Application of Support Vector Machines for Recognition of Handwritten Arabic/Persian Digits

Javad Sadri **Ching Y. Suen** **Tien D. Bui**

Centre for Pattern Recognition and Machine Intelligence (CENPARMI)
Department of Computer Science, Concordia University
Suite GM 606, 1455 de Maisonneuve Blvd. West,
Montreal, Quebec, Canada H3G 1M8
Tel : 1-514-848-7950 , Fax : 1-514-848-4522
Email: { j_sadri, suen, bui }@cenparmi.concordia.ca

Abstract: A new method for recognition of isolated handwritten Arabic/Persian digits is presented. This method is based on Support Vector Machines (SVMs), and a new approach of feature extraction. Each digit is considered from four different views, and from each view 16 features are extracted and combined to obtain 64 features. Using these features, multiple SVM classifiers are trained to separate different classes of digits. CENPARMI Indian (Arabic/Persian) handwritten digit database is used for training and testing of SVM classifiers. Based on this database, differences between Arabic and Persian digits in digit recognition are shown. This database provides 7390 samples for training and 3035 samples for testing from the real life samples. Experiments show that the proposed features can provide a very good recognition result using Support Vector Machines at a recognition rate 94.14%, compared with 91.25% obtained by MLP neural network classifier using the same features and test set.

Keywords: Optical Character Recognition (OCR), Feature Extraction, Machine Learning, Support Vector Machine (SVM), Multiple Support Vector Classifiers, MLP Neural Network

1. Introduction

Optical character recognition (OCR) is one of the most successful applications of automatic pattern recognition [1], and handwritten digit recognition has long been an active topic in OCR and classification/learning research. Numerous approaches have been proposed for pre-processing, feature extraction, learning/classification, post-processing, and some standard image databases are widely used to evaluate the performance [2]. However most of these research activities have been done on Latin digits, and little work was done on Arabic/Persian handwritten digit recognition. In the last ten years, a few papers were published on Arabic/Persian digits recognition, and they used different methods for feature extraction such as: geometric moments invariants, Zernike moments[3], shadow coding[4]; and different methods for classification such as: statistical, fuzzy, or neural approaches [3,4]. But since online standard database for Arabic/Persian digit recognition is not available, researchers used their own databases to evaluate these methods. As a result, these methods are not comparable to one other. Therefore, it is very important to introduce a standard database for measuring the performance of these methods. It is also important to apply the state of the art classification techniques and new feature extraction approaches to recognize Arabic/Persian digits and to improve the recognition rate.

Although Support Vector Machine (SVM) as one of the state of the art classification techniques has been used for Latin digit recognition [2], according to our knowledge, it has not been used for Arabic/Persian digits recognition. This paper presents application of SVMs for recognition of isolated handwritten on CENPARMI Indian (Arabic/Persian) digit database [5]. The motivation for this study is to present a new method for feature extraction and compare performances of SVM and MLP neural network classifiers on a new Arabic/Persian handwritten digit database. This database is available for research community upon request to CENPARMI [5]. This paper is organized as follows: first CENPARMI Indian digit database is introduced. Preprocessing and feature extraction methods are described in section 3 and 4 respectively. In section 5 we describe SVM classifiers and their combination. In section 6 we present our results and finally draw our conclusions.

2. The Database

Although Persian, and Arabic digits (so called Hindi or Indian digits [5]) look very similar, we noticed there are important differences between Persian, and Arabic ways of handwriting of digits that are shown in Table 2-1. Persian handwritten digits form normally 13 classes because of two different ways of writing 0,4,and 6 (see row b Table 2-1). At the same time Arabic handwritten digits form normally 11 classes because of two different ways of writing 3 (see row c Table 2-1), and also the shape of 5 is different in these two languages. CENPARMI Indian digit database, which is used for our experiments in the next sections, has been designed for isolated handwritten Indian digits. It is based on the Arabic style of handwriting of digits (see row c of Table 2-1), and the distribution of the samples in this database is presented in Table 2-2. This database contains 7390 isolated digits for training set, and 3035 digits for testing sets, and all these digits have been collected from real bank cheques [5]. Here we only consider recognition of 10 classes of digits that are common between Arabic and Persian languages (the first 10 columns of Table 2-1). A few such samples used for training set are shown in figure 1. Training and test sets of CENPARMI Indian digit database are distinct sets, but they have similar samples.

(a)	0	1	2	3	4	5	6	7	8	9	0	3	4	6
(b)	۰	۱	۲	۳	۴	۵	۶	۷	۸	۹	۰		۴	۶
(c)	٠	١	٢	٣	٤	٥	٦	٧	٨	٩		٣		

Table 2-1. (a) –Latin digits, (b)-Persian digits, (c)-Arabic digits (Hindi digits)

3. Preprocessing

Because all images in the database are clean and without noise, we did not include noise reduction in our system, but in a real system we need to remove noise from the images. Since images in the database have different sizes (see figure 1), as a preprocessing step in our system, we normalize the size of all the images by changing their size to 64 by 64 pixels, by using the nearest neighbor interpolation method [6] (see figure 2). Then we used these normalized images for the next step (feature extraction). This kind of normalization makes our features invariant to size, and translation, hence the way of feature extraction for all samples will be the same.

Digit	0	1	2	3	4	5	6	7	8	9	Total
Training	3793	782	545	362	307	649	279	233	246	194	7390
Testing	1574	304	225	144	133	263	111	109	98	74	3035

Table 2-2. Distribution of the samples in different classes in training and testing sets

0 1 2 3 4 5 6 7 8 9

Figure 1. Some samples of the digits in the database and their equivalent Latin digits

Figure2. Normalized digits

4. Feature Extraction

The traditional goal of feature extractors is to characterize an object by measurements whose values are very similar to the other objects in the same category, but very different for the objects in different categories [7]. Here after the preprocessing step, different features and their combinations were obtained, and we selected a set of 64 features. Below we explain how to compute these features. First, the matrix of each normalized digit is considered from four different views (top, right, bottom, and left) as shown in figure 3.

In figure 3 directions we selected for each view is also shown. For obtaining each view, the number of white pixels is counted until we reach to the boundary of the digit, and represented as a function (curve) which shows the changes of the boundary that seen from that view in the directions shown. These functions (curves) encode the structural features of the outer boundary of the digit, for example in figure 3 right view shows linear changes in the boundary which seen from this view. The main idea is to transfer the features of each digit in to one-dimensional signals (functions of one variable for example distance or time), and to process these signals for obtaining another features and further recognition. For each signal (curve), after some smoothing by median filtering (with window size 3 pixel), one-dimensional derivative is computed (Figure 4), smoothed, and sampled. Here the sampling rate of ¼ is used. According to our normalization method, each side of the digit has 64 pixels, hence 64 values are obtained from the sampling of the derivatives, and they are used as features for the classification step. These features are easy to interpret, compute and they have good information about the structure of the digit.

5. Support Vector Machines

Support Vector Machines (SVMs) are new learning machines; and they were introduced in 1995 by Vapnik et al. In the past few years they generated a great deal of interest in the community of machine learning because of their excellent generalization in many learning problems [8], such as in handwritten

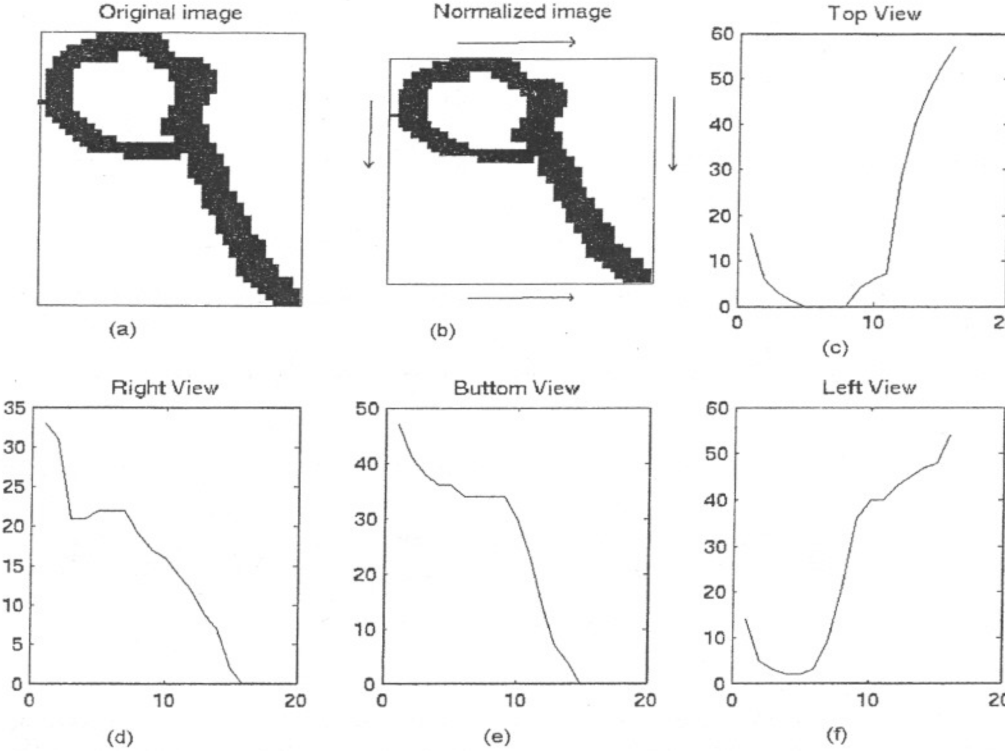

Figure 3. a:Original image, b:Normalized image, c-f:Different views from top, right, bottom, left

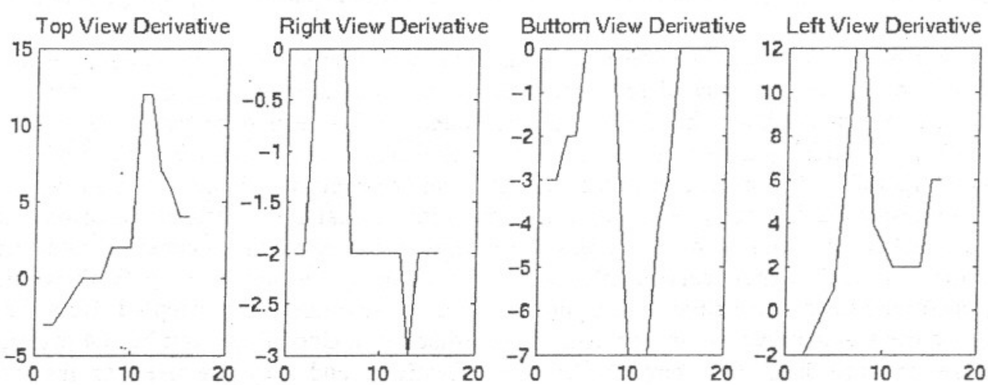

Figure 4. Derivatives of different views in parts of c, d, e, f of figure 3

digit recognition, face recognition, and novelty detection [9]. Here we only introduce some basic formulas for SVMs, and without loss of generality we consider two class classification problems. (for more details, see [9,10]). Given the training samples $\{(X_i, y_i)\}, i = 1,\ldots,N, y_i \in \{-1,+1\}, X_i \in R^d$ where X_i is a d dimensional training sample, y_i is the class label for each X_i, and N is the number of training samples.

Support vector machines first map the data from the input space to a very high dimensional Hilbert space H (so called feature space), using a mapping ϕ.

$$\phi : R^d \rightarrow H$$

The mapping ϕ is implemented implicitly by a kernel function K that satisfies Mercer's conditions [10]. Kernels are functions that give the inner product of each pair of vectors in the feature space, such that:

$$K(X_i, X_j) = \langle \phi(X_i), \phi(X_j) \rangle, \quad \forall X_i, X_j \in R^d$$

where $\langle . \;\; , \;\; . \rangle$ is a notation for inner product [10]. Then in the high dimensional feature space H, we try to find an optimal hyperplane by maximizing the margin between the two classes (see figure 5), and bounding the number of training errors.

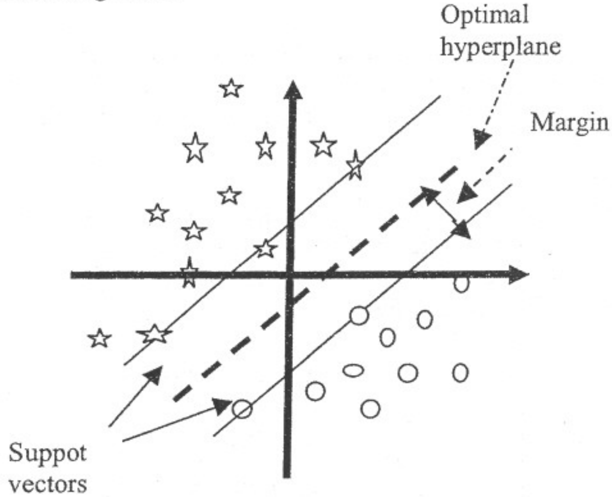

Figure 5. Optimal hyperplane with maximum margin

The decision function can be given as:

$$\begin{aligned} f(X) &= \psi(\langle W, \phi(X) \rangle - b) \\ &= \psi(\sum_{i=1}^{N} y_i\, \alpha_i \langle \phi(X_i), \phi(X) \rangle - b) \\ &= \psi(\sum_{i=1}^{N} y_i \alpha_i K(X_i, X) - b). \end{aligned}$$

where

$$\psi(u) = \begin{cases} 1 & if \quad u > 0 \\ -1 & otherwise \end{cases}$$

and

$$W = \sum_{i=1}^{N} y_i \alpha_i \phi(X_i)$$

If α_i is nonzero then corresponding sample X_i is called support vector [9]. Training an SVM is to find $\alpha_i, i = 1,\dots,N$, which can be achieved by minimizing the following quadratic cost function.

$$\text{Minimize:} \quad L_D(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j K(X_i, X_j)$$

$$\text{Subject to:} \quad \begin{cases} 0 \le \alpha_i \le C, \quad i = 1,\dots,N \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases}$$

where C is a parameter chosen by the user, a large C corresponds to a higher penalty allocated to the training errors. Since the kernel K is semi-positive definite, and the constraints define a convex set [10], the above optimization problem reduces to a convex quadratic programming. Therefore the solutions W, and b can be determined [8,11], and the optimal hyperplane is specified by these solutions.

5.1 Multiple SVM Classifier

Since we have 10 classes of digits, therefore we need 10 SVM classifiers or 10 hyperplanes to separate the digits from each other. For example, one classifier for digit zero (so called SVM0) will separate all the samples of zero from the other digits, and so on. This method of designing multiple SVM classifiers is called one against the others as illustrated in figure 6.

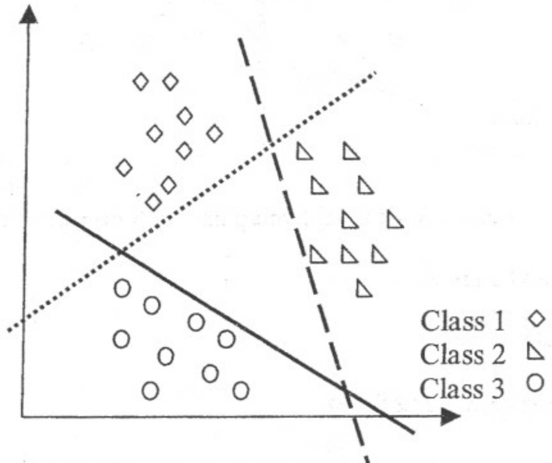

Figure 6. One against the others method for a three class problem

From the 10 outputs of the 10 SVM classifiers, we take the maximum. The SVM classifier that gives the maximum output will determine the class label for the input digit (see decision module in figure 7). In the next section we present the experimental results of our system.

6. Experimental results

Radial Bases Function (RBF) Kernel is used in the experiments, this kernel has just one parameter, and it is very popular. Its general form is shown in equation 6-1,

$$K(X_i, X_j) = \exp(-\theta \cdot \|X_i - X_j\|^2) \quad , \quad \theta > 0 \ , \quad X_i, X_j \in R^d \qquad \text{Eq. 6-1}$$

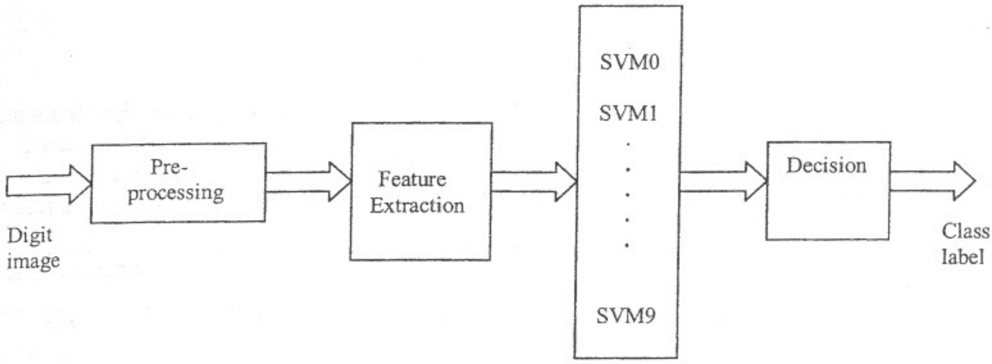

Figure 7. Outline of the system, decision is based on the maximum output of the SVMs

where X_i, X_j are two input vectors and θ is the parameter of the kernel [12]. By changing parameter θ in the kernel function and selecting suitable values for parameter C (as discussed in section 5), we tried to obtain the best recognition rate for each SVM and at the same time for the multiple SVM classifier on the test set. In our experiments values 0.02 and 10 are used for θ and C respectively. The overall performance of recognition rate on the test set for the multiple SVM classifier was 94.14%. In Table 6-1 the numbers along the main diagonal shows the correct recognition of the multiple SVM classifier. As we see most of the errors occur between digits 0, and 1 or 0, and 5. This is due to the high similarity of these digits in the Arabic style of handwriting (see figure 1). So we need more features for distinguishing them.

	0	1	2	3	4	5	6	7	8	9	Total
0	1525	31	0	1	0	11	1	0	2	3	1574
1	37	264	1	0	0	2	0	0	0	0	304
2	6	0	213	3	3	0	0	0	0	0	225
3	2	0	4	134	2	0	1	1	0	0	144
4	4	0	6	0	123	0	0	0	0	0	133
5	29	0	0	0	0	233	0	1	0	0	263
6	0	2	0	0	0	0	107	0	0	2	111
7	4	1	0	3	0	0	0	101	0	0	109
8	5	0	0	0	0	0	0	0	93	0	98
9	6	1	0	0	0	0	3	0	0	64	74

Table 6-1. Recognition matrix for multiple SVM classifier

For comparison with MLP neural network classifier, different MLP classifiers were tested, the one that gave the highest performance had one input layer, two hidden layers, and one output layer, with (64-50-32-10) neurons. The overall recognition rate for this network on the test set was 91.25% as shown in Table 6-2.

7. Conclusion and future works

In this paper we have developed a new set of features for Arabic/Persian digit recognition. Experiments show these features have good discrimination ability. Along with these features, for the first time, we used the Support Vector Machines (SVMs) to recognize Arabic/Persian digits. Also for training and test sets, we used a new database (CENPARMI Indian digit database) with relatively large number of real life

Classifier	Recognition Rate	Error Rate (Substitution Rate)	Rejection Rate
Multiple SVM Classifier	94.14%	5.86%	0%
MLP Classifier	91.25%	8.75%	0%

Table 6-2. Recognition rate for two classifiers with the same features

samples. Finally we presented our results on this database, and compared these results with those from a MLP neural network classifier. We obtained recognition rate for SVM classifier, and for MLP classifier on the same test set. Our database is available to research community upon request to CENPARMI [5]. We think this database can be considered as a basis for the evaluation of Arabic/Persian handwritten digit recognition systems, however this database should be improved further by including enough samples for different figures of some digits such as 0,4, 6, and 5 that exist in Persian.

Acknowledgements: Authors would like to thank NSERC of Canada, and MSRT of Islamic Republic of Iran for their financial support.

References

[1] O. D. Trier, A. K. Jain, T. Taxt, "Feature Extraction Methods for Character Recognition- A Survey," Pattern Recognition, Vol. 29, No. 4, pp.641-662, 1996

[2] C.-L. Liu, K. Nakashima, H. Sako, H. Fujisawa, "Handwritten Digit Recognition Using State-of-he-Art Techniques," Proceedings of 8th IWHFR, Canada, 2002

[3] M. H. Shirali-Shareza, K. Faez, A. Khotanzad, "Recognition of Handwritten Farsi Numerals by Zernike Moments Features and a Set of Class Specific Neural Network Classifiers," Proceedings of The International Conference on Signal Processing Applications, and Technology (ICSPAT), vol. 2, pp.998-1003, Dallas, Texas, 1994

[4] M. H. Shirali-Shareza, K. Faez, A. Khotanzad," Recognition of Handwritten Arabic/Persian numerals by Shadow Coding and an Edited Probabilistic Neural Network," Proceedings of International Conference on Image Processing, Vol.3, Washington D.C., 1995

[5] Y. Al-Ohali, M. Cheriet, C. Y. Suen, "Databases for Recognition of Handwritten Arabic Cheques," Proceedings of 7th IWHFR, Amsterdam, The Netherlands, pp. 601-606, 2000

[6] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Prentice Hall, 2002

[7] R.O.Duda, P.E. Hart, D.G.Stork, Pattern Classification, John Wiley & Sons, Inc., 2001

[8] J. Dong , A. Krzyzak, C.Y. Suen, "A Fast SVM Training Algorithm," Proceedings of First International Workshop: SVM2002, Niagara Falls, Canada, August 2002

[9] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-Based Learning Methods, Cambridge University Press, 2000

[10] J.C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," Data Mining and Knowledge Discovery, pp.121-167, 1998

[11] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall, 1999

[12] M. G. Genton, "Classes of Kernels for Machine Learning: A Statistics Perspective," Journal of Machine Learning Research, Vol.2, pp. 299-312, 2001