

---

# Ensembling: Model of histogram of oriented gradient based handwritten devanagari character recognition system

S. P. Deore<sup>1,2,\*</sup>, A. Pravin<sup>1</sup>

1. Département of Computer Science & Engineering,  
Sathyabama Institute of Science and Technology, Chennai, India

2. Department of Computer Engineering, M.E.S. College of Engineering,  
Pune, S.P. Pune University, Maharashtra, India

anudeore9@gmail.com

---

*ABSTRACT.* It is very easy for the humans to recognize a character after gaining some basic knowledge about the language, but when it comes to the computer, the computer cannot recognize the character until and unless it is trained properly. We need to transfer the knowledge to the computer so that the computer can automatically recognize the character based on the knowledge provided. The Devanagari script is used as a base language for many different languages. This paper presents the system which recognize the handwritten devanagari isolated characters using ensembling of classifiers. In this paper HWCR system to recognize the Devanagari character using ensembling of classifiers is proposed. Recognition of Devanagari character is done in three main steps. The first step is pre-processing of the character image in which binarization and complementary of an image are performed. The second important step is feature extraction for which it uses histogram of oriented gradient as a feature. Third step is classification in which three different classifiers are used and their performance are analysed. The three classifiers are SVM, K-NN and NN. Results of these classifiers are combined together and given to the ensembler who classifies the class label based on maximum voting method. The average recognition rate is achieved by the proposed HWCR system is 88.13% using ensembling.

*RÉSUMÉ.* Il est très facile pour les humains de reconnaître un caractère après avoir acquis quelques connaissances de base sur la langue, mais quand il s'agit de l'ordinateur, il ne peut pas reconnaître le caractère à moins qu'il n'a pas été correctement formé. Nous devons transférer les connaissances sur l'ordinateur afin que celui-ci puisse reconnaître automatiquement le caractère en fonction des connaissances fournies. Le script Devanagari est utilisé comme langue de base pour de nombreuses langues. Cet article présente le système qui reconnaît les caractères isolés devanagari manuscrits en utilisant l'ensemble des classificateurs. Dans cet article, nous proposons un système HWCR pour reconnaître le caractère Devanagari en utilisant l'ensemble de classificateurs. La reconnaissance du caractère devanagari se fait en trois étapes principales. La première étape est le prétraitement de l'image de caractère dans lequel la binarisation et la complémentarité d'une image sont

effectuées. La deuxième étape importante est l'extraction de caractéristiques pour lesquelles il utilise histogramme de gradient orienté comme une caractéristique. La troisième étape est la classification dans laquelle trois classificateurs différents sont utilisés et leurs performances sont analysées. Les trois classificateurs sont SVM, K-NN et NN. Les résultats de ces classificateurs sont combinés ensemble et donnés à l'ensembleur qui classe l'étiquette de classe en fonction du mode de vote maximal. Le taux de reconnaissance moyen obtenu par le système HWCR proposé est de 88,13% avec l'ensembleur.

**KEYWORDS:** devanagari character, K-NN, SVM, NN, HWCR.

**MOTS-CLÉS:** Caractère devanagari, K-NN, SVM, NN, HWCR.

DOI:10.3166/TS.34.7-20 © 2017 Lavoisier

## 1. Introduction

Handwritten Character Recognition (HWCR) is a very important space of analysis in image processing area due to its many important applications in banks, post offices, etc. Main objective of this area is to provide automation and to reduce the gap between human being and machine. Online HW CR and Offline HWCR are two important types of HWCR system. Online HCR means accepting the input directly on electronic devices such as digitizer, PDA, or sensors and it uses timestamp process. Offline HCR means accepting inputs images from a paper by optical scanning using scanner and converting it into computer format. As per literature review it is observed that the character recognition of Indian scripts has a great demand.

Handwritten character recognition is very complex due to many reasons. Firstly, shape similarity between characters. Secondly, handwriting of a person is different in various circumstances. Thirdly, noise present while collecting data. With this we also have to consider large variations in stroke primitives due to different handwriting styles. In English language there are only 26 characters, however Indian scripts encompass vowels and consonants total further as Compound Characters therefore recognition of Indian languages is difficult as compared to English language. There are 12 vowels, 36 consonants and 10 numbers in Devanagari script. The work done for Devanagari scripts can be adopted for different languages such as Marathi, Hindi, etc. India's national language Hindi is world's third most popular language after Chinese and English (Jayadevan *et al.*, 2011). The challenge here is to create a software system that will recognize handwritten Devanagari characters as these characters can have different font size, font style and shape.

In character recognition, the very first step is pre-processing of an image. Various task carried out in pre-processing stage some are binarization, image complement and thinning. This step removes noise from an image. In the process of handwriting recognition, it is important to identify correct feature set so next step is feature extraction which extract unique features from input images which will be later on given to the classifiers for recognition.

Trier *et al.* (1996) explained different feature extraction methods for different character images like binary, thinned, gray scale etc. Boukharouba and Bennia proposed new method Freeman chain code for feature extraction which does not

required feature normalization process. For Persian numerals recognition they have used SVM classifier and achieved good recognition rate. Efficient feature extraction methods always helped to improve the accuracy of the algorithm and to distinguish each class. Chacko *et al.* (2011) used wavelet energy feature and extreme learning machine on Malayalam characters in which a new and robust parameter called wavelet energy is derived using wavelet transformation. Thaiklang and Seresangtakul proposed a system which recognized Isarn Dharma characters in which a system using histogram of oriented gradient (HOG) and zoning features is implemented. SVM and ANN two classifiers are used for classifications which are giving them outstanding result on HOG. Koerich (2003) carried out different experiments to determine which classification techniques are suitable for unconstrained English handwritten characters. They have performed various experiments on uppercase and lowercase letters as separate classes and merged classes i.e. those letters having similar shapes and got improved results. Shi and Wang (2015), used Stroke detector-based models for recognizing numeric characters and they achieved 82.6% accuracy. Recognition of numeric postal codes for a multi-script is presented in reference (Basu *et al.*, 2010) in which MLP and SVM classifiers have been on Latin, Devanagari, Roman, Bangala dataset and achieved average 96.72% recognition rate. In recent years, deep learning approach is getting more popularity due to its various successful applications to computer vision, pattern recognition, speech recognition, natural language processing, and recommendation systems. It is also suitable for big data analysis. Perez and Wang explained different methods to reduce over fitting problem in which different data augmentation techniques are elaborated to increase database size. Zanchettin *et al.* (2012) presented hybrid KNN-SVM method for cursive character recognition. Using combination of KNN-SVM they observed significant improvement in recognition result. Hybrid approach of KNN is defined in (Yu *et al.*, 2016), which classify data using local and global information of query samples. Hanmandlu and Murthy (2007) presented fuzzy model based on exponential membership functions on Handwritten Hindi and English numerals whereas Dietterich, given a review on three methods of ensembling: Error-correcting output coding, Bagging, and boosting and explained how ensembling of different classifiers improve overall recognition rate compared with single classifier. Various ensemble methods have been presented for word (Dietterich, 2000; Unter and Bunke, 2003) and numeral recognition (Huang and Suen, 1995; Oliveira *et al.*, 2006; Ye *et al.*, 2002) and concluded that ensembling method really improve recognition rate. Sahare *et al.* explained different algorithms for character segmentation and SVM, KNN used for recognition of multilingual Indian document images of Latin and Devanagari scripts in which both printed and handwritten images are used for experiments and achieved good recognition rate for both segmentation and recognition. Jangid *et al.* explored a new approach of layer wise training of deep convolutional neural network with adaptive gradient methods. A page-level handwritten document image dataset of 11 official Indic scripts are presented in which report the benchmark results for handwritten script identification. In paper pixel distribution features of numerical characters are calculated and then based on these features all numerals are classified in appropriate groups. Author compares this approach with histogram matching and obtained improved results.

