International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.8, No.7 (2015), pp.41-50
http://dx.doi.org/10.14257/ijsip.2015.8.7.05

Handwritten Devanagari Numeral Recognition by Fusion of Classifiers

[1] Prabhanjan S and [2] R Dinesh

[1,] *Research Scholar, School of Engineering and Technology, Jain University, Bangalore*
[2,] *Research Supervisor, School of Engineering and Technology, Jain University, Bangalore*
prabhan_us@rediffmail.com, dr.dineshr@gmail.com

Abstract

The abstract is to Recognition of handwritten Devanagari numerals has many applications especially in the field of postal automation, document processing and so on. Due to its vast applications, many researchers are actively working towards development of effective and efficient hand written character/numeral recognition. Devanagari script is widely used script in Indian sub-continent, also devanagari script forms the basis for many other scripts in Indian sub-continent. In this paper, we have proposed a hybrid method to recognize handwritten devanagari numerals. The proposed method uses, stacking approach to fuse the confidence scores from four different classifiers viz., Naïve Bayes (NB), Instance Based Learner (IBK), Random Forest (RF), Sequential Minimal Optimization (SMO). Also, the proposed method extracts both local and global features from the handwritten numerals. In this work, we have used Fourier Descriptors as global shape feature. Whereas, the pixel density statistics from different zones of the numeral to describe the numerals locally. The proposed method has been tested on large set of handwritten numeral database and experimental results reveal that the proposed method yields the accuracy of 99.685%, which is the best accuracy reported so far for the datasets considered. Hence the proposed method outperforms contemporary algorithms.

Keywords: *Classifiers, Handwritten Numeral Recognition, Fourier Descriptor, Machine Learning, stacking*

1. Introduction

Optical character recognition (OCR) translates scanned images documents into its equivalent electronic form. There are many successful systems available for effectively perform the OCR. However, the problem becomes extremely complex for recognition of handwritten characters or numerals due to wide variations in hand writing. Cursive writing makes problem even worse, also often the handwritten characters/numerals are written continuously making the segmentation complex. Hence, the accuracy of the existing systems for recognizing handwritten numerals is still far from the acceptable level. Handwritten character/numeral recognition has many applications in the field of postal automation, automatic mail sorting, document classification etc.

Devanagari is the most widely used script in Indian sub-continent. More than 500 million Indians are using Devanagari script for writing and speaking. Devanagari script is used for writing Hindi, Sanskrit, Konkani and Nepali languages. Many OCR-systems have been reported for a printed Devanagari script with high recognition rate. However, very few research works have been reported for recognizing handwritten Devanagari script.

Recognition of handwritten Devanagari script is challenging due to the variations in the writing style for different people and also due to different moods. Hence the handwritten

ISSN: 2005-4254 IJSIP
Copyright ⓒ 2015 SERSC

Devanagari numeral recognition is still an active research area. In this paper, we have proposed a hybrid approach by first fusing local and global features at feature level and subsequently, the confidence scores of multiple classifiers are fused by approach of stacking. Stacking is historically one of the first ensemble learning methods, which combines several base classifiers that are absolutely different classes of machine learning methods, by means of a "Meta -Classifier" [15, 16].

2. Literature Review

In pattern recognition handwritten recognition is still a challenging task inspite of decades of research. Due to vast applications of handwritten character/numeral recognition, many classification algorithms have been proposed in the past to recognize character/numeral of various languages like English, Arabian, Roman, etc. [1] used normalized distance feature using a fuzzy model for recognition of handwritten Hindi numerals and they obtained recognition rate of 92.67%. Further, [2] proposed a method for recognition handwritten characters using MQDF based classifier using directional chain codes features for blocks in bounding boxes and they got 98.86 % accuracy for numerals and 80.36 % for characters. [3] Obtained an accuracy of 91.28% with multilayer perceptron (MLP) neural network using multiresolution features based on wavelet transforms. [4] Proposed a method for recognition of Devanagari characters based on regular expression, in their work, the characters were converted into a sequence and then geometrical properties of the characters were converted into a regular expression. The converted regular expressions were used for matching and the minimum edit distance was used to obtain the matching score. They obtained 82% recognition rate. [5] Have used ensemble of classifier for recognition of Devanagari characters. In their work they have adopted multi stage classification, in the first stage characters were classified based on the shirorekha (a top line on the devanagari character). In next stage classification is performed with different feature sets like Histogram, Profile, Edge based features, etc. They have obtained recognition rate of 94.2 %. [6] Implemented a system for recognition of Devanagari handwritten recognition using genetic algorithm based on diagonal features they have reported 85.78 % accuracy. [18] Proposed a multistage classification for recognition of unconstrained handwritten characters, first stage based on fuzzy inference system and second stage based on structural features using feed forward neural network and obtained 96.95% accuracy. [19] Implemented multistage neural network for recognition of numerals based on structural and geometric features and obtained detection rate of 93.17%.

From the above discussion it is clear that there are few methods available for recognition of handwritten devanagari characters/numerals. However, their recognition rates are still far from the acceptable levels. Also, most of the results reported in the existing papers were based on the very limited number of datasets and they are not evaluated on a standard datasets. There are no common datasets to evaluate methods on a common ground; thereby there was need for standard datasets. Recently, [6 and 7] have generated standard handwritten Devanagari datasets for isolated numerals. In this work, we have used the datasets generated by [6-7] to evaluate the proposed method.

From the discussion it is evident that there is a need for an effective and efficient approach for recognition of handwritten devanagri numeral system which produces acceptable recognition rate. In this paper, we have presented an approach based on multi-level fusion, one at feature level and another at classifier level to improve the overall recognition accuracy of handwritten devanagari numerals.

Rest of the paper is organized as follows. In section 3 we present the details of the proposed method, Experimental results were detailed in section 4, and finally concluding remarks are drawn in section 5.

 Copyright ⓒ 2015 SERSC

3. Proposed Scheme

In this section, we present the proposed approach for recognition of handwritten devanagari numerals. Figure 1 shows sample of hand written numerals from 0 thru 9.

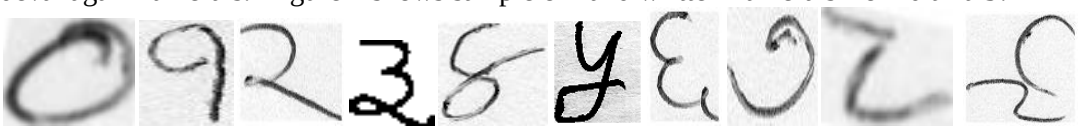

Figure 1. Sample Images

The overall architecture of the proposed system for handwritten devanagari numeral is shown in in Figure 2.

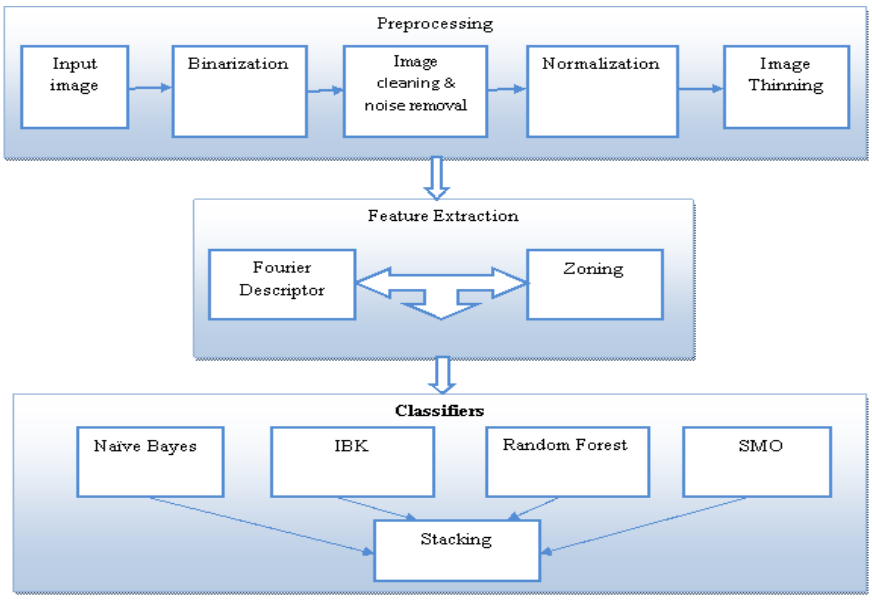

Figure 2. Proposed System Architecture

3.1. Preprocessing

3.1.1. Binarization: Generally, gray scale images will have several variations due to the nature of writing by different users, thickness of writing, smearing of ink on the paper, the tint of the paper etc. These variations pose challenges on recognition system and hence it is slightly simpler to process the bi-level image. In order to convert the gray scale image in to binary image we have employed thresholding approach. Initially, to get the optimal threshold we had employed automatic threshold using Otsu's method [20]. However, Otsu threshold did not yield good binarization, due to large variation. Hence, we have employed custom thresholding for the dataset considered. We found the threshold in the range 150 and 220 is ideal for the datasets; the range has been empirically derived. The sample binary image is shown in Figure 3 (b).

3.1.2. Image Cleaning and Noise Removal: Most of the imaging sensors having electro mechanical limitations and hence it is impossible to eliminate the noise at the source. Hence there is a need for an effective mechanism to handle the noise at preprocessing level. The most common type of noise occurs in the image is isolated dots caused by salt and pepper kind of noise. These isolated dots need to be removed to maximize the recognition rate. In this work, we have employed various image filtering techniques to remove the nose. Images are smoothed using linear filtering operation, further to eliminate the small, isolated

Copyright ⓒ 2015 SERSC

components and bridge the narrow opening in the characters, morphological opening and closing operations are performed. The sample output of preprocessing operation is shown in Figure 3 (c).

3.1.3. Image Normalization: Feature extraction is very vital step in any recognition system. Most of the existing classifiers require the features to be of uniform dimension (feature vector length should be same). In order to ensure that feature vectors are of the same length, we have normalized the images through the standard image resizing operation followed by bilinear interpolation. This operation not only ensures the feature dimension is fixed; also it makes feature extraction process simple and efficient. In this work, all images have been resized (normalized) to the standard size of 40x40. The value for the resizing has been fixed to 40x40 by empirical study. The sample normalized image is shown in Figure 3 (d)

3.1.4. Image Thinning: Thickness of the numeral stroke varies depending on the writer, kind of pen used and the resolution of the image while scanning. Hence making the algorithm width dependent, to make the algorithm independent of the above factors, it is essential to perform single pixel width, so that the above factors are compensated. For this purpose, in this work, we have performed the morphological thinning operation, to get single pixel width numeral. The sample output of the thinning operations is presented in Figure 3 (e).

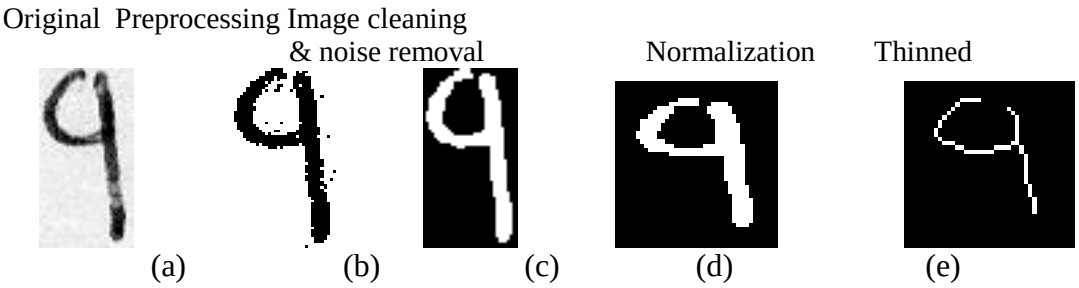

Figure 3. Preprocessing of Devanagari Numerals

The preprocessed images are subjected for feature extraction. The following subsection explain the details of feature extraction process.

3.2. Feature Extraction

Extracting suitable features to represent and describe the image is very vital step in any recognition system. In this research work, we have considered two sets of features. One is derived using Fourier Descriptors and second derived from Zonal features. Reason for selecting Fourier descriptor is to accommodate the global characteristics of numerals. Also, Fourier descriptors are very effective in representing the shape of numerals and widely used for shape representation [8-11]. On the other hand, Zonal features are effective in capturing the local variations of numerals. Local features are very essential to capture the finer variation with in numerals.

Generally, for character recognition applications, Fourier Descriptor has been applied to the boundary coordinates of the planar curve since character's boundaries is a closed curve, the sequence of (x, y) coordinates that specifies the curve is periodic. In the proposed work, for a given numeral, the boundary curve has been extracted from the thinned numeral image. Fourier transforms are applied to the sequence of complex numbers formed by x + iy, where x and y are the coordinate locations of the point on the boundary curve of the numeral. Similarly transformations are applied in sequence of complex number formed by y+ix. We have considered, the both x+iy and y+ix to take care of flip invariance. The Magnitude of these coefficients is used as features. 44 features used from x+iy coefficients and 14 features used from y+ix coefficients.

 Copyright ⓒ 2015 SERSC

Further, to extract the local features, we have employed zonal feature extraction on the numeral images. For this purpose, image has been divided into 16 blocks. For each block the total number of foreground pixels (pixels corresponding to numerals) is counted and the count is considered as a feature value for that corresponding block. Hence, a total of 16 zonal features are obtained. Finally, the Global features extracted using Fourier Descriptors are combined with the local features obtained by zonal feature analysis are fused together using the concatenation fusing. This has resulted in a total of, 75 features for a given numeral. This set of 75 features is used for both training and testing the classifier.

3.3. Classifiers

In the proposed research work we have used 5 different classifiers. All five classifiers are independently trained respective kernels are stored in the knowledge base. The classifiers used in this work are NB, IBK, RF, SMO and Stacking.
A brief overview of the used classifier is given below:

3.3.1. Naïve Bayes: A Naive Bayes classifier is a probabilistic classifier based on Bayes theorem. The Naïve Bayes classifier can be trained very efficiently in supervising the learning setting. Naïve Bayes is a conditional model relates a dependent class variable C with a small number of outcomes or classes, conditional, or several feature variables F1 through FN P (C | F1, F2,….. Fn).

Parameter estimation based on maximum likelihood. Advantage of Naïve Bayes is a small amount of data required for training to estimate the parameters for classification because it assumes independence of variables and computes only the variances of the variables for each class. The Naive Bayes classifier combines this model with a maximum aposteriori (MAP) decision rule. Bayes classifier is the function classify defined as follows :

$$Classify\ (F1 \ldots Fn) = \underset{C}{argmax}\ p(C=c) \prod_1^n p(Fi - fi | C = c) \quad (1)$$

Where C is a dependent class variable with a small number of outcomes or classes, conditional on several feature variables F1 through FN [12]

3.3.2. Instance Based Learner: Instance-based learning compares new problem instances with instances stored in memory during training. Instance based learning is a lazy learning, its complexity grows with data because it constructs a hypothesis directly from training instances. The advantage of this model is that it adapt to previously unseen data. K-Nearest Neighbor (K-NN) algorithm is an instance-based learning algorithm. Its classification is non-parametric method and regression that predicts class memberships based on the K closest training samples in the feature space. If K = 1, then the object is assigned to the single nearest neighbor class. This algorithm is sensitive to the local structure of the data. The training samples in K-NN are vectors in a multidimensional feature space, each with a class label. During training phase feature vectors and class labels of training samples are stored in memory. During the classification phase, K is a user-defined constant, and an unlabeled vector is classified by assigning the label which is most common among the K training samples nearest to that query point. Euclidean distance metric is used as a distance metric for continuous variables. The Euclidean distance between two vectors is as defined below where *a* and *a'* are vectors of the same length .

$$d\ (a, a') = \sqrt{(a1 - a1')^2 + \cdots . + (an - an)^2} \quad (2)$$

The accuracy of the K-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance. Much research effort has been put into selecting or scaling features to improve classification [12]. The K-NN instance based learner is implemented as IBK in Weka tool.

Copyright ⓒ 2015 SERSC

3.3.3. Random Forest: Random Forest is an ensemble learning method for classification and regression. It constructs a number of decision trees at training time and outputting the class that is the mode of the classes.

It is a combination of tree predictors where each tree depends on the values of a random vector sampled independently with the same distribution of all trees in the forest. The basic principle is that a group of "weak learners" can come together to form a "strong learner". Introducing the right kind of randomness makes them accurate classifiers.

Single decision trees often have high variance or high bias. Random Forests attempts to mitigate the problems of high variance and high bias by averaging to find a natural balance between the two extremes [13]. Random Forest can be viewed as weighted neighborhood. This model built from training set $\{(x_{i,}y_i)\}\underset{i=1}{\overset{n}{\smile}}$ that make predictions $\hat{y}$ for new points x' by looking at the "neighborhood" of the point, given by weight function W:

$$\hat{y} = \sum_{i=1}^{n} W(x_{i,}x^{'})y_i \tag{3}$$

$W(x_{i,}x^{'})$ is the non-negative weight of the i^{th} training point relative to the new point x'. For any particular x', the weights must sum to one. $W(x_{i,}x^{'})$ is the fraction of the training data that falls into the same leaf as x'[15].

RF averages the predictions of a set of *my* trees with individual weight functions, its predictions are

$$\hat{y} = \frac{1}{m}\sum_{j=1}^{m}\sum_{i=1}^{n} W_j\,(x_{i,}x')y_i \;= \sum_{i=1}^{n}\left(\frac{1}{m}\;\sum_{j=1}^{m} W_j\,(x_{i\,},x')\right)\,y_i \tag{4}$$

3.3.4. Sequential Minimal Optimization: SMO solves support vector machine (SVM) quadratic programming (QP). SMO breaks the QP problem into a QP sub problem. Consider a binary classification problem with a dataset $(x_1, y_1)\ldots (x_n, y_n)$ where x_i is an input vector and y_i ϵ {-1,+1} is a binary label is binary label. A soft-margin support vector machine is trained by solving a quadratic programming problem, which is expressed as follows

$$max_{\alpha}\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\;\sum_{i=1}^{n}\sum_{j=1}^{n} y_i\,y_j\,K(x_i\,,x_j)\alpha_i\alpha_{j,} \tag{5}$$
$$n$$

Subject to:

$0\leq \alpha_i \leq C$, for i=1,2,….n,

$$\sum_{i=1}^{n} y_{\,i}\;\alpha_i$$

Where C is an SVM hyper parameter and K (x_i, x_j) is the kernel function supplied by the user; and the variables α_i are Lagrange multipliers.

SMO breaks the problem into a series of smallest possible sub-problems, which are then solved analytically. Because of the linear equality constraint involving the Lagrange multipliers α_i, the smallest possible problem involves two such multipliers α1 and α2, for any two multipliers α1 and α2, the constraints are reduced to:

$0\leq\alpha1,\alpha2\leq C$

$$y1\alpha1+y2\alpha2=k,$$

And this reduced problem can be solved analytically: one needs to find a minimum of a one-dimensional quadratic function. k Is the negative of the sum over the rest of the term in the equality constraint, which is fixed in each iteration [14]. In our work we have used Pearson VII universal kernel (PUK) based on Ustun *et al.* [17].

 Copyright ⓒ 2015 SERSC

3.3.5. Stacking: Stacking is historically one of the first ensemble learning methods. It combines several base classifiers, which can belong to absolutely different classes of machine learning methods, by means of a "Meta -Classifier" that takes as its inputs the output values of the base classifiers [15, 16]. In this paper, stacking is concerned with combining NB,IBK,RF and SMO algorithmic as base classifiers on a devanagari numeral dataset S, which consists of fusion of Fourier descriptor and zoning feature vectors (xi) and their classifications (yi). In the first phase, a set of base-level classifiers NB, IBK, RF and SMO is generated, where Ci=Li(S). In the second phase, meta-level logic boost is learned that combines the outputs of base-level classifiers. To generate a training set for learning the meta-level classifier 10 fold cross validation procedure is applied. We apply each of NB,IBK,RF and SMO learning algorithms to almost the entire dataset, leaving subset of size one-tenth of the original dataset are left out for testing: $\forall i = 1,...,n : \forall k = 1,...,N : C_i^k = Lk(S - si)$. We then use the learned classifiers to generate predictions for si=$\hat{y}_i^k \; C_k^i(xi)$. The meta-level dataset consists of the form $\left(\hat{y}_i^1, \ldots, \hat{y}_i^n\right), yi)$ where the features are the predictions of the NB,IBK,RF and SMO.

4. Experimental Results

In this section, we present the detailed experimental analysis of the proposed approach. The dataset considered for the experimentation includes both standard data sets released by ISI Kolkata, CPR12 and our own dataset generated through multiple users [6] & [7]. Overall, the number of images considered for this experimentation is 24,782 images with non-uniform distribution among images. We have developed the algorithms using Matlab and Weka.

The evaluation results of our cross validation experiments are shown in Table 1. Table 1 shows the classification accuracy across the four classifiers: NB, IBK, RF, SMO and Stacking by combining NB, IBK, RF and SMO as base classifiers with Logic Boost as Meta-classifier.

It can be observed from Table-1, Stacking outperforms the other four classifiers on the datasets with a better classification accuracy of 99.68%. However, the RF and SMO results were statistically similar to that stacking, where they were both statistically better than NB and IBK. This indicates that the SMO and RF are well suited for Devanagari numeral recognition

Table 1. Comparative Results for Various Classifiers

Classifiers	Data Size	Accuracy for 10 fold cross validation
NB	24782	95.004
IBK	24782	98.70
RF	24782	99.17
SMO	24782	99.56
Stacking	24782	99.6853

In order to demonstrate the generalization and stability about the classifier, we also carried out the experimentation with varying number of training samples and its recognition response. The ROC curves from the above experimentation have been given in the Figure 5.

Copyright ⓒ 2015 SERSC

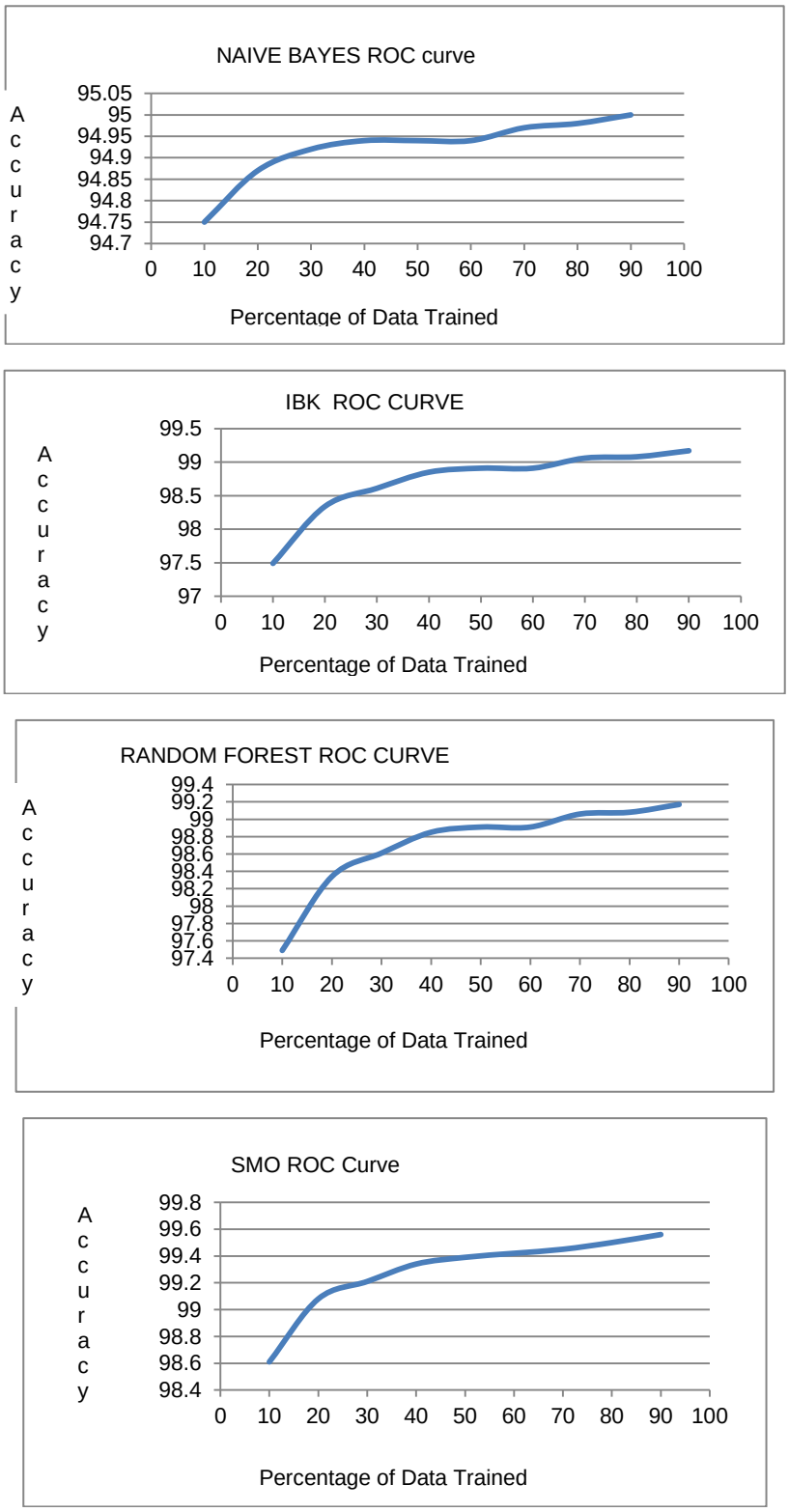

Figure 5. Classifiers ROC Curve

The combined performance of all classifiers with varying training dataset and its recognition rate is presented in Figure 6.

Copyright ⓒ 2015 SERSC

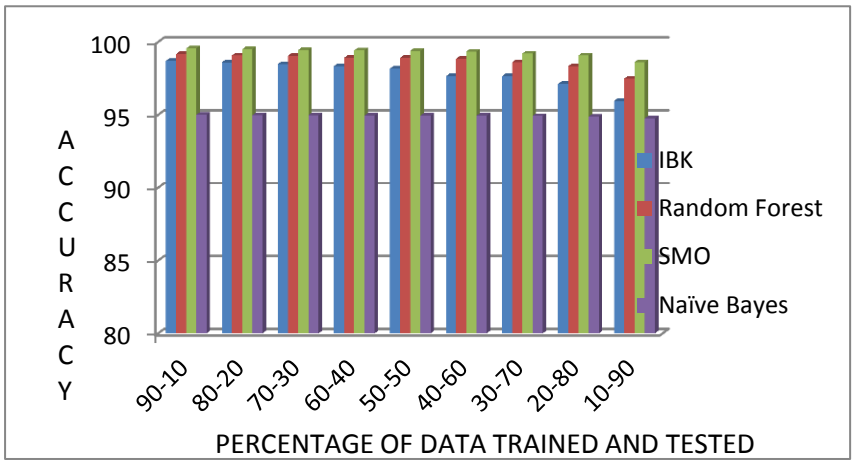

Figure 6. Combined Performance of All Classifiers

From the above experimentation, it is clear that the proposed method yields very good results overall. Specifically, combination of NB, IBK, SMO and RF gives best result among all other classifiers.

In order to establish the superiority of the proposed method over other existing contemporary algorithms, we have compared the results of the proposed method with other existing methods. Table 2 shows Comparison result. It is evident from the comparison results that the proposed method outperform other existing methods.

Table 2. Shows Comparision of Numerals Results by Reseachers

Sl NO	Method proposed by	Data size	Accuracy Obtained
1	Hanmandlu and Murthy [1]	4750	92.67%
2	U. Bhattacharya[3]	22700	91.28%
3	Satish Kumar[5]	11000	94.2 %
4	Ved Prakash Agnihotri[6]	1000	85.78%
5	S. Pal[2]	11270	98.86%
6	**Proposed Method**	**24782**	**99.68%**

Condition based selection of classifier is a most interesting work and it yields best possible classifier selection based on feature set. However, condition based selection of classifier is beyond the scope of this paper. We have kept that work as a future work to select the classifiers sets based on the initial output of the classifier telling whether; the given character is a numeral, vowel or consonant.

5. Conclusions

In this paper, we have presented a novel approach for recognition of handwritten Devanagari numerals. In this work, we have captured both Global and Local features through Fourier Descriptors and Zoning. Subsequently, we have trained multiple classifiers namely NB, IBK, RF & SMO. Finally, the scores from all classifiers were combined to get the final verdict using stacking. The extensive experimentation revealed that the proposed method yielded 99.685% recognition rate, which was superior when compared to most of the existing methods. Also the number of features used in the proposed methods was very less and easy to compute.

Copyright ⓒ 2015 SERSC

References

[1] O. V. Ramana Murthy and M. Hanmandlu, "Fuzzy Model Based Recognition of Handwritten Hindi Numerals", in Intl.Conf. On Cognition Recognition, (**2005**), pp. 490-496.
[2] N. Sharma, U. Pal, F. Kimura, and S. Pal, "Recognition of Off-Line Handwritten Devanagari Characters Using Quadratic Classifier," in ICVGIP 2006, LNCS 4338, (**2006**), pp. 805 – 816.
[3] B. B. Chaudhuri, R. Ghosh and M. Ghosh, U. Bhattacharya(2005), "On Recognition of Handwritten Devnagari Numerals, "In Proc. of the Workshop on Learning Algorithms for Pattern Recognition (in conjunction with the 18th Australian Joint Conference on Artificial Intelligence)",Sydney (**2005**), pp. 1-7.
[4] P. S. Deshpande, L. Malik, and S. Arora, "Fine classification and Recognition of Handwritten Devnagri characters with regular expressions and minimum edit distance method", Journal of Computers, vol. 3, no. 5, May (**2008**).
[5] S. Kumar, "A Three Tier Scheme for Devanagari Hand-printed Character Recognition," in *World Congress on Nature & Biologically Inspired Computing*, (**2009**), pp. 1016-1021
[6] V. P. Agnihotri, "Offline Handwritten Devanagari Script Recognition,"*Information Technology and Computer Science, MECS Press*, vol. V-4, (**2012**), pp. 37-42, July.
[7] R. Kumar and K. K. Ravulakollu, "Handwritten Devangari Digit Recogntion:Benchmarking on new dataset", Journal of Theoretical and Applied Information Technology, 28th February (2014), vol. 60 ISSN: 1992-8645
[8] C. S. Lin and C. L. Hwang, "*New Forms of Shape Invariants from Elliptic Fourier Descriptors*", Pattern Recognition", vol. 20, no. 5, (**1987**), pp. 535-545.
[9] C. T. Zahn and R. Z. Roskies, "*Fourier descriptors for plane closed curves*", IEEE Transactions on Computers, (**1972**).
[10] H. Kauppinen, T. Seppänen and M. Pietikäinen, "An Experimental Comparison of Autoregressive and Fourier-Based Descriptors in 2D Shape Classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 2, (**1995**), pp. 201-207.
[11] E. Persoon and K. S. Fu, "Shape discrimination using Fourier descriptors", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 3, (**1986**), pp. 388-397.
[12] E. O. Omidiora, I. A. Adeyanju and O. D. Fenwa, "Comparison of Machine Learning Classifiers for Recognition of Online and Offline handwritten Digits",(2013) Vol.4, No.13, ISSN 2222-1719
[13] Wikipedia, "Random Forest", Available: http:// http://en.wikipedia.org/wiki /Random_forest (March 10, 2015).
[14] Wikipedia, "Sequential Minimal Optimization", Available: http://en.wikipedia.org/wiki/ Sequential_minimal_optimization (August 14, 2014).
[15] D. H. Wolpert, "Stacked generalization. Neural Network", vol. 5, (**1992**), pp. 241-259.
[16] A. K. Seewald, "How to Make Stacking Better and Faster While Also Taking Care of an Unknown Weakness", In: Nineteenth International Conference on Machine Learning, (**2002**), pp. 554-561.
[17] K. Pearson, "Contributions to mathematical theory of evolution: II", Skew variation in homogeneous material
[18] S. Shelke and S. Apte, "A fuzzy based classification scheme for unconstrained handwritten Devanagari character recognition", In Proc. of International Conference Information & Computing Technology (ICCICT- 2015) on (**2015**), January 15-17.
[19] V. J. Dongre and H. M. Vijay, "Devanagari offline handwritten numeral and character recognition using multiple features and neural network classifier", In Proc. of International Conference on Computing for Sustainable Global Development (INDIACom) (**2015**) March 11-13.
[20] N. Otsu, "A threshold selection method from gray-level histograms", Systems, Man and Cybernetics, IEEE Transactions on vol. 9, no. 1, 0018-9472, (**1979**) January, pp. 62-66.

 Copyright ⓒ 2015 SERSC