# Offline Handwritten Devanagari Vowels Recognition using KNN Classifier

Rakesh Rathi,
Dept. of CS&IT
Govt. Engg. College, Ajmer, Rajasthan, India

Ravi Krishan Pandey & Vikas Chaturvedi , Dept. of CS&IT
Govt. Engg. College, Ajmer, Rajasthan, India

Mahesh Jangid,
Dept. of Comp. Sci. Manipal University, Jaipur, Rajasthan, India,

## ABSTRACT
The discussion in the paper is regarding to the recognition of handwritten Devanagari vowels by means of a classifier named as K-NN (K- Nearest Neighbour). Before applying classifier, feature extortion is accomplished for extracting the feature points (FP) i.e. also known as division points (DP). In this paper the feature extortion is perform through recursive sub division technique, which is first time implemented on Devanagari vowels. K-NN classifier is functioned for the learning and the testing phases, through which the recognition go ahead to the high performances in terms of recognition rate, pre-processing and classification speed. Authors tested the described approach using the ISI (Indian Statistical Institute), Kolkata's handwritten Devanagari vowels database containing 9191 samples, which is divided into 1:3 as testing and training samples respectively. In the recognition process using K-NN classifier 88 vowels are total wrongly identified out of 2281vowels. The recognition rate comes out to be 96.14%.

## General Terms
Offline Handwritten Character Recognition

## Keywords
OHCR (Offline Handwritten Character Recognition), K-NN (K- Nearest Neighbor), Recursive Sub Division (A Feature Mining Technique)

## 1. INTRODUCTION
In present scenario pattern recognition is going to be a very gigantic field, a numerous numbers of researchers, organizations and companies are working with the field. One of its implicational sub fields is handwriting recognition. Which can be further divided into two parts as a) online handwriting recognition and b) offline handwriting recognition. Online handwriting recognition is performing in real time i.e. detection is carried out at the same time when the user provides the input by writing on a surface of a gadget which is meant for the online detection. In offline handwriting recognition the inputs are handwritten documents, papers or any surface which can be scanned through a scanner machine. The scanned document afterwards departed for the pre-processing steps for making a perfect input for the recognition process. Which includes binarization process, morphological function (in this isolated objects are removed), smoothing and normalization. Then after a classifier is trained on the data come out from the pre-processing steps.

In this paper k-nearest neighbor algorithm (K-NN) is used as the technique of classification of objects based on closest training examples in the feature space. K-NN is a type of instance-based learning or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is sensitive to the local structure of the data. Nearest neighbor rules in effect compute the decision boundary in an implicit manner. It is also possible to compute the decision boundary itself explicitly, and to do so in an efficient manner so that the computational complexity is a function of the boundary complexity [1].The training examples are vectors in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples.

In the classification phase, k is a user-defined constant, and an unlabelled vector (a query or test point) is classified by assigning the label which is most frequent among the k training samples nearest to that query point. Usually Euclidean distance or Overlap metric (or Hamming distance) is used as the distance metric. Often, the classification accuracy of K-NN can be improved significantly if the distance metrics learned with specialized algorithms such as Large Margin Nearest Neighbor or Neighborhood components analysis.

The best choice of k depends upon the data; generally, larger values of k reduce the effect of noise on the classification, but make boundaries between classes less distinct. A good k can be selected by various heuristic techniques, for example, cross-validation. The special case where the class is predicted to be the class of the closest training sample (i.e. when k = 1) is called the nearest neighbor algorithm.

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. A particularly popular approach is the use of evolutionary algorithms to optimize feature scaling [2]. Another popular approach is to scale features by the mutual information of the training data with the training classes. In binary (two class) classification problems, it is helpful to choose k to be an odd number as this avoids tied votes. One popular way of choosing the empirically optimal k in this setting is via bootstrap method [3].

## 2. RELATED WORK
OCR is a very old idea among the several described in the past of the pattern recognition using computers. In recent time, Devanagari character recognition becomes the field of practical usage. During modern years, research toward Indian handwritten character identification is attained increased consideration although the initial research article on offline handwritten Devanagari characters identification was printed in year 1977. Many approaches have been proposed toward handwritten Devanagari numeral, character and word recognition in the past decade [4]. Two approaches are mainly used in handwritten character recognition. First approach is

based on segmentation method and the other is free of segmentation method (also known as holistic approach). For first approach, initially the segmentation of words into pseudo characters or characters have been done and afterwards recognized. As a result, the accomplishment of the identification module lies on the performance of the segmentation techniques. In second approach the whole word is treated as a sole unit and it recognizes exclusive of performing explicit segmentation [5].

A Noise removal of the document is also an important step toward the recognition. Bajaj et al. [6] used a median filtering-based approach for noise removal from the images of handwritten Devanagari characters. They considered advanced features based on the outline representations of each of the 4 frequency components (low–low, high–high, low–high and high–low) of wavelet-filtered picture. Bajaj et al. represented each handwritten devanagari numeral and character using three types of features: i) descriptive component features ii) moment features of left, right, upper, and lower profile curves and iii) density features. For handwriting recognition a neural network-based system is premeditated

Thinning-based features are also used in Devanagari handwritten character recognition. From the thinned images of handwritten Hindi characters, 3 dissimilar types of feature points, that is to say cross, end and branch, points are extracted first in [7].A syntactic representation (SR) of features is used for handwritten character recognition. This demonstration is synchronized next to the group of prototype SRs of handwritten characters for a probable match.

Ramteke and Mehrotra [8] have evaluated the performance of a variety of techniques which are based on the moment invariants on handwritten Devanagari characters. Features are extracted may based on image partition, moments, principal component axes, perturbed moments and correlation coefficient.

Sharma et al. [9] are the used directional chain-code information of contour points for the handwritten Devanagari characters for extracting features. Blocks are formed from the segmentation of bounding box of characters and a Chain-Code Histogram (CH) is obtained for each blocks based on the CH, they have used sixty four-D features for the recognition. Modified Quadratic Discriminate Function (MQDF) classifier has been used for Devanagari character recognition.

In [10], a method is proposed based on cubic spline interpolation for determining smooth and continuous edges in the images of handwritten Devanagari characters. Edge direction histogram features are used along with PCA for enhancing recognition accuracies of handwritten Devanagari characters.

For extracting the features, a box approach is proposed by Hanmandlu et al. [11], [12] for handwritten characters, which requires a spatial division of the numeral image into boxes. Hanmandlu et al. projected a model based on fuzzy set scheme for identification of the handwritten Devanagari characters by demonstrating them by the exponential membership functions, which act as the fuzzy model. Changing the exponential membership functions fixed to fuzzy sets does the recognition. Fuzzy sets are imitative from the features consists of normalized distances proposed using Box approach.

# 3. PROCEDURE

Around the 11th century AD Devanagari script is came out from the modifications in Brahmi script. In starting it was developed for writing Sanskrit language but afterwards it was adopted for writing numerous other languages for example Hindi, Gujrati, Nepali and Marathi. The perfect Devanagari Vowels are shown in table 1. Similarity between आ;इ,ई ;उ ,ऊ ;ए ,ऐ ; