Recognition of Devanagari Handwritten Numerals using Gradient Features and SVM

Ashutosh Aggarwal Rajneesh Rani RenuDhir

Department of Computer Science and Engineering
Dr B.R. Ambedkar National Institute of Technology
Jalandhar- 144011, Punjab (India)

ABSTRACT

Recognition of Indian languages is a challenging problem. In Optical Character Recognition (OCR), acharacter or symbol to be recognized can be machine printed or handwritten characters/numerals. Several approaches in the past have been proposed that deal with problem of recognition of numerals/character depending on the type of feature extracted and way of extracting them. In this paper also a recognition system for isolated Handwritten Devanagari Numerals has been proposed. The proposed system is based on the division of sample image into sub-blocks and then in each sub-block Strength of Gradient is accumulated in 8 standard directions in which Gradient Direction is decomposed resulting in a feature vector with dimensionality of 200. Support Vector Machine (SVM) is used for classification. Accuracy of 99.60% has been obtained by using standard dataset provided by ISI (Indian Statistical Institute) Kolkata.

General Terms

Pattern Recognition, Indian Scripts, Optical Character Recognition.

Keywords

Devanagari NumeralRecognition, HandwrittenRecognition, Gradient, Gradient Feature Extraction,SVM.

1. INTRODUCTION

Optical character recognition (OCR) is a field of automatic recognition of different characters from a document image. This field is divided into two parts one is recognition of machine printed characters and second is recognition of handwritten characters. Now days, recognition of handwritten characters is very challenging task because different people have different handwriting styles. So, handwritten OCR is still a subject of active research.

Devanagari script is the most widely used Indian script and around 500 million people use it. Recognizing Handwritten Numerals have numerous applications including those in postal sorting, bank cheque processing, job application form sorting and automatic scoring of tests containing multiple choice questions.

Techniques used in OCR system follow mainly two steps (a) A feature vector is formed from sample image and (b) The feature vector is classified into classes using a classifier. The right type of feature extraction method selected plays very important role to achieve high accuracy. So the feature extraction algorithm must be capable to generate similar feature sets for a variety of instances of the same symbol. Ivind and Jain[1] present a survey of various feature extraction methods used in character recognition. On the other hand, choice of classifier, to discriminate given features, is not an easy task because classifier recognition accuracy depends on training set and number of free parameters of classifier.

In the literature survey we have found that numbers of authors have attempted to recognize the Handwritten Devanagari Numerals with different-2 techniques. G.S.Lehal and NivedanBhatt[4] have proposed a contour extraction technique and obtained 89% accuracy. Reena Bajaj et al.[5] have employed three different kinds of features namely, density features, moment features and descriptive features for classification of Devanagari Numerals and obtained 89.68% accuracy. R J Ramteke et al.[6] have proposed a method based on invariant moments and the divisions of image for the recognition of numerals and achieved 92% accuracy. U. Bhattacharya et al.[7] have used a combination of ANN (Artificial Neural Net-work) and HMM (Hidden Markov Model) classifier on 16273 samples of Handwritten Devanagari Numerals and obtained 95.64% accuracy. N Sharma et al.[18] have proposed a quadratic qualifier based technique and used 22546 samples for his experiment and achieved 98.86% accuracy. But before our work highest accuracy of 99.56 for Devanagari Numerals were registered by U.Pal et.al.[8]. The features used by them were obtained from the directional information of the contour points of the numerals. A Modified Quadratic Discriminant Function (MQDF) was used to classify the extracted features. Other approaches for recognition of Devanagari numerals can found in [2], [3], [9-20].

In our proposed approach, initially the Gradient Vector is calculated at all image pixels and sample image is divided into 9x9 sub-blocks. Then in each sub-block Strength of Gradient is accumulated in each of 8 standard directions in which Gradient Direction is decomposed. Finally image is down sampled to 5x5 blocks from 9x9 blocks using a Gaussian Filter giving a feature vector of dimensionality 200 (5x5x8). Accuracy of 99.60% is obtained using Support vector Machines (SVM) as classifier.

An overview of the paper is as follows: In Section 2, the Devanagari Numeral dataset provided by ISI Kolkata which we have used in our experiment is described. Section 3, covers our proposed approach for Devanagari Numeral Recognition right from pre-processing of images to use of Gradient as our Feature Extraction Technique and finally a brief introduction of SVM classifier. In Section 4, Experimental results and analysis are provided. Finally, the conclusion & acknowledgment have been offered in Section 5 and 6 respectively.

2. DEVANAGARI NUMERAL DATABASE

The database is provided by the ISI (Indian Statistical Institute, Kolkata) [21]**.** This dataset is similar to what used by [19] in their work. The reason for using this dataset is that, this dataset is known to be the standard bench mark dataset of Devanagari Numerals and various authors all across globe use this dataset as their base for testing their feature extraction algorithms and comparing their results with those obtained by the others. This dataset contains all the possible variations in handwriting styles. The printed Devanagari Numerals are shown in *Fig.1* and it is seen that there are many variations in the shapes of numerals 5, 8 and 9 in their printed forms. In *Fig. 2*, there are shown few samples of the Handwritten Devanagari Numerals from this very database. The distributions of training data and testing data are shown in *Table 1.*

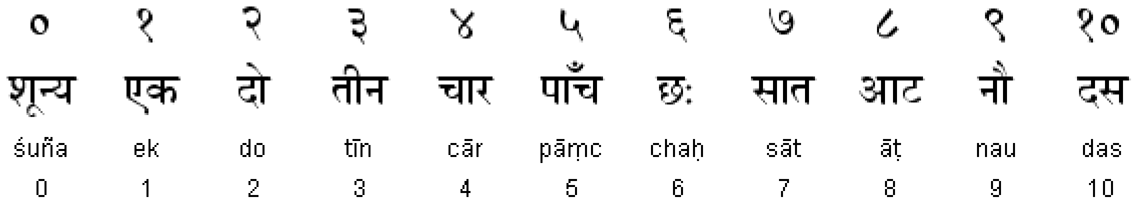
० १ २ ३ ४ ५ ६ ७ ८ ९ १०
शून्य एक दो तीन चार पाँच छः सात आठ नौ दस
śuña ek do tīn cār pāṃc chaḥ sāt āṭ nau das
0 1 2 3 4 5 6 7 8 9 10

Fig. 1: Devanagari Numerals

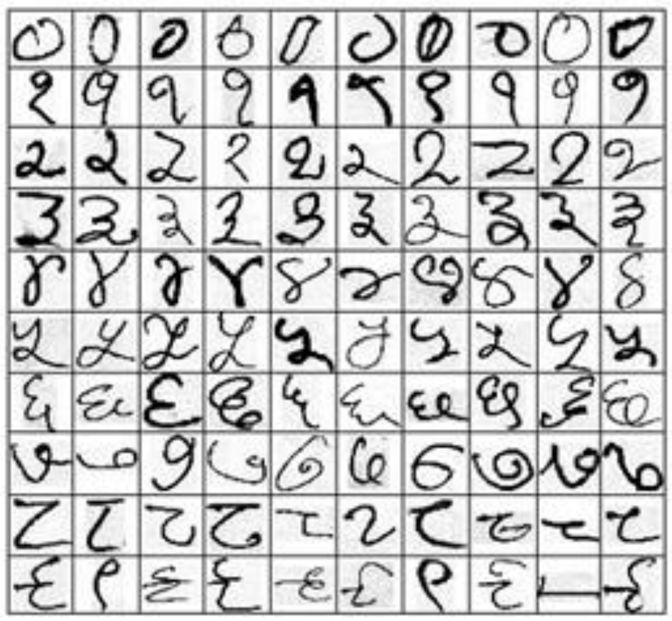

Fig. 2: Handwritten Devanagari Numeral Samples

Table 1: Distribution of numerals in Devanagari Database provided by ISI Kolkata

Digits	Training Set	Test Set	Total
0	1844	368	2212
1	1891	378	2269
2	1891	378	2269
3	1882	377	2259
4	1876	376	2252
5	1889	378	2267
6	1869	374	2243
7	1869	378	2247
8	1887	377	2264
9	1886	378	2264
Total	18784	3762	22546

3. PROPOSED METHOD

In this experiment we have used the standard database of 22546 images provided by ISI Kolkata. The entire database consists of grayscale images that contains noise and are not in normalized form. We have used SVM classifier with RBF Kernel for classification and obtained 99.60% accuracy by 5-fold cross-validation of whole database.

Our entire work is divided into various stages right from pre-processing of non-normalized images into normalized ones in Stage 1 to utilization of Gradient feature extraction technique to extract features from these normalized Devanagari Numeral images in Stage 2 and finally ending up with very promising recognition results obtained by using SVM as our classifier.

3.1 Stage 1: Pre-processing

Character normalization is considered to be the most important pre-processing operation for character recognition. Normally, the character image is mapped onto a standard plane (with predefined size) so as to give a representation of fixed dimensionality for classification. The goal for character normalization is to reduce the within-class variation of the shapes of the characters/digits in order to facilitate feature extraction process and also improve their classification accuracy. In our approach we have used Non-Linear character normalization technique to map images on standard planeof size90×90. For more information on various normalization techniques refer [22].

Other steps that have been performed apart from character normalization in order to pre-process the image before feature extraction are listed below and similar to what done by [19]:

i) Intensity values of an image were adjusted.
ii) Images were converted into binary form by choosing threshold value 0.8.
iii) All connected components (objects) that have fewer than 30 pixels were removed from the binary images.
iv) Median filtering, which is a nonlinear operation often used in image processing to reduce "salt and pepper" noise was applied on all images.

3.2 Stage 2: Gradient Feature Extraction

In our approach we have used gradient information of numeral images as our extracted features.This approach is adopted from our earlier practice [23] used to recognize isolated Devanagari handwritten characters.

3.2.1 Introduction to Gradient

The gradient measures the magnitude and direction of the greatest change in intensity in a small neighbourhood of each pixel. (In what follows, "gradient" refers to both the gradient magnitude and direction). Gradients can be computed either by means of the Sobel operator or Robertz operator or Prewitt operator. In our work we have used Sobel operator to determine **Gradient Vector** $[\mathbf{G_x},\mathbf{G_y}]^{\mathbf{T}}$, where $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$ are the horizontal and vertical gradient components. The Sobel templates used to compute the horizontal & vertical components of the gradient are shown in *Figure 3*.

1	2	1
0	0	0
-1	-2	-1

Horizontal Component

1	0	1
-2	0	2
-1	0	1

Vertical Component

Fig. 3: Sobel masks for gradient

Given an input image I of size $N_1 \times N_2$, at each pixel (i, j) of image I where i=1 to N_1 and j=1 to N_2, its neighbourhood is convolved with these templates to determine these $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$, respectively. (The eight neighborhood of pixel (i, j)is shown in *Figure 4*).

(I-1, J-1)	(I, J-1)	(I+1, J-1)
(I-1, J)	(I, J)	(I+1, J)
(I-1, J+1)	(I, J+1)	(I+1, J+1)

Figure 4: 8-neighborhood of pixel (I, J)

Eq. (1) and (2) represents mathematical representation of $\boldsymbol{G_x}$ and $\boldsymbol{G_y}$:

$$G_x(x,y) = I(i+1,j-1) + 2 * I(i+1,j) + I(i+1,j+1) - I(i-1,j-1) - 2 * I(i-1,j) - I(i-1,j+1) \quad (1)$$

$$G_y(x,y) = I(i-1,j+1) + 2 * I(i,j+1) + I(i+1,j+1) - I(i-1,j-1) - 2 * I(i,j-1) - I(i+1,j-1) \quad (2)$$

The Gradient Strength and Direction can be computed from the Gradient Vector $[G_x,G_y]^T$ *as shown below using Eq. (3) and (4):*

The Gradient Magnitude is then calculated as:

$$Gradient\ Magnitude = |G(i,j)| = \sqrt{(G_x(i,j))^2 + (G_y(i,j))^2} \quad (3)$$

Gradient direction is calculated as:

$$theta(i,j) = tan^{-1}\{G_y(i,j)/G_x(i,j)\} \quad (4)$$

3.2.2 Calculation of Gradient Feature Vector

A Gradient Feature Vector is composed of the strength of gradient accumulated separately in different directions. The Gradient Feature Vector which we have used in our approach comprises of 200 features per numeral image or in other words we can say that its dimensionality is 200. Generation of this 200 dimensional Gradient feature vector is explained below:

After obtaining gradient vector of each pixel, the gradient image is decomposed into four orientation planes or eight direction planes (chaincode directions) shown in *Figure 5*. The angle range $\{\theta | -\pi \le \theta < \pi\}$is divided into 8-ranges as shown in *Figure 6* where $\Omega_1=\Omega_2=\Omega_3=\Omega_4=\Omega_5=\Omega_6=\Omega_7=\Omega_0=2\pi/8$.

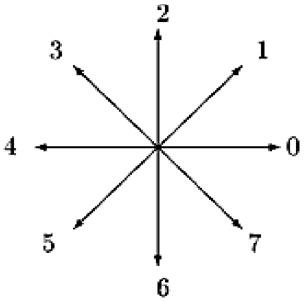

Fig. 5: 8 directions of Chaincodes

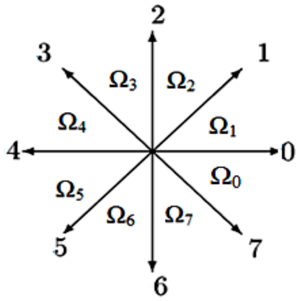

Fig. 6: 8 elementary vectors with corresponding angle areas

- The Directional Feature Vector of each pixel is obtained by decomposing its Gradient Vector into components along these standard direction planes (shown in *Figure 5 & 6).* If a gradient direction lies between two standard directions, it is decomposed into two components along those two directions resulting into generation of directional feature vector A_{ij} as shown in *Figure 7.*

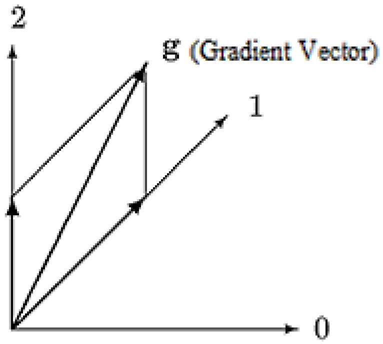

Fig. 7: Decomposition of gradient direction.

- The numeral image is divided into 81(9 horizontal × 9 vertical) blocks. The strength of the gradient is accumulated separately in each of 8 directions, in each block, to produce 81 local spectra of direction.We define the Feature vector *F* with dimensionality 8×9×9 = 648.
- The spatial resolution is reduced from 9×9 to 5×5 by down sampling every two horizontal and every two vertical blocks with 5 × 5 Gaussian Filter to produce a feature vector of size 200 (5 horizontal, 5 vertical, 8 directional resolution). So, ***F** is* condensed into a 200-dimensional vector **GFV** (**G**radient **F**eature **V**ector) with a Gaussian filter **W**(shown in *Figure 8*).

- The variable transformation (y = $x^{0.4}$) is applied to feature set to make the distribution of the features Gaussian-like.

The 5 × 5 Gaussian Filter used in this step is the high cut filter to reduce the aliasing due to the down sampling as done in paper [24].

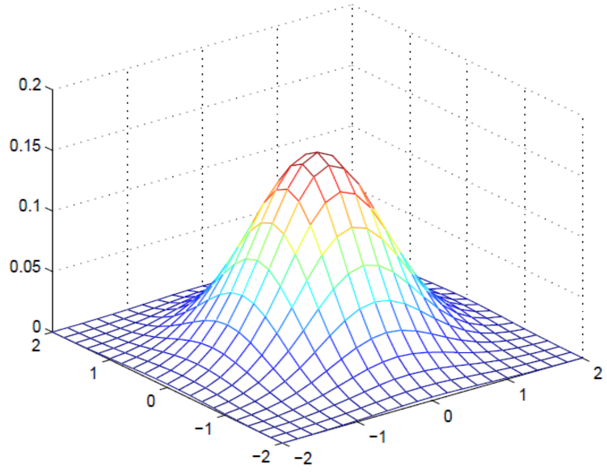

Fig. 8: 2-Dimensional Representation of Gaussian filter W

3.3 Stage 3: Classification using SVM

Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. The standard SVM classifier takes the set of input data and predicts to classify them in one of the only two distinct classes. SVM classifier is trained by a given set of training data and a model is prepared to classify test data based upon this model. For multiclass classification problem, we decompose multiclass problem into multiple binary class problems, and we design suitable combined multiple binary SVM classifiers. Our problem is also a multiclass problem in which the classifier has to differentiate between 10 different classes of Devanagari numerals. We obtained such multiclass SVM classifier tool LIBSVM available at [25]. A practical guide for SVM and its implementation is available at [26], [27].

According to how all the samples can be classified in different classes with appropriate margin, different types of kernel in SVM classifier are used. Commonly used kernels are:Linear kernel, Polynomial kernel, Gaussian Radial Basis Function (RBF) and Sigmoid (hyperbolic tangent). The effectiveness of SVM depends on kernel used, kernel parameters and soft margin or penalty parameter C.

The common choice is RBF kernel, which has a single parameter gamma (g orγ). We also have selected RBF kernel for our experiment. Radial Basis Function (RBF) kernel, denoted as

$$K(x_i,x_j) \equiv exp(-\gamma \parallel x_i - x_j \parallel^c)$$

Best combination of (C, γ) for optimal result is obtained by grid search by exponentially growing sequence of (C, γ) and then each combination is cross validated. Finally the combination giving highest cross validation accuracy is selected as optimal.

Ink-fold cross validation we first divide the training set into k equal subsets. Then one subset is used to test by classifier trained by other remaining k-1 subsets.

4. EXPERIMENTAL RESULTS

In order to classify the handwritten numeral and evaluate the performance of the technique, we have carried out the experiment by setting parameters gamma (γ), and cost parameter(C).

The training set of Devanagari Handwritten Numerals provided by ISI, Kolkata contains 18784 samples was used to determine the best combination of γ and C. We experimented with different-different values of the gamma (γ) as shown in *Table 2* and obtained maximum recognition accuracy of 99.607% at gamma (γ) equal to 0.004 and 0.005and cost (C) = 500.*Figure 9* shows the graphical representation of data tabulated in *Table 2*. This combination of γ =0.004 and C=500 was then tested on entire dataset of 22546 images with 5-fold cross validation and 99.60% recognition rate was obtained.

Table 2: The recognition accuracy of Devanagari Numeralsat different-2 values of gamma (γ)

S. No.	**5 –folds, Dataset size = 22,546, Cost(C)=500,**	
	Gamma (γ)	**Recognition Accuracy**
1	0.0008	99.463%
2	0.001	99.494%
3	0.002	99.527%
4	0.003	99.579%
5	**0.004**	**99.607 %**
6	**0.005**	**99.607%**
7	0.006	99.562%
8	0.007	99.533%
9	0.008	99.481 %
10	0.009	99.454 %
11	0.01	99.427 %
12	0.05	99.392 %

While observing the results at other values of parameter C, it is analyzed that increasing the value of C irrespective of any change in γ slightly decreases the recognition rate, but after a certain increment normally after 64 at higher values of C the recognition rate becomes stable. In contrast, the recognition rate always changes with the change in γ. The optimized results are obtained at C=500 and γ value in range 0.002 – 0.006.

4.1 Comparison with earlier approaches

Table 3 illustrates past work done in the recognition of Handwritten Devanagari Numerals and comparison of our approach with all of them.

5. CONCLUSION & FUTURE SCOPE

In the literature, many techniques for recognition of Devanagari Handwritten Numerals have been suggested. In this paper an effort is made towards recognition of Devanagari Numerals and obtained recognition accuracy of 99.60% on the database, provided by ISI, Kolkata.Use of Gradient as Feature Extraction Technique requires only a few simple arithmetic operations per image pixel which makes them suitable for real-time applications. Our approach is flexible in a way that the same algorithms can be used, without modification, for feature extraction in a variety of

OCR problems. These include recognition from handwritten, machine-printed, grayscale/binary and low-resolution images. Due to its logical simplicity, ease of use and high recognition rate, Gradient Features should be used for recognition purposes in other Indian Scripts like Gurmukhi, Malayalam etc. where not much research is conducted for their recognition. More research work should be conducted in using Gradient Features in combination with other feature extraction techniques & different-2 classifiers in order to improve recognition accuracy.

6. ACKNOWLEDGMENT

The authors like to thank Indian Statistical Institute (ISI) Kolkata for their sincere help towards the provision of standard numeral dataset for our experiment.

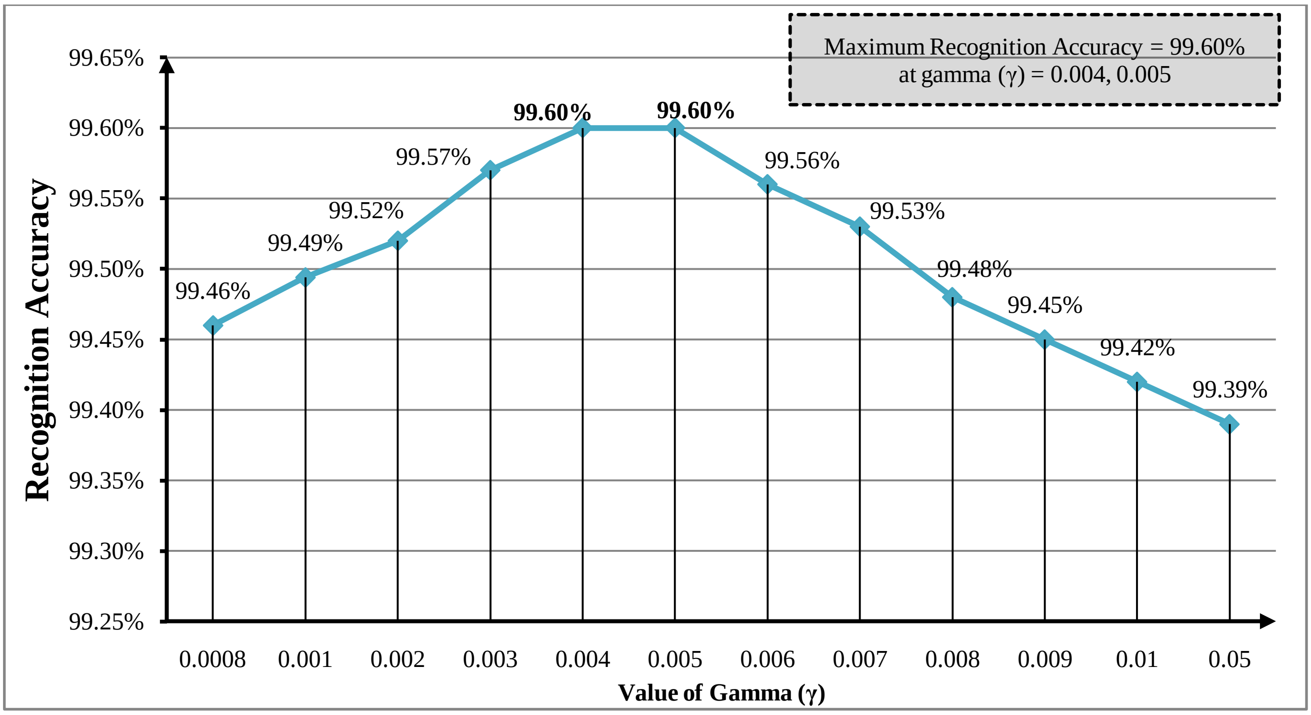

Figure 9: Graph shows the recognition accuracies of Devanagari Numeralsat different values of gamma

Table 3:Comparison of Devanagari Numeral recognition with earlier approaches

S.No	Method	Data Set(Size)	Accuracy (%)
1	Bajaj et al. [5]	2,460	89.68
2	Ramteke et al. [6]	2,000	92
3	Lakshmi et al. [16]	9,800	94.25
4	Hanmandlu et al. [10]	3,500	95
5	U.Bhattacharya et al. [7]	16274	95.64
6	Hanmandlu et al. [11]	3,500	96
7	Elnagar et al. [13]	1,500	96
8	Garain et al. [14]	12,000	96
9	Basu et al. [17]	3,000	97.00
10	Rajput et al. [12]	13,000	97.85
11	Pal et al. [15]	23,340	98.41
12	Sharma et al. [18]	22,546	98.86
13	Mahesh Jangid et al. [19]	22,546	98.98
14	U.Bhattacharya et al. [9]	22,546	99.27
15	Patil et al. [3]	2,000	99.50
16	Pal et al. [8]	22,546	99.56
17	**Proposed Approach**	**22,546**	**99.60**

7. REFERENCES

[1] Ivind due trier, Anil Jain, torfiinnTaxt, "A feature extraction method for character recognition-A survey ", Pattern Recg, vol 29, No 4, pp-641-662, 1996.

[2] SandhyaArora, DebotoshBhattacharjee, Mita Na-sipuri, D. K. Basu, M. Kundu, " Recognition of Non-Compound Handwritten DevnagariCharac-ters using a Combination of MLP and Minimum Edit Distance", International Journal of Computer Science and Security (IJCSS), Volume (4): Issue-1 pp 107-120.

[3] P M Patil, T R Sontakke," Rotation, scale and trans-lation invariant handwritten Devanagari numeral character recognition using general fuzzy neural network", Pattern Recognition, Elsevier, 2007.

[4] G S Lehal, Nivedan Bhatt, "A Recognition System for Devnagri and English Handwritten Numerals", Proc. Of ICMI, 2000.

[5] Reena Bajaj, Lipika Day, SantanuChaudhari, "Devanagari Numeral Recognition by Combining De-cision of Multiple Connectionist Classifiers", Sadhana, Vol.27, Part-I, 59-72, 2002.

[6] R.J.Ramteke, S.C.Mehrotra, "Recognition Hand-written Devanagari Numerals", International jour-nal of Computer processing of Oriental languages, 2008.

[7] U. Bhattacharya, S. K. Parui, B. Shaw, K. Bhattacharya, "Neural Combination of ANN and HMM for Handwritten Devnagari Numeral Recognition".

[8] U. Pal, T. Wakabayashi, N. Sharma and F. Kimura, "Handwritten Numeral Recognition of Six Popular Indian Scripts", Proc. 9th ICDAR, Curitiba, Brazil, Vol.2 (2007), 749-753.

[9] U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals,"IEEE Trans. Pattern Anal. Mach. Intell. , vol. 31, no. 3, pp. 444–457, Mar. 2009.

