Using Box Approach in Persian Handwritten Digits Recognition

Omid Rashnodi
Computer Engineering Department
Science and Research Branch, Islamic Azad University Ahvaz, Iran

Hedieh Sajedi
Computer Engineering Department
Amirkabir university of Technology, Tehran, Iran

Mohammad Saniee Abadeh
Electrical and Computer Engineering Faculty
Tarbiat Modares University
Tehran, Iran

ABSTRACT
In this paper, appropriate feature set based on the box approach has been proposed to achieve higher recognition accuracy and decreasing the recognition time of Persian numerals. In classification phase, support vector machine (SVM) with linear kernel has been employed as the classifier. Feature sets consists of 163 dimensions, which are the average angle and distance pixels which are equal to one in each box the box approach. The scheme has been evaluated on 60,000 handwritten samples of Persian numerals. Using 60,000 samples for training, scheme was tested on other 20,000 samples and 98.945% correct recognition rate was obtained.

Keywords
Box approach, Support vector machines, linear kernel, Persian Numerals

1. INTRODUCTION
Nowadays, recognition systems are used in many fields that have different nature. The optical character recognition (OCR) was started from the recognition of machine printed digits and characters and then it was developed to the recognition of machine printed words. Gradually, handwritten digit, character and word recognition were introduced into this domain. Most researches have been done in Latin languages. Typically an ordinary OCR includes three indicated phase. Pre-processor, feature extraction and classifications phase that output of each step is the input of next step. In the phase Pre-processing operations such as slant correction, normalization and thicking have been done; in phase feature extraction the box approach is used. The last phase (classification) of SVM (Support Vector Machine) as classifier is used. The flowchart of a typical OCR can be shown as Fig1.

