International Journal of Computer Applications (0975 – 888)
Volume 48– No.11, June 2012

Online Handwriting Recognition of Hindi Numerals using Svm

Deepika Wadhwa
Computer Science and Engineering
Thapar University, Patiala
Patiala, Punjab, India

Karun Verma
Assistant Professor, CSE
Thapar University, Patiala
Patiala, Punjab, India

ABSTRACT
Handwriting recognition has attracted many researchers across the world. Recognition of online handwritten Hindi numerals is a goal of many research efforts in the pattern recognition field. This paper presents an online handwritten Hindi numeral recognition system using Support Vector Machines. Co-ordinate points of the input handwritten numeral are collected; various algorithms for pre-processing are applied for normalizing, resampling and interpolating missing points. Angle, curvature along with the x and y co-ordinates are extracted from the input handwritten numeral. The data obtained is then used for recognition using the kernel functions of SVM. The recognition accuracies are obtained on different schemes of data using the four kernel functions of SVM.

General Terms
Character recognition, online handwriting recognition.

Keywords
Preprocessing, Feature extraction, SVM (Support Vector Machine).

1. INTRODUCTION
The amount of information that can be processed and stored by computers is increasing at a tremendous rate. Given this increase in the rate, the ease with which the information can be exchanged between the user and a computer is becoming a serious problem.

In order to be effective, interface should be natural to the user as well as efficient. Various input devices have some limitations when compared with input through natural handwriting for complex scripts like Chinese due to large number of alphabets. Thus, handwriting recognition proves to be a very attractive input method. It is the ability of a computer to receive and interpret intelligible handwritten input from sources such as paper documents, touch screens and other devices [1]. It has two distinct areas: offline, in which the image of the handwritten text is sensed from a piece of paper by optical scanning, or the movement of the pen tip may be sensed by a digital pen based computer screen surface called online recognition.

Various phases of online handwriting recognition are generalized in Figure: 1. First phase is data collection, which collects the sequence of co-ordinate points of the moving pen. A device called Transducer is required to capture handwriting as it is written with the help of a digital pen. Noise and distortions present in the input text due to some limitations is removed using pre-processing which forms the second phase. This phase consists of five main steps [2] namely: size normalization and centering, interpolation, smoothing, slant correction and resampling. The feature extraction phase determines which properties of the preprocessed data are most meaningful and should be used further for recognition. Various techniques are used to extract features like Direction code histogram [3]. In segmentation phase, data are represented at character or stroke level so that nature of each character or stroke is studied individually. The four best known approaches for recognition are: template matching, syntactic or structural matching, neural networks and statistical classification [4]. Post processing is applied in order to correct the results which are misclassified due to various problems by applying linguistic knowledge.

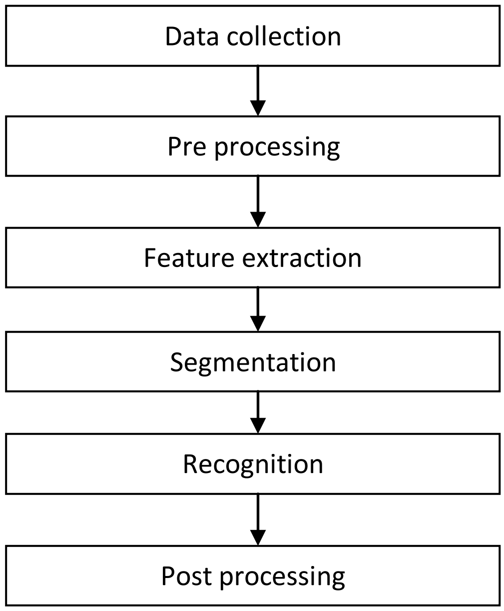

Figure: 1 Phases of Online Handwriting Recognition

In this paper, an Online Handwritten Hindi Numeral Recognition System has been developed using Support Vector Machine. The input data is first preprocessed using Normalization, Interpolation, resampling, slant correction and smoothing. Features are then extracted from the pre processed data. Pre processed data along with the extracted features are then fed to recognition phase where recognition is done using the kernel functions of SVM.

The rest of the paper is organized as follows: Section 2 contains an introduction to Hindi numerals, section 3 introduces SVM and its advantages, section 4 presents the experiments with SVM and its kernels with results obtained and section 5 discusses the conclusions that conclude the paper and the future work to be implemented further.

2. INTRODUCTION TO HINDI NUMERALS

Devanagari is the most popular amongst all Indic scripts. It is the main script for writing Hindi and various other languages. It is a two-dimensional composition of symbols attached to one or more of the four sides of a basic character, also called Conjunct character. Devanagari script is written from left to right order. It also has a native set of ten symbols for numerals (Figure 2). The present study is based on recognizing these numerals in online handwriting.

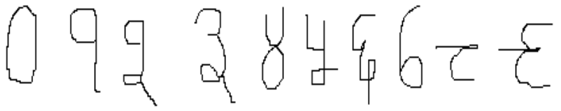

Figure 2: Hindi numerals from 0-9

A method for the recognition of handwritten Hindi numerals based on structural descriptors of numeral shapes is presented in [5]. NN classifiers and neural networks are applied for handwritten Hindi numeral recognition in [6] [7]. In [8], input fuzzy modeling is used for the recognition of handwritten Hindi numerals. SVM's have been used for recognition of numerals of Kannada script in [9].

3. INTRODUCTION TO SVM

SVM was first heard in 1992, introduced by Boser, Guyon and Vapnik. SVMs are a set of related supervised learning methods used for classification and regression. They are based on the concept of decision planes that define decision boundaries. A decision plane is the one that separates between a set of objects having different class memberships.

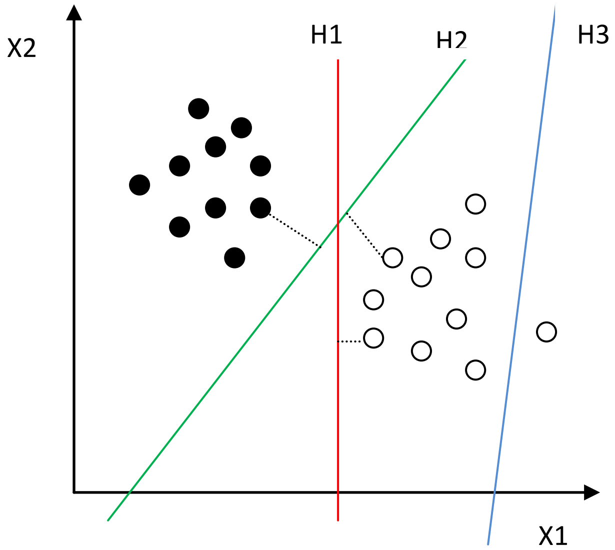

Figure 3: Concept of decision planes

In the above Figure 3, objects belong either to solid or hollow circles. H3 does not separate the two classes of objects; H1 does with a small margin and H2 with the maximum margin.

Classification tasks based on drawing separating lines to distinguish between objects of different class memberships are known as hyperplane classifiers. SVMs are particularly suited to handle such tasks [10]. The process of re-arranging the objects is known as mapping.

The idea of SV Machines is to map the training data non-linearly into a higher dimensional feature space and construct a separating hyperplane with maximum margin there. Thus, it is a hyperplane classifier that aims to maximize the geometric margin of the hyperplane [11]. This yields a non-linear decision boundary in input space as in Figure 4. The data points on the margin are called support vectors.

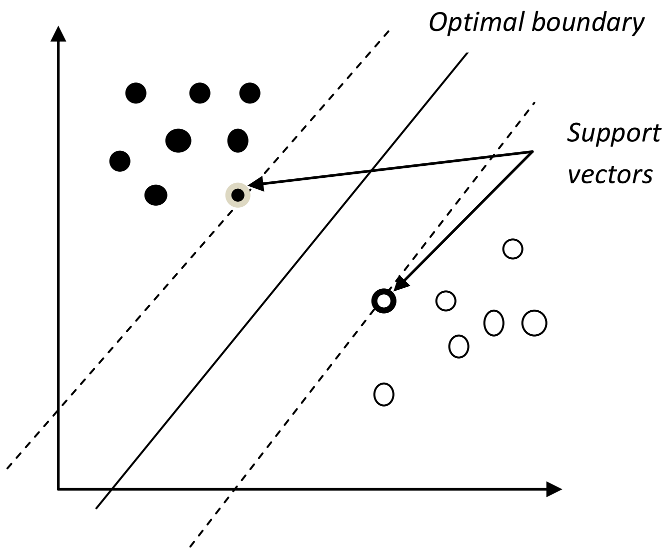

Figure 4: concept of SVM

Support vector machine classifiers fall under the category of statistical classifiers. They have gained immense popularity in recent years providing excellent recognition results in various applications. It has been used as an alternative to methods such as neural networks, hidden markov models due to its following advantages:

- SVMs exhibit good generalization.
- Few parameters are required for tuning the learning method as compared to neural network where architecture and various parameters must be found.
- It takes into account structural behavior along with the experimental data for a principled generalization capability based on SRM (structural risk minimization) [12].

In [13], SVM approach is used to recognize strokes in Telugu script. The set of strokes are segmented into subsets based on the relative position of the stroke in a character. An SVM based classifier is built for recognition of strokes in each subset. A rule based approach is used to recognize a character from the sequence of stroke classes given by the stroke classifier. SVM classifier gives better results as compared to the other classifiers for the handwritten numeral recognition of Kannada script in [14]. Basic SVM only handle two-class classification. Multiclass classification requires training of many two class classifiers and in classification, voting schemes are used for selecting the correct class [15].

4. RECOGNITION OF NUMERALS

Experiments are conducted to find the recognition accuracies using SVM. Data collection, pre processing, feature extraction and recognition are the main phases that are followed and are described as below:

4.1 Data Collection

The characters are inputted with the help of mouse. Points generated by the pen movements are stored in a list. The numerals are written by ten writers. Each writer was asked to write ten samples of each Hindi numeral, thus 1000 samples in all are created during this work. The data is then sent for pre-processing.

4.2 Pre-Processing

In all five pre processing operations are performed on the collected data. Normalization of the stroke is done so that every stroke is of the same size. When a character is drawn with high speed, it will have some missing points in it. These missing points are interpolated using Bezier [16], in which a set of four consecutive points is considered for obtaining the Bezier curve. Bezier interpolation is applied between the points where distance is greater than one. Resampling is done to fix the number of points to 64 and retain the shape of the character. Sharpness of edges is removed by smoothing in which each point of the list is modified by the mean value of the neighbors and angle subtended at the k position from each end [17]. Slant correction is done to overcome the bend or slant in handwriting using 8-direction code method [18].

4.3 Feature Extraction

Two features are extracted from the collected data namely direction angles and curvature. Direction angles are calculated based on the 8-direction code method between points 1 to 3; 3 to 5 and so on. Each angle is assigned a number from 1 to 8 depending on the 8-direction code method. Curvature is the amount by which a geometric object deviates from being flat, or straight in the case of a line. It is calculated between points 1, 2, 3; 3, 4, 5 and so on. The formula used for calculating radius of curvature is:

$$\text{Radius of curvature} = \frac{\left[1 + \left(\frac{dy}{dx}\right)^2\right]^{\frac{3}{2}}}{\frac{d^2y}{dx^2}}$$

Thus, in all 32 direction angles and 32 curvature points are calculated for the 64 resampled points. These features, along with the x, y coordinates collected are stored in a file and directly fed to the recognition phase.

4.3 Recognition using SVM

By the use of kernel function k(x, y) selected to suit the problem in SVM, it is possible to compute the separating hyperplane without explicitly carrying out the mapping into the feature space [19]. The four kernel functions [20] used in SVM are described below:

- Linear: Linear SVN is linearly scalable with the size of training data set.

$$k(x, y) = x * y$$

- Polynomial: It is a non-stationary kernel suited for problems where training data is normalized.

$$k(x, y) = (x.y) \wedge d$$

Where d is the polynomial degree

- RBF (radial basis function): It is defined on the interval [-1, 1]

$$k(x, y) = \exp(-\| x - y \| \wedge 2 / (2\sigma \wedge 2))$$

- Sigmoid: With gain κ and offset θ

$$k(x, y) = \tanh(\kappa(x.y) + \theta)$$

4.3.1 Implementation

Angle and curvature for each set of input co-ordinate (x, y) are calculated. This data is directly used for recognition. 75% data has been used in training and 25% in testing. The data are partitioned into following six schemes for recognition purpose:

Scheme 1: x, y, direction angle, curvature

Scheme 2: x, y, direction angle

Scheme 3: x, y, curvature

Scheme 4: direction angle, curvature

Scheme 5: direction angle only

Scheme 6: curvature only

For each scheme, the four kernel functions of SVM are used. Recognition accuracy is calculated for each scheme separately using the four kernels.

4.3.2 Results

Results obtained by the recognition using SVM for all the four kernels for each of the six data schemes are depicted in the following tables:

Table 1: SCHEME 1

Kernel	Accuracy
Linear	**98.700**
Polynomial	**98.100**
RBF	**96.500**
Sigmoid	**10.000**

Table 2: SCHEME 2

Kernel	Accuracy
Linear	**98.900**
Polynomial	**98.100**
RBF	**98.200**
Sigmoid	**97.800**

Table 3: SCHEME 3

Kernel	Accuracy
Linear	**97.400**
Polynomial	**97.400**
RBF	**96.900**
Sigmoid	**10.000**

Table 4: SCHEME 4

Kernel	Accuracy
Linear	**96.400**
Polynomial	**96.400**
RBF	**96.500**
Sigmoid	**10.000**

Table 5: SCHEME 5

Kernel	Accuracy
Linear	**95.800**
Polynomial	**95.200**
RBF	**95.400**
Sigmoid	**91.700**

Table 6: SCHEME 6

Kernel	Accuracy
Linear	**83.800**
Polynomial	**83.700**
RBF	**72.700**
Sigmoid	**10.000**

5. CONCLUSIONS

The main goal of this study was to develop an online handwritten Hindi numeral recognition system. This paper describes the pre processing, feature extraction and the recognition phase. Pre processing and feature extraction are done prior to recognition to increase the efficiency of the character recognition system. Direction angle and curvature are the two features extracted. Recognition is done using four kernel functions of SVM by dividing the data into six schemes depending on the features extracted. Good recognition accuracies have been obtained for all the six schemes and the kernels. Results obtained are reasonably good when the linear kernel is used as compared to the other kernels. The highest accuracy shown by the linear kernel is 98.900%. The results also prove that direction angle and curvature are two very important features and enhance the recognition process. The two features showed good results even when used individually for character recognition.

The present work can be extended for alphanumeric characters and Devanagari words. Recognition accuracies can be further improved by extracting more features from the input handwritten character and by applying new pre processing techniques such as word rotation. By increasing the database and the number of writers in future, recognition accuracies can be further improved.

6. REFERENCES

[1] Wikipedia, http://en.wikipedia.org/wiki

[2] Jaeger, S., Manke, S., Reichert, J., and Waibel A. 2001. Online Handwriting Recognition: The Npen++ Recognizer. International Journal of Document Analysis and Recognition, vol. 3, no. 3, pp. 169-180.

[3] Patil, P. M. and Sontakke, T. R. 2007. Rotation, Scale and Translation invariant Handwritten Devanagari Numeral Character Recognition using General Fuzzy Neural Network. Pattern Recognition Society.

[4] Jain, A. K., Duin, R. P. W. and Mao, J., 2000. Statistical Pattern Recognition: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1.

[5] Elnagar, A., Al-Kharousi, F., and Harous, S. 1997. Recognition of Handwritten Hindi Numerals using Structural Descriptors. IEEE.

[6] Dr. Mahdy, Y. B. and ENG. El-Melegy, M. T. 1996. Encoding Patterns for Efficient Classification by Nearest Neighbor Classifiers and Neural Networks with Application to Handwritten Hindi Numeral Recognition. Proceedings of ICSP.

[7] Hanmandlu, M., Grover, J., Madasu, V. K., and Vasikarla, S. 2007. Input Fuzzy Modeling for the Recognition of Handwritten Hindi Numerals. International Conference on Information Technology.

[8] Sharma, A., Kumar, R., and Sharma, R. K. 2008. Online Handwritten Gurmukhi Character Recognition Using Elastic Matching. Congress on Image and Signal Processing.

[9] Rajashekararadhya, S. V. and Ranjan, P. V. 2009. Support Vector Machine based Handwritten Numeral Recognition of Kannada Script. IEEE International Advance Computing Conference.

[10] Bellegarda, E. J., Bellegarda, J. R., Namahoo, D., and Nathan K. S., 1993. A probabilistic framework for online handwriting recognition. Proceedings of IWFHR III, pp. 225-234.

[11] Shanthi, N. and Duraiswamy, K. 2010. A Novel SVM-Based Handwritten Tamil Character Recognition System. Pattern Anal Applic.

[12] Bahlmann, C., Haasdonk, B., and Burkhart, H. 2002. Online Handwriting Recognition with SVM- A Kernel Approach. Proceedings of the eighth International Workshop on Frontiers in handwriting Recognition.

[13] Jayaraman, A., Sekhar, C. C., and Chakravarthy, V. S. 2007. Modular Approach to Recognition of Strokes in Telugu Script. Ninth international Conference on Document Analysis and Recognition.

[14] Rajashekararadhya, S. V., and Ranjan, P. V. 2009. Support Vector Machine Based Handwritten Numeral Recognition of Kannada Script. IEEE International Advance Computing Conference.

[15] Ahmad, A. R., Khalid, M., Gaudin, C. V., and Poisson, E. 2004. Online Handwriting Recognition using Support Vector machine. IEEE.

[16] Unser, M., Aldroubi, A., and Eden, M. 1993. B-Spline Signal Processing: part 2- Efficient Design and Applications. IEEE Transactions on Signal Processing, Vol. 41, No. 2, pp. 834-848.

[17] Kavallieratou, E., Fakatakis, N., Kolkkinakis, G. 2002. An Unconstrained Handwriting Recognition System. International Journal of Document Analysis and Recognition, vol 4, No. 4, pp. 226-242.

[18] Ding, Y., Kimura, F. and Miyake, Y. Slant Estimation for Handwritten Words by Directionally Refined Chain Code.

[19] HEARST, Marti A., et al., 1998. Support Vector Machines. IEEE Intelligent Systems, 13(4), 18-28.

[20] Shanthi, N. and Duriaswamy, K. 2010. A Novel SVM Based Handwritten Tamil Character Recognition System. Pattern Anal Applic.