[10] M. Hanmandlu and O. V. R. Murthy, "Fuzzy model based recognition of handwritten numerals," Pattern Recognit., vol. 40, pp. 1840–1854, 2007.

[11] M. Hanmandlu, A. V. Nath, A. C. Mishra, and V. K. Madasu, "Fuzzy model based recognition of handwritten hindi numerals using bacterial foraging," in Proc. Int. Conf. Comput. Inf. Sci., 2007, pp. 309–314.

[12] G.G.Rajput and S.M.Mali, "Fourier descriptor based isolated Marathi handwritten numeral recognition," Int. J. Comput. Appl., vol. 3, no. 4,pp. 9–13, 2010.

[13] A.Elnagar and S. Harous, "Recognition of handwritten Hindi numerals using structural descriptors," J. Exp. Theor. Artif. Intell. , vol. 15, no. 3, pp. 299–214, 2003.

[14] U. Garain, M. P. Chakraborty, and D. Dasgupta, "Recognition of hand-written Indic script digits using clonal selection algorithm," in Lecture Notes in Computer Science 4163 , H. Bersini and J. Carneiro, Eds. New York: Springer-Verlag, 2006, pp. 256–266.

[15] U. Pal, R. K. Roy, K. Roy, and F. Kimura, "Indian multi-script full pin-code string recognition for postal automation," in Proc. 10th Conf.Document Anal. Recognit., 2009, pp. 456–460.

[16] C. V. Lakshmi, R. Jain, and C. Patvardhan, "Handwritten Devnagari numerals recognition with higher accuracy," in Proc. Int. Conf. Comput.Intell. Multimedia Appl. , 2007, pp. 255–259.

[17] S. Basu, N. Das, R. Sarkar, M. Kundu, M. Nasipuri, and D. K. Basu,"A novel framework for automatic sorting of postal documents with multi-script address blocks," Pattern Recognit. , vol. 43, pp. 3507–3521, 2010.

[18] N. Sharma, U. Pal, F. Kimura, and S. Pal, "Recognition of offline hand-written Devnagari characters using quadratic classifier," in Proc. Indian Conf. Comput.Vis.Graph.Image Process. , 2006, pp. 805–816.

[19] Mahesh Jangid, RenuDhir and Rajneesh Rani,"A Novel Approach: Recognition of Devanagari Handwritten Numerals ", International Journal Of Electrical, Electronics And Computer Systems (Ijeecs) Volume 1, Issue 2, April 2011. ISSN: 2221-7258(Print) ISSN: 2221-7266.

[20] R. Jayadevan et. al "Offline Recognition of Devanagari Script: A Survey",IEEE Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 41, No. 6, November 2011.

[21] U. Bhattacharya and B.B. Chaudhuri, "Databases for Research on Recognition of Handwritten Char-acters of Indian Scripts," Proc. Eighth Int"l Conf. Document Analysis and Recognition (ICDAR "05), vol. 2, pp. 789-793, 2005.

[22] Cheng-Lin Liu, Kazuki Nakashima, Hiroshi Sako, Hiromichi Fujisawa, "Handwritten digit recognition: investigation of normalization and feature extraction techniques" Elseveir Pattern Recognition, 37 (2004) 265 – 279.

[23] Ashutosh Aggarwal et al. "Handwritten Devanagari Characters Recognition using Gradient Features" IJARCSSE, page 85-90, Vol. 2 Issue 5 May 2012.

[24] Akinoria Kawamura et. al. "Online Recognition of Freely handwritten Japaneese Characters Using Directional feature Densities" IEEE XPLore Digital Library 1992.

[25] Chih-Chung Chang and Chih-Jen Lin, LIBSVM: a library for support vector machines, 2001. Software available: http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[26] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin, "A Practical Guide to Support Vector Classification",[Online]. http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

[27] Chih-Chung Chang, and Chih-Jen Lin, "LIBSVM: A Library for Support Vector Machines", [Online]. http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.