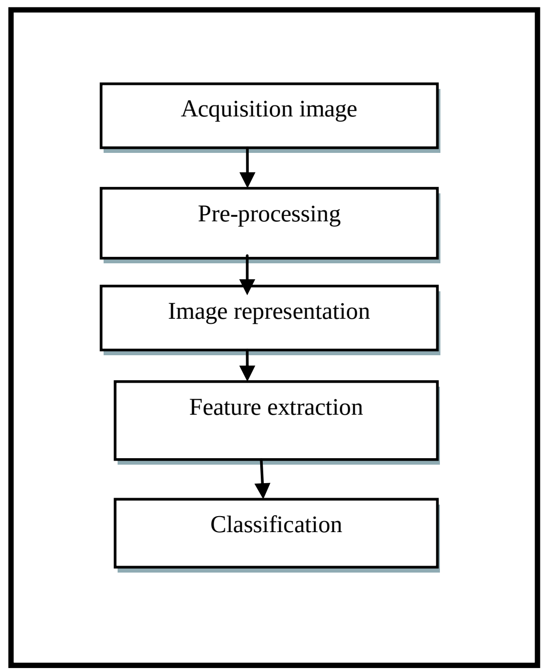

Fig1: Recognition system of handwriting digits

Recognition of handwritten characters is one of the most interesting topics in pattern recognition domain. In OCR applications, handwritten character recognition, especially digit recognition, is dealt with in postal mail sorting, bank check processing, form data entry, etc. For these applications, the performance (accuracy and speed) of digit recognition is crucial to the overall performance. While in pattern classification and machine learning communities, the problem of handwritten digit recognition is a good example to test the classification performance [26]. Due to increasing Persian/Arabic writing usage in many day-to-day businesses in Persian countries, it has been become necessary for machines to understand handwritten materials in Persian. As a part of Persian scripts, numeral strings and isolated numerals play an enormous role. OCR for handwritten documents in some languages (English, Chinese, Japanese, etc.) has reached to a promising level [6]. The OCR for Persian has not grown up like above mentioned languages because of the cursive-ness of handwritten in Persian and multiple forms of each character with respect to its position in words. Achieving this goal, the effect Also box approaches has been used to study of average angle and distance pixels as features which are equal to one in each box. Both of used methods keep the information of input images. In this paper, SVM with Linear kernel has been used as the classifier. In the literature survey particularly relevant to the Persian/Arabic languages, there are many methods for feature extraction and classification. As feature extraction methods segmentation and shadow code [1, 7, 11], fractal code [13], profiles [3, 15], moment [5], template [8], structural feature (points, primitives) [14] and wavelet [9, 12] have been used. For classification, different types of Neural Networks [1, 5, 7, 8, 11, 12, and 13], SVM's [3, 9, and 15] and Nearest Neighbor [14] have been applied. Investigating of previous researches on Persian/Arabic numerals recognition, it seems that more appropriate and effective feature set could have been developed to react recognition phase. To overcome this problem, supplying a more effective feature set has been proposed based on input image average angle and distance pixels which are equal to one in each box, box approaches. Then SVM for classification can be used. This type of feature set expresses the physical shape of input image and extracts its local information and provides more suitable accuracy in experimental part. It is worth mentioning that in the proposed system we apply some pre-processing techniques such as slant detection, thicking, normalization, binarization and etc. The organization of rest of the paper is as follows: Section 2 describes the process overview. Section 3 gives the preprocessing techniques. Section 4 deals with the feature extraction. In Section 5 the classification is presented. Section 6 briefly reports the results of Persian Numeral recognition and Section 7 gives the conclusions.

I.1. Persian numerals characteristics

Persian numerals are used in Iran and in some of its neighboring countries. Comparable to other scripts, in Persian also there are 10 numerals. In Persian/Arabic scripts, alphabets are written from right to left but digits are written from left to right. Persian and Arabic numerals are almost the same; but there are some important differences between handwriting of digits of these two scripts [27]. Generally, in Persian digits, there are two types of writing for the digits 0, 2, 4, 5 and 6. These characteristics make the recognition of Persian numerals more complicated than in other languages.

2. PROCESS OVERVIEW

The character recognition system is usually validated by running it on independent test sets, on which the system has not been trained. For these tests to be conclusive, the validation sets should include a fairly large number of samples to reflect the variety of writing styles that are found in real-life applications.

2.1. Database

For the purpose of validation we need a standard database. For experimental analysis, we considered 60,000 samples for training and 20,000 samples for testing as mentioned in [10].These samples were extracted from different registration forms of entrance examinations of universities in Iran containing Iranian Postal and National Codes. The images were scanned at 200 dpi resolution [10]. The database also includes some samples that cannot be recognized even by humans. The database is divided into two disjoint sets, one for training and another for testing. The training set captures as many variations and different styles of a numeral class as possible. The features extracted from the training set are stored in the KB and at the recognition time, used as reference features for comparing with those of an unknown numeral. So if the training set contains a large number of samples with varied writing styles, the feature set computed from them will be able to reflect these variations, making the recognition system independent of the variations in writing styles. The size of the training set can be different for each numeral class. For example, numeral zero does not have much variation in its writing style, while numeral seven can be written in different ways. Thus, the training set for numeral zero will be smaller than that of numeral seven. The task of the recognition of handwritten numerals has been broken down into the following steps:

(i) Binarization of sample image;

(ii) Correction of image slant;

(iii) Thicking

(iv) Normalization of the image to a standard size;

(v) Feature extraction;

(vi) Recognition;

To enable recognition, steps (i)–(iv) are applied on a training set of all 10 numerals as part of the pre-processing. While performing feature extraction, simultaneously the KB of reference features is created. These steps are depicted in Fig. 2.

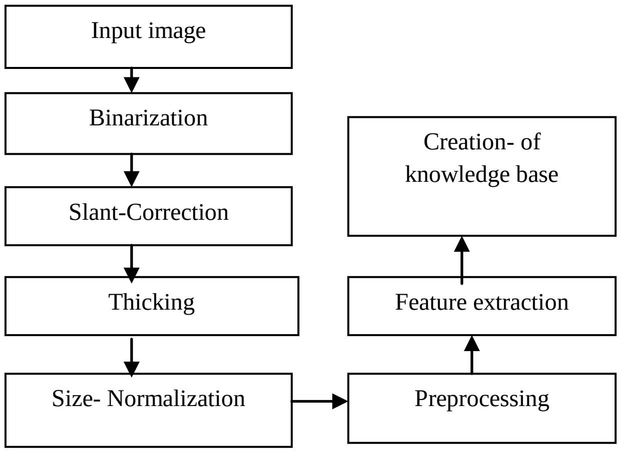

Fig.2. Block diagram describing system implementation

3. PREPROCESSING

The steps of preprocessing are briefly discussed in the following:

3.1. Binarization

Frequently, binarization is carried out before the character recognition phase. Ideally an input character should have two tones, i.e., black and white pixels (commonly represented by 1 and 0, respectively). Image binarization converts an image of up to 256 gray levels into a two-tone image [31].

3.2. Slant correction

A practical character recognizer must be able to maintain high performance regardless of the size and slant of a given character or word. For handwritten characters, one of the major variations in writing styles is caused by slant, which is defined as the slope of the general writing trend with respect to the vertical line. It is important that the system be insensitive to slant. The slant correction algorithm is as follows: The image matrix is divided into upper and lower halves. The centers of gravity of the lower and upper halves are computed and connected. The slope of the connecting line defines the slope of the window (image matrix). The slant-corrected image is obtained by applying the following transformation to all black pixels with coordinate points x, y in the original image [31]:

$$x' = x - y\tan(\beta - def) \qquad (1)$$

$$y' = y$$

where x and y are slant corrected coordinates and *def* is a parameter specifying the default (normal) slant. This is because change the image topology and the correction operation usually creates rough contours on the character. Slant correction needs to precede other preprocessing tasks, i.e., it is applied before thicking. This is because thicking tends to change the image topology and the correction operation usually creates rough contours on the character.

3.3. Thicking

In this section, each pixel of the image 5 pixels is thicked.

3.4. Size normalization

After the binarization, there would be extra 0's on all four sides of a numeral. To standardize the numerals, extra rows and columns containing only zeros are removed from all four sides of the numeral. Hence images are fitted into a window of size 45 ×45. Normalization is thus the process of equating the size of all numerals in order to extract features on the same footing. To achieve this, we use standard bilinear transformation, by which, every input bitmap P, of size m×n, is transformed into a normalized bitmap Q, of size p×q. Both P and Q are quadrilateral regions [31].

4. THE PROPOSED FEATURE EXTRACTION TECHNIQUE

Feature extraction is the crucial phase in numeral identification as each numeral is unique in its own way, thus distinguishing itself from other numerals. Hence, it is very important to extract features in such a way that the recognition of different numerals becomes easier on the basis of the individual features of each numeral. To extract features in proposed method, which is the most effective part of OCR's systems, the following tasks have been performed.

1- For extracting features, we use the Box- approach in Refs. [22], [23]. In this approach for extracting features from the slant corrected, size normalized and thicked binary image Character is considered. The box approach based on spatial division of character image is proposed for feature extraction. In this, the character image of the size of 45× 45 is fitted into horizontal and vertical grid lines of 9× 9. Thus, in this case 81 boxes of size 5×5 are devised and then these boxes are superimposed on the image, so that some of the boxes will have a portion of the image and others empty as shown Fig.3. However, all boxes are considered for analysis. By taking the bottom left corner as the absolute origin (0, 0), the vector distance fork the pixel in b^{th} box at location (i, j) is calculated as:

$$d_k^{\ b} = (i^2 + j^2)^{\frac{1}{2}} \qquad (2)$$

By dividing the sum of distances of all 'l's pixels present in a box with their total number, a normalized vector distance (γ_b) for each box is computed as:

$$\gamma_b = \frac{1}{n_b}\sum_{k=1}^{n_b} d_k^{\ b}, b = 1,2,3,...,81 \qquad (3)$$

where n_b is number of pixels in b[th] box. The above vector distances constitute a set of features based on distance. Similarly, for each k[th] 'l' pixel in a box, the corresponding angle is computed as, $\theta_k = \arctan(-\frac{j}{i})$ for a pixel at (I, J). Then the sum of all angles in a box b is normalized with the number of 'l' pixels present in that box to yield a normalized angle α_b:

$$\alpha_b = \frac{1}{n_b}\sum_{k=1}^{n_b} \theta_k^{\ b} \qquad b = 1,2,3,...,81 \qquad (4)$$

These 81 pairs constitute the complete feature set of a character, which are used for recognition. The points resulting from these 81 feature pairs have been back substituted in their respective boxes store generate the pattern of original character. All the pairs of features corresponding to each box are arranged in a sequential order starting from box-1 to box-81, as shown in Table 1 for further analysis:

Table 1: Features extraction from the Boxes

Number per box	**Two features per box**
Box-1	λ1 , α1
Box-2	λ2 , α2
…	…
Box-81	λ 162 , α162

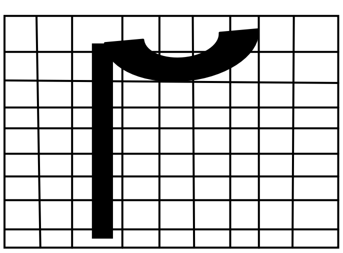

Fig. 3.a. Portions of the numeral lie within some boxes while others are empty

Table 2: important Kernels

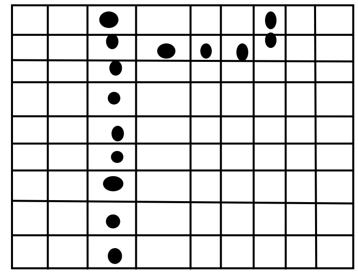

Fig. 3.b. Portions of the numeral lie within some boxes while others are empty

1- Calculating a ratio of length to width of each preprocessed image and considering it as geometrical features for image. The following formula has been utilized to achieve this ratio:

$$ratio = \frac{height}{width} = \frac{length}{width} \qquad (3)$$

According to above steps, 163 features are extracted for each image. In the other words, for each image a feature vector 163 dimensions is considered.

5. CLASSIFICATION

Support vector machines (SVMs) are particular classifiers which are based on the margin-maximization principle. They perform structural risk minimization, which was introduced to machine learning by Vapnik, and have produced excellent generalization performance [16, 17]. For nonlinear problems, SVMs use the kernel trick to produce nonlinear boundaries. The idea behind kernels is to map training data nonlinearly into a higher-dimensional feature space via a mapping function and to construct a separating hyper plane which maximizes the margin. The construction of the linear decision surface in this feature space only requires the evaluation of dot products $\Phi(x).\Phi(y) = k(x, y)$, where $k(x, y)$ is called the kernel function [18, 19, 20, 21]. The discriminate function of a binary SVM is computed by [24]:

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i k(x, x_i) + b \qquad \textbf{(5)}$$

where l is the number of learning patterns, y_i is the target value of learning pattern x_i (+1 for the first class and -1 for the second class), b is a bias and k(x, x_i) is a kernel function which implicitly defines and expanded feature space:

$$k(x, x_i) = \phi(x).\phi(x_i) \qquad \textbf{(6)}$$

where $\phi(x)$ the feature is vector in the expanded feature space and may have Infinite dimensionality [15]. Three types of kernels polynomial kernel, RBF kernel and Linear Kernel are frequently used .They are computed by table2:

Kernel	Definition
Polynomial	$k(x, x_i, p) = (1 + xx_i)^p$
RBF	$k(x, x_i, \sigma^2) = \exp(-\frac{\lVert x - x_i \rVert^2}{2\sigma^2})$
Linear	$k(x, x_i) = x.x_i$

where p, σ are the parameters of the corresponding kernels.

The coefficients $\alpha_i (i = 1,2,..., l)$ in Eq. (5) are determined by solving the following optimization problem:

Minimize $\tau(w) = \frac{1}{2}\lVert w \rVert^2$ **(7)**

Subject to $y_i . f(x_i) \geq 1 - \xi_i \quad , \xi_i \geq 0, i = 1,2,3,..., l$

This is a quadratic programming problem and can be converted in to the following dual problem:

Minimize

$$w(x) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \quad \textbf{(8)}$$

Subject to $0 \leq \alpha_i \leq c, i = 1,2,..., l \; and \quad \sum_{i=1}^{l} \alpha_i \; y_i = 0$

where C is a parameter to control the tolerance of classification errors in learning. Various methods have been proposed to solve the optimization problem (8), especially when there is a large number of training samples, among them, we used the sequential minimal optimization method (SMO) proposed by Platt [25]. SVM was defined for two-class problem and it looked for the optimal hyper-plane, which maximized the distance, the margin, between the nearest examples of both classes, named support vectors (SVs) [2]. Different techniques can be used to extend the binary support vector classifier for the multi-class problems. In this paper is used the One-rest method. In this case, N different binary classifiers are created, where N is the number of the classes. Each of these classifiers is trained to separate one of the classes from the others. An input pattern is assigned to the class of the binary classifier, which gives the maxi mum output value for that pattern. The linear SVM can be extended to a non-linear classifier by using kernel functions like polynomial, sigmoid and Gaussian kernels. We have tested linear, Gaussian, sigmoid and polynomial kernels during our experiments and we received the best result using Linear kernel thus we employed SVMs with Linear kernel as classifier. Details of SVM can be found elsewhere [2, 4].The input feature sets were the 163-dimension. All the SVMs trained with the respective training feature sets and the results explored by using separate test data. We obtained the best results with the linear kernel.

5.1. Soft margin SVM

The soft margin separating hyper plane is used to deal with non-separable data. A set of ξ_i slack variables is introduced to allow errors (or points inside the margin) during the training. A hyper parameter *C* is used to tune the trade-off between the amount of accepted errors and the maximization of the margin:

$$\min \quad \frac{1}{2}\lVert w \rVert^2 + \sum_{i=1}^{N} \xi_i$$

$$s.t. \quad y_i (x_i^t w + b) \geq 1 - \xi_i, i = 1,2,3,..., N$$

This new formulation leads to the same dual problem
With *box constraints* on the Lagrange multipliers:

$$0 \leq \alpha_i \leq C, i = 1,2,3,..., N$$

The tuning of the hyper parameter C is a delicate task.

A common method is to perform a grid search, i.e. to test many values of C and estimate for each the generalization error (usually by cross-validation or on an independent validation set). But this procedure is very time consuming. Some authors proposed other methods including evolutionary tuning [28] or gradient-based approaches [29, 30] for tuning of SVM hyper parameters.

6. PERFORMANCE OF THE PROPOSED SYSTEM

Using 60,000 samples for training, we tested our scheme on other 20,000 samples and obtained 98.945% accuracy. From the experiment, we got an accuracy of 99.6% when the 60,000 data were used as training and the same data set was used for testing.

Further, we considered some noisy images in our test data. The result showed the effectiveness of the proposed feature extraction technique (Table3).

Table 3. Confusion matrix of the result

A	B	C	D	E	F	G	H	I	J	← Classified as
1965	7	0	0	0	25	0	3	0	0	**A=C_0**
1	1996	0	0	0	0	0	3	0	0	**B=C_1**
0	1	1968	3	22	0	3	1	0	2	**C=C_2**
0	3	17	1962	14	4	0	0	0	0	**D=C_3**
0	4	9	15	1972	0	0	0	0	0	**E=C_4**
8	0	0	0	6	1977	0	0	8	1	**F=C_5**
0	6	2	4	2	0	1976	0	0	10	**G=C_6**
0	6	1	0	0	0	0	1993	0	0	**H=C_7**
0	1	0	0	0	0	0	0	1997	2	**I=C_8**
3	3	8	0	0	0	3	0	0	1983	**J=C_9**

Thus the recognition accuracy of each digit would be according to the Table 4.

Table4: Accuracy recognition of Persian digits

9	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**	**Digits**
99.15%	99.85%	99.65%	98.8%	98.85%	98.6%	98.1%	98.4%	99.8%	98.25%	**Accuracy**

6.1. Confusion pairs

In our experiment (with the 98.945% accuracy), we observed confusion numerals in the recognition phase between some digits. In Table 3, we showed Detail of confusing results. The major confusions were amongst 2, 3 and 4. This happened because 2, 3 and 4 look likes each other. From the Table II it may be noted that out of 2000 samples of number three (3), 17(0.0085%) samples misrecognized to numeral 2 and 14(0.007%) samples Misrecognized to numeral 4. In some of the samples, little confusions were also between 0 and 1.

6.2. Comparison of results

To compare the performance of our method, we consider most of the works that are available for Persian numeral recognition. It may be noted from Table 5 that all the existing works were evaluated on smaller datasets. The highest dataset of size 10,000 was used by a recent work due to Ziaratban et al. [8], Where as we used 80,000 data for our experiment. The Highest accuracy was obtained from the work due to Soltanzadeh et al. [3] but they have experimented with Only 8,918 samples and used 257 dimensional features.

We considered 80,000 samples for evaluation of our system and obtained 98.945% accuracy using only163 features.

Table 5: Comparison of different algorithms

Algorithms	**Data size**		**Accuracy**	
	Train	**Test**	**Train**	**Test**
Shirali-shahrezaetal. [1]	2600	1300	--	97.80
Soltanzadeh,Rahmai [3]	4979	3939	--	99.57
Dehghan, Faez [5]	6000	4000	--	97.01
Harifi.,Aghagolzadh [7]	230	500	--	97.60
Ziaratban et al. [8]	6000	4000	100	97.65
Mowlaei, Faez [9]	2240	1600	100	92.44
Hosseini,Bouzerdm [11]	480	480	--	92.00
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
ProposedAlgoritm	60000	20000	99.6	98.945

7. CONCLUSION

In this paper, an efficient feature extracting technique is proposed. From experimental results, it is evident that our features resulted worthy performances (98.945%, 99.6%). We noted that most of misclassified samples were from classes of 2, 3 and 4, which have similar shapes. The recognition of such similar numerals is difficult even by human being. It is obvious that by removing confusion among few classes, we can achieve better performance. To achieve better results (less time for testing, feature set with smaller dimensions .and more accuracy recognition) combination methods which are extracting features and classifiers can be applied.

8. REFERENCES

[1] M. H. Shirali-Shahreza, K. Faze 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, pp. 436-439, 1992.

[2] C. Burges, “A Tutorial on support Vector machines for pattern recognition”, Data Mining & Knowledge Discovery, vol. 2, pp. 1- 43, 1998.

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

[4] V. N. 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, pp. 507-510, 1997.

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

[7] A. Harifi and A. Aghagolzadeh,” A New Pattern for Handwritten Persian/Arabic Digit Recognition”, Journal of Information Technology vol. 3, pp. 249-252, 2004.

[8] M. Ziaratban, K. Faez and F. Faradji “Language-Based Feature Extraction using Template-Matching in Farsi/Arabic Handwritten Numeral Recognition”, Proceedings of 9th International Conference on Document Analysis and Recognition, vol.1, pp. 297-301, 2007.

[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, pp. 547-554, 2003.

[10]H. Khosravi and 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, pp. 1133-1141, 2007.

[11]H. Mir Mohammad Hosseini and A. Bouzerdoum,”A Combined Method for Persian and Arabic Handwritten Digit Recognition”, Australian New Zealand Conference on Intelligent Information System, pp. 80 – 83, 1996.

[12]A. Mowlaei, K. Faez ,A. Highlight, ”Feature Extraction with Wavelet Transform for Recognition of Isolated Handwritten Farsi/Arabic Characters and Numerals”, Digital Signal Processing vol. 2, pp. 923- 926, 2002.

[13]S. Mozaffari, K. Faez & H. Rashidy Kanan, “Recognition of Isolated Handwritten Farsi/Arabic Alphanumeric Using Fractal Codes”, Image Analysis and Interpretation, 6thSouthwest Symposium, pp. 104-108, 2004.

[14]S. Mozaffari, K. Faez and M. Ziaratban, “Structural Decomposition and Statistical Description of Farsi/Arabic Handwritten Numeric Characters”, Proceedings of the 8th Intl. Conference on Document Analysis and Recognition, vol. 1, pp. 237- 241, 2005.

[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, pp. 300-307, 2003.

[16] V.N. Vapnik, “Principles of risk minimization for learning theory, Advances in Neural Information Processing Systems”, vol. 4, Morgan Kaufman, San Mateo, CA, pp. 831 – 838, 1992.

[17] V.N. Vapnik, “Statistical Learning Theory”, Wiley, New York, 1998.

[18] B.E. Boser, I. Guyon, V. Vapnik,” A training algorithm for optimal margin classifiers”, computes. Learn. Theory pp.144–152, 1992.

[19] N. Cristianini, J. Shawe-Taylor, “An Introduction to Support Vector Machines”, Cambridge University Press, Cambridge, 2000.

[20] B. Scholkopf, A.J. Smola,” Learning with Kernels”, MIT Press, Cambridge, MA, 2002.

[21] J. Shawe-Taylor, N. Cristianini, “Kernel Methods for Pattern Analysis”, Cambridge University Press, Cambridge, 2004.

[22] M. Hanmandlu, M.H.M. Yusof, M. Vamsi Krishna, “Off-line signature verification and forgery detection using fuzzy

modeling”, Pattern Recognition vol. 3, no. 38, pp.341–356, 2005.

[23] M. Hanmandlu, K.R. Murali Mohan, S. Chakra borty, S. Goyal, D. Roy Choudhury, “Unconstrained handwritten character recognition based on fuzzy logic”, Pattern Recognition vol. 3, no. 36 , pp. 603–623, 2003.

[24].Burges, C.J.C., A tutorial on support vector machines for pattern recognition. Data Min. Know. Disc. 2, pp.121–167, 1998.

[25]Platt, J.C., Fast training of support vector machines using sequential minimal optimization. In: Scholkopf, B., Burges, C., Smola, A.J. (Eds.), Advances in Kernel Methods—Support Vector learning. MIT Press, Cambridge, Massachusetts, pp. 185–208. Chapter 12, 1999.

[26]N. Sharma, U. Pal and F. Kimura, “Recognition of Handwritten Kannada Numerals”, Proceedings of the 9th International Conference on Information Technology, vol. 1, pp. 133-136, 200

[27]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, pp. 300-307, 2003.

[28] F. Friedrichs, C. Igel, Evolutionary tuning of multiple SVM parameters, in: M. Verleysen (Ed.), Proceedings of the 12th European Symposium on Artificial Neural Networks (ESANN), Bruges, Belgium, pp. 519–524, 2004.

[29] O. Chapelle, V.N. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn pp. 131–159, 2002.

[30] N.E. Ayat, M. Cheriet, C.Y. Suen, Optimizing the SVM kernels using an empirical error minimization scheme, in: S. Lee, A. Verri (Eds.), Proceedings of the International Workshop on Pattern Recognition with Support Vector Machines, Lecture Notes in Computer Science, vol. 2388, pp. 354–369, 2002.

[31] M. Hanmandlu, O.V. Ramana Murthy, Fuzzy model based recognition of handwritten numerals, the journal of pattern recognition society, pp. 1840 – 1854, 2006.