Ramya et al. implemented SVM classifier to test efficiency on Kannada handwritten characters and then recognition rates are analysed for different SVM kernels. Mane et al. presented a Customized Convolutional Neural Network (CCNN) which learn the features automatically and predict the class of Marathi numerals from a large data-set of 80000 numerals. Additionally, visualization of the intermediate CCNN layers is presented which explains the dynamics of the presented network.

From above discussion it is seen that there are several character recognition techniques such as SVM, NN, CNN etc. Each of these techniques has its own advantages and disadvantages. The drawbacks and loopholes of using individual techniques can be overcome by ensembled together. Ensembling the classifiers is an emerging approach which binds together the advantages of different classification techniques. By fusion of classifiers this proposed work aims to improve and maximise the rate of recognition. Bristow and Lucey (2014) proved that how SVM trained on HOG features improve recognition rate. They also explained the factors like preserving second order statistics and locality of interactions which helps in improving recognition. Experiments are performed on expression recognition and pedestrian detection tasks. Hence in this work it is decided to use HOG-SVM combination for Devanagari Handwritten character recognition to observe its effects and really to get the improved recognition result. With this two more classifiers K-NN and NN are explored. To see the combined effect of these three classifiers in this paper an ensemble model working on the basis of maximum voting method is implemented. Mainly ensemble technique is employed to boost the accuracy rate of the HWCR system. This paper is organized as follows: Section 2 introduces the state of art literature review, limitation and scope of existing research, Section 2 highlights on Devanagari script, whereas section 3 focus on the propose architecture method. Section 4 present obtained experimental results and concluding remark is given in section 5.

## 2. Devanagari script

One of the most popular scripts in India called Devanagari, the approach of inscribing this script is left to right direction. Devanagari script is the base of many Indian languages such as Hindi, Marathi, Sanskrit, Nepali etc. This script comes under Brahmic family of scripts. Figure 1 shows the printed Devanagari Characters which consist of 12 vowels and 36 consonants. Figure 2 shows sample of one person handwritten Devanagari characters.

अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ	अं	अः
---	---	---	---	---	---	---	---	---	---	----	----

(a) Consonant characters

क	ख	ग	घ	ङ	च	छ	ज	झ	ञ	ट	ठ
ड	ढ	ण	त	थ	द	ध	न	प	फ	ब	भ
म	य	र	ल	व	स	ष	श	ह	क्ष	त्र	ज्ञ

(b) vowel characters

Figure 1. Printed samples

### 3. Architecture of propose method

In this a HWCR system is implemented which is used to recognize 12 vowels and 36 consonants of Handwritten Devanagari character to isolate unconstrained character images as a complete image. Figure 3 depicts architecture of proposed HWCR system. It has two phases: Training and Testing (Recognition).

#### 3.1. Database collection

The dataset used to train our system is taken from Kaggle Datasets (Acharya *et al.*, 2016), which is available online. These images are captured and already scanned using a scanner. The image of this dataset has black background colour and white foreground colour. In this work, database consists of total 10560 images. As there are in all 48 classes (12 vowels and 36 consonants) in Devanagari script, we took 220 images of each 48 classes for the database creation. Both SVM and K-NN are given 170×48 images for training purpose and 50×48 images for testing purpose. NN is provided with 160×48 images for training, 50×48 images for testing and 10×48 for validation purpose. The recognition rate is calculated on the basis of testing dataset.

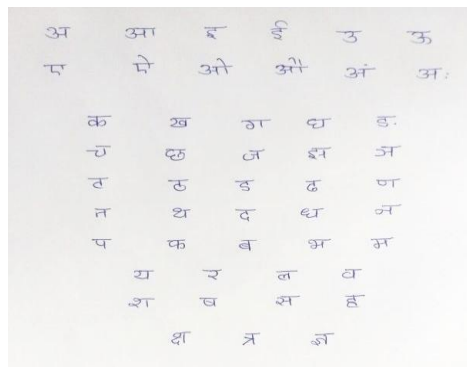


Figure 2. Handwritten samples of Devanagari script

**3.2. Pre-processing**

Pre-processing is the next important stage after database collection. Pre-processing performs some task in order to make the image more accurate and feasible for further processes. Pre-processing is carried out to remove the unwanted noisy signals from the image which helps in increasing recognition rate. Main goal of pre-processing is to enhance the image without any disturbance to its originality. The resizing and binarization operation are performed on image.

**3.2.1. Resizing**

All the images are resized into a standard size which is of 32×32.

**3.2.2. Binarization**

It is a process of converting the RGB image given as input into the grey scale image. This image is then transformed into binary image (0 & 1). The usefulness of an image is maintained using binarization. This process is of two phase analysis in which the pixel with intensity less than 1 has black background, otherwise it has white background. Figure 4(a) shows original input image and Figure 4(b) shows binarized pre-processed image.

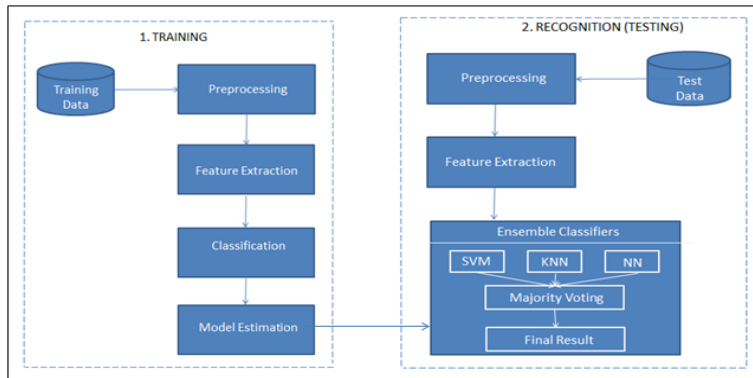


Figure 3. Architecture of proposed HWCR system



(a) Original input image (b) Binarized Pre-processed image

Figure 4. Original input and binarized pre-processed image

### 3.3. Feature extraction

To improve the efficiency of the algorithm set of features are extracted from images. Histogram of oriented gradients (HOG) features is used to extract shape information. In this method x and y gradients of an image is used. The x- gradients corresponds with the vertical lines and y-gradient corresponds with the horizontal line. A gradient image means an image where a lot of non-essential information is skipped (e.g. constant coloured background). For each pixel in the image, the gradient has a magnitude and direction. The histogram is an array of 9 bins whose value corresponds to angles from 0 to 160. The bin in the histogram is selected based on the direction and the value to be stored in the bin is selected from the magnitude corresponding to the direction. Figure5 (a) shows pre-processed image and Figure5 (b) shows extracted HOG features of the image.

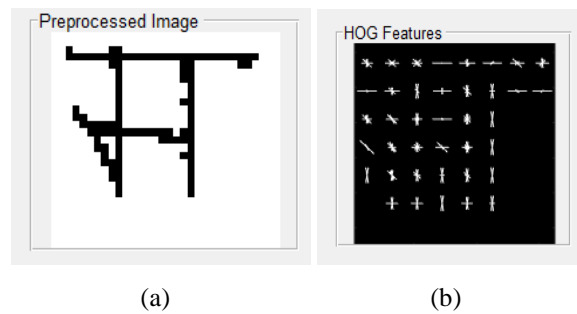


Figure 5. Pre-processed image and it's HOG feature

### 3.4. Classification

This is the final phase of character recognition system. The proposed system recognizes the character using K-NN, SVM and NN classifiers separately as well as Ensembling is done to improve and maximise the rate of recognition using advantages of different classification techniques.

#### 3.4.1. SVM

Support Vector Machine (SVM) is one of the popular supervised learning algorithm used for classification which classifies the data in best possible way. It is a maximum margin classifier because it forms the widest margin that separates the two groups. The distance between the support vectors and the hyperplane are as far as possible. Each data item is plotted in high dimensional feature space. So formally SVM builds a hyperplane or number of hyperplanes which can be used for different tasks. Hyperplane is used to split the data into two classes. In case of non-linear classification, SVM uses kernel function which maps the given input data into higher dimensional space so that it becomes possible to perform separation using

lines instead of fitting non-linear curves to the data. Linear classification can be easily done by the SVM model.

Originally SVM is binary class classification algorithm which give results in form of 'yes' or 'no'. To generate multiclass SVM there is a need to implement several binary SVMs. One-vs-rest and one-vs-one are two known methods used for a k-class problem, k, and  $k(k-1)/2$  binary SVM classifiers are performed. One-vs-rest method is used to implement multiclass SVM for our proposed system. The SVM Model is trained by giving 170 images per class for training and 50 per images class for testing.

### 3.4.2. K-NN

K-NN is one of the simplest classification and regression algorithm. The KNN algorithm doesn't really learn anything. It is a lazy learning algorithm because it does not create a model instead it classifies the new data based on the training data set. Given N training vector, K-NN algorithm identifies the k nearest neighbours of the input data, regardless of labels. For classification of unknown data it checks its k-nearest neighbours from the training data and finally distance is computed using Euclidean distance formula. Based upon the value of k, the characters are classified into different classes. k=1 gives better result as compared to other values of K. For higher dimensional data, weighted K-NN can be used. The equation for distance (Zanchettin *et al.*, 2012) is:

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

$$d(x, y) = \sum_{i=1}^n |x_i - y_i|$$

$$d(x, y) = \sqrt[p]{\sum_{i=1}^n (x_i - y_i)^2}$$

Where x is the feature vector of the input data that we want to find, y is a data of the training database, n the feature vector size and p is a constant (Zanchettin *et al.*, 2012). The K-NN Model is trained by giving 170 images per class for training and 50 per images class for testing.

### 3.4.3. Neural network

Artificial Neural networks are the most powerful learning model. Artificial Neural networks is an interconnected group of nodes. It consists of number of neurons that are implemented on the basis of how the neurons of the human brain works. Each of the neurons represents a node in NN. These nodes are connected by directed links and organized into different number of layers. Input, hidden and output are three layers used in NN. Data are given through the input layer. Input layer process data using initial weight and passes to hidden neurons. Then hidden layer apply activation function to generate new output. Each output unit sums its weighted input signals and generate final output. Final output is compared with



target pattern and error information is calculated. Using back propagation error information passed back to hidden layer and weights are adjusted to get desired output. Our model specifications are:

- 
1. Input :
    - Training : 160 images per class
    - Testing : 50 images per class
    - Validation : 10 images per class
  2. Number of Hidden Layers : 1
  3. Number of Hidden Layer Neurons :70
  4. Number of epochs: 107
  5. Activation function of Hidden layer: Tangent Sigmoid
  6. Activation function of Output layer: Linear
  7. Error Function used : Mean square error
- 

#### 3.4.4. Ensemble model

An ensembleing technique is collection of similar and dissimilar classifiers whose outputs are combined to give new model to classify the test data. Ensembleing improves the overall accuracy and it makes classifier robust. Assume the ensembleing of  $n$  identical or different classifiers  $(f_1, f_2, \dots, f_n)$ . If all the classifiers are identical, then result of ensembleing is identical to the result of individual classifier. If all the classifiers are different then the correct result will be achieved by majority voting method. Thus fusion of dissimilar classifiers gives better performance than the individual classifiers. So, ensembleing can be performed by using different classifiers, which are discussed above. Ensembleing method is used to improve the recognition rate of the character image. It combines the results of all the classifiers used and then predicts the class label of the character using the maximum voting strategy. Voting method requires parallel architecture. It is non adaptive in nature. Figure 6 shows the proposed ensembleing model:

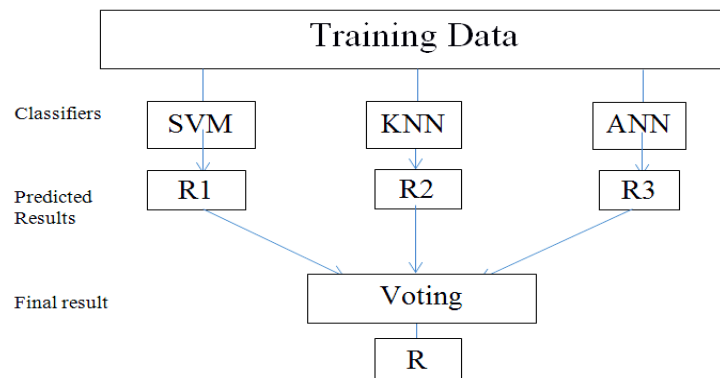


Figure 6. Ensembleing model

#### 4. Experimental results

MATLAB image processing tool is used for implementation. We have a dataset of 10560 Devanagari character images where each 48 characters have 220 images. 170 images are given for training of SVM and KNN while 50 images are given for testing purpose. For NN 160 images are given for training, 50 for testing and 10 for validation purpose. First the dataset is pre-processed performing task such as binarization and resizing of an image. The results of binary and thinned images for  $[8 \times 8]$  cell size are shown in the Table1. We also performed skeletonization (thinned) operation and experienced that thinning pre-processing operation is not better option for HOG feature extraction method which gives us less recognition rate as compared to without thinning binarized image due to gradients of an image plays very important role for feature extraction and magnitude of gradients is large around edges of images compared with flat area. Where ever change in intensity values of an image magnitude of gradients also affects. So when we performed thinning operation on an image it affect on higher intensity values actually it neglect higher intensity values. For SVM, KNN and NN it is reduced by 4.34%, 5.16% and 0.8% respectively. Hence we only did binarization in pre-processing and omitted the skeletonization task. Table 1 shows the recognition rate with and without thinning operation.

*Table 1. Thinning operation result*

Classifier	Without Thinned	Thinned
SVM	87.38 %	83.04 %
KNN	84.08 %	78.92 %
NN	82 %	81.2%

After pre-processing, HOG feature extraction method was carried out which extracted 324 features of each image in the dataset using  $[8 \times 8]$  cell size. We also extracted features using  $[2 \times 2]$  and  $[4 \times 4]$  cell size. We observed that with the cell size  $[8 \times 8]$  recognition rate improved by 1% approximately as shown in Table2. Pre-processed image size is  $32 \times 32$ , on that image  $[8 \times 8]$  cell working nicely which extract enough features for recognition. These feature sets are then given to the three classifiers for training purpose. The accuracy results for different cell sizes are shown in the Table 2.

*Table 2. Results of different classifiers using different cell size*

Classifier	$[2 \times 2]$	$[4 \times 4]$	$[8 \times 8]$
SVM	85.42%	86.58%	87.38%
KNN	82.54%	84.08%	85.21%
NN	79.6%	82.4%	82%

Figure 7 represent recognition rate of each 1 to 48 class characters obtained for SVM, KNN and NN classifiers. Class 1 contain 220 images of Devanagari character अ, class 2 contain आ, so on. Our data set consist of 220 images of each 48 class. Comparatively dataset size is limited. From Figure6 it is observed that SVM is working relatively good as compared with other two classifiers. For characters like ई,ए,ऐ,अं,क,च,झ SVM giving more that 95% recognition accuracy. For characters like अं,च,श KNN giving 100% accuracy. For character फ़ the recognition rate is low with all three classifiers. NN classifier required large amount of data for learning, hence its recognition rate is less as compared with SVM and KNN. But NN is fast learning algorithm and when our data size increase it will give improved recognition rate. From Figure7 it is also seen that graph of three classifier is non-linear so that one can combined these three classifiers to create single model called as ensembling model. Figure6 shows ensembling model where output of each classifier is combined and using majority voting algorithm final output is generated. Majority voting is simplest and popular ensembling method used for classification. Table 3 shows the obtained improved results compared to individual classifiers shown in Table 1. The confusion matrix is used as performance measure to calculate the recognition rate of individual classifiers as well as for ensembling model.

Some characters are confused due to shape similarity among them. To see that confusion matrix is calculated. Table 4 shows some sample characters with its confused characters corresponding to the classifier.



Figure 7. SVM, KNN, NN classifiers recognition rate

Table 3. Ensembling model result on different cell size

Ensembling Model	[2×2]	[4×4]	[8×8]
SVM+K-NN+NN	86.58%	87.75%	88.13%

In Devanagari script many similar shaped characters are there so that errors are occur while recognition. Figure 8 shows different samples of Devanagari

Handwritten characters. It is observed that for some characters there exist variations in each sample due different handwriting styles and their number of strokes.



Figure 8. Samples of handwritten Devanagari characters by five writers

Table 4. Confusion matrix for some characters

Alphabet	SVM	KNN	NN
आ	अ,अं,भ,म	अ,क,ख	
ई	ब	अं,ब	ऐ,अं,ट,ह,ज्ञ
ए	र	स,र,इ	इ, र
ऐ	ऊ,र	इ	ट,र
ओ	औ,अः	औ,अं	ऊ,औ
औ	अ,ओ,अं	ओ,अं,अ	ओ,अं,ल
अं		ओ,औ,ज	ई,ओ,औ,ढ
अः	अ,भ,म	-	आ,न,भ,म
क	अ	झ,अ,फ	भ,व
ख	स,व,ज्ञ	ग	ढ,प,र,स
ट	ठ,द,र,व,स	ठ,ढ,द,र,त,थ,	द,ढ,र,व
ड	इ,ड	ड,ब,इ,ढ	ड,र,ष,ह
ढ	ख,छ,द,ब,क्ष	उ,ऊ,क,ख,ट,व	क्ष,द,ट,क
ण	प	ध,थ,प	अ,ग,ज,थ,प,फ
त	व	न,क्ष	ट,ल

## 5. Conclusion

Recognition of handwritten Indian languages is very tedious and difficult task due to the different styles used by different peoples. Devanagari script is used as base for various Indian languages. The proposed offline handwritten Devanagari character recognition system recognizes character or individual letters in Devanagari script using offline dataset. We used total of 10560 images of vowels and consonants. 170 images of each 48 classes (12 vowels and 36 consonants) were used for training of SVM and KNN. 50 images of each 48 classes were used for testing of SVM and KNN. ANN is trained using 160 images of each 48 classes and tested using 50 images of each class. In this paper HOG features is used for the recognition of handwritten Devanagari character. HOG on binary as well as thinned image is implemented. Three classifiers SVM, k-NN and NN are used whose results are then combined together to form ensemble model.

The result shows that - i) HOG works better in case of binary image as compared to the thinned image , ii) [8×8] cell size gives better result as compared to [2×2] and [4×4] cell size, iii) SVM with HOG gives more accuracy rate as compared to K-NN and ANN, iv) SVM with HOG recognition rate is 87.38 % , KNN with HOG gives 84.08 % and ANN with HOG gives 82 % accuracy rate, v) Ensemble model give the recognition rate of 88.13 % which is greater than the rate of each individual classifier.

### Acknowledgment

*Authors thank to Dr. S. H. Gawande, Professor, M.E.S. College of Engineering, Pune, India for continuous support and encouragement.*

## References

- Acharya S., Pant A. K., Gyawali P. K. (2016). Deep learning based large scale handwritten Devanagari character recognition. *IEEE Proc. of International conference on Software, Knowledge, Information Management and Applications (SKIMA2015), Kathmandu, Nepal*, pp. 1-6. <http://dx.doi.org/1109/SKIMA.2015.7400041>
- Basu S., Das N., Sarkar R., Kundu M., Nasipuri M., Basu D. K. (2010). A novel framework for automatic sorting of postal documents with multi-script address blocks. *International Journal on Pattern Recognition*, Vol. 43, No. 10, pp. 3507-3521. <http://dx.doi.org/1016/j.patcog.20>
- Bertolami R., Bunke H. (2008). Hidden Markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, Vol. 41, No. 41, pp. 3452-3460. <http://dx.doi.org/1016/j.patcog.2008.04.003>
- Bristow H., Lucey S. (2014). Why do linear SVMs trained on HOG features perform so well? *Cornell University Library*, pp. 1-8.
- Chacko B. P., Vimal Krishnan V. R., Raju G., Babu A. P. (2011). Handwritten character recognition using wavelet energy and extreme learning machine. *International Journal of*

- Machine Learning and Cybernetics*, Vol. 3, No. 2, pp 149-161.  
<http://dx.doi.org/1007/s13042-011-0049-5>
- Dietterich T. G. (2000). Ensemble methods in machine learning. in: multiple classifier systems (MCS 2000). *Lecture Notes in Computer Science*, Vol. 1857, pp. 1-15.  
[http://dx.doi.org/1007/3-540-45014-9\\_1](http://dx.doi.org/1007/3-540-45014-9_1)
- Hanmandlu M., Murthy O. V. R., (2007). Fuzzy model based recognition of handwritten numerals. *Pattern Recognition.*, Vol. 40, No. 6, pp. 1840-1854.  
<http://dx.doi.org/1016/j.patcog.2006.08.014>
- Huang Y., Suen C. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 17, No. 1, pp. 90-94.
- Jayadevan R., Kolhe S. R., Patil P. M., Pal U. (2011). Offline recognition of Devanagari script: A survey. *IEEE Transactions on Systems, Man, and Cybernetics-part C: Applications and Reviews*, Vol. 41, No. 6. pp. 782-796. <http://dx.doi.org/1109/TSMCC.20>
- Koerich A. L. (2003). Unconstrained handwritten character recognition using different classification strategies. *Proceedings of International Workshop on Artificial Neural Networks in Pattern Recognition*. <http://dx.doi.org/1109/ICIP.2005.1530112>
- Oliveira L. S., Morita M., Sabourin R. (2006). Feature selection for ensembles applied to handwriting recognition. *Document Analysis Recognition*, Vol. 8, No. 4, pp. 262-279.  
<http://dx.doi.org/1007/s10032-005-0013-6>
- Shi C. Z., Wang C. H. (2015). Stroke detector and structure based models for character recognition: A comparative study. *IEEE Transactions on Image Processing*, Vol. 24, No. 12, pp. 4952-4964. <http://dx.doi.org/1109/TIP.2015.2473105>
- Trier O. D., Jain A. K., Taxt T. (1996). Feature extraction methods for character recognition- A survey. *Pattern Recognition*, Vol. 29, No. 4, pp. 641-662. [http://dx.doi.org/1016/0031-3203\(95\)00118-2](http://dx.doi.org/1016/0031-3203(95)00118-2)
- Unter S. G., Bunke H. (2003). Ensembles of classifiers for handwritten word recognition. *Document Analysis Recognition*, Vol. 5, No. 4, pp. 224-232.  
<http://dx.doi.org/1007/s10032-002-0088-2>
- Ye X., Cheriet M., Suen C. Y. (2002). StrCombo: Combination of string recognizers. *Pattern Recognition Letters*, Vol. 23, No. 4, pp. 381-394. [http://dx.doi.org/1016/S0167-8655\(01\)00171-4](http://dx.doi.org/1016/S0167-8655(01)00171-4)
- Yu Z., Chen H., Liu J., You J., Leung H., Han G. (2016) Hybrid k-nearest neighbour classifier. *IEEE Transactions on Cybernetics*, Vol. 46, No. 6, pp. 1263-1275.  
<http://dx.doi.org/1109/TCYB.2015.2443857>
- Zanchettin C., Bezerra B. L. D., Azevedo W. W. (2012) A KNN-SVM hybrid model for cursive handwriting recognition. *IEEE World Congress on Computational Intelligence (WCCI 2012)*, Brisbane, QLD, Australia, pp. 843-850.  
<http://dx.doi.org/1109/IJCNN.2012.6252719>