the related class is known as the winner class.

V. Experimental Results

The proposed algorithm implemented using a MATLAB program and tested by Farsi handwritten database [2],[3]. The database includes 8600 samples for ten Farsi handwritten digits, 860 samples for each digit. Fig. 6 shows samples of Farsi handwritten digits in the database.

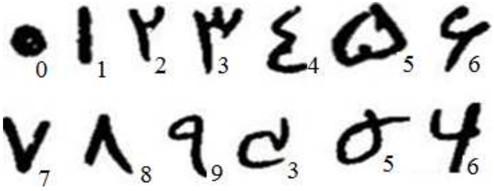

Fig. 6. Samples of Farsi handwritten digits in the database

We used 660 samples of each digit for training and remaining 200 samples for test. We repeated the test and train process several times and calculate the average of results as final results.

To compare the results of proposed algorithm with those of other algorithms, we implemented two methods presented in references [2] and [3] as well. In [2] the combination three Perceptron neural networks with different number of neurons in middle layer are used for digit recognition. The paper use PCA features for recognition. Table 1 shows the results of this algorithm for different number of PCA features and different number of neurons in middle layer. The maximum recognition rate for this algorithm is 80.1%.

The algorithm in reference [3] utilizes 10 two-class MLP classifiers for digit recognition. Table 2 shows the results of this algorithm for different number of PCA features and different number of neurons in middle layer. The maximum recognition rate for this algorithm is 95.4%.

Table 3 illustrates the results of proposed method for different number of features. As it is shown in this table, the result of the proposed algorithm is better than two previous methods.

TABLE I
Recognition Rates for Different Number of PCA Features and Different Number of Neurons in Middle Layers for Algorithm of Reference [2]

PCA features	Number of neurons in middle layer of MLP			
	400	600	700	800
10	60	73.1	75.2	72.5
20	55.4	74.7	**80.1**	77.1
30	65.2	77.2	77	78.2

TABLE II
Recognition Rates for Different Number of PCA Features and Different Number of Neurons in Middle Layers for Algorithm of Reference [3].

PCA features	Number of neurons in middle layer of MLP					
	10	15	20	25	30	35
10	91.3	92.5	93.2	93.5	93.9	94
20	92.2	93.4	94.4	95	**95.4**	95.3
30	92.2	93.4	93.3	93.6	94.5	94.1

TABLE III
Recognition Rates for Different Number of Features Using the Proposed Algorithm

Feature Extraction	Number of features				
	10	20	30	40	50
PCA	91.3	93.4	95.1	94.6	94.2
PCA-LDA	93.1	94.2	**96.5**	95.3	93.7

Some parts of errors of the proposed algorithms are related to images taken with low resolution.

To show the efficiency of proposed algorithm in the presence of digit rotation, we tested the proposed algorithm using images with rotation. For this purpose we rotated 30 percent of test images with different angles. Table 4 shows the results of proposed algorithm and the algorithm of reference [3] for test data with different rotations which are applied randomly clockwise or counter clockwise to test data.

As it is shown in table 4, the proposed algorithm is more robust to digit rotation and PCA-LDA feature extraction algorithm exhibits more stability.

TABLE IV
Recognition Rate for Test Data with Different Rotation.

Algorithm	Rotation angle		
	15	30	45
Algorithm of reference [3]	88.7	85.2	84.6
Proposed algorithm with PCA features	94.2	93.3	92.6
Proposed algorithm with PCA-LDA features	95.6	94.8	93.2

VI. Conclusion

In this article, a new method for the recognition of Farsi handwritten digits was introduced. We used PCA and PCA-

LDA algorithm for feature extraction which PCA-LDA algorithm is more robust against image rotation. We used a new method called cluster based weighted SVM for classification. We tested the proposed algorithm using a database with and without rotation and the results showed the efficiency of the proposed algorithm.

REFERENCES

[1] A. Webb. *Statistical pattern recognition*, 2nd ED., John Wiley & Sons, 2003.

[2] H. N. Karizi, R. Ebrahimpour and E. kabir, "Combination of classifiers for recognition of Farsi handwritten digits*," in Proc. 3rd Iranian Conference on Machine Vision and Image (MVIP2004)*, pp. 115-119, 2004. [Published in Farsi]

[3] M. Nahvi, M. Rafeei, R. Ebrahimpour and E. Kabir, "Combination of two-class classifiers for the recognition of Farsi handwritten digits," *in Proc 16th Iranian Conference on Electrical Engineering ICEE2008*, pp. 203-207, 2008. [Published in Farsi]

[4] B. S. Manjunath, and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837-842, August 1996.

[5] H. Al-Usefi and S. Udpa, "Recognition of Arabic characters," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 853-857, August 1992.

[6] M. Khosravi , and E. Kabir, "Introducing a very large dataset of handwritten Farsi digits and a study on their varieties," *Pattern Recognition. Letter*, vol. 28, no. 10, pp. 1133–1141, February 2007.

[7] P. Bahri, "Farsi handwritten character recognition using Neuro-fuzzy algorithms", MSc. Thesis, Iran University of Science & Technology, Tehran, 1998.

[8] M. Turk, and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, March 1991.

[9] S. Abe, *Support vector machines for pattern recognition*, Springer Verlog London limited, 2005.

[10] I. Lindsay Smith, "A tutorial on principal component analysis" 2002. Online: http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf.

[11] K. Khatatneh ,"Probabilistic artificial neural network for recognizing the Arabic hand written characters," *Journal of Computer Science*, vol. 3, no. 12, pp. 881-886, 2006.

[12] S. Avidan, "Support Vector Tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 1064-1072, 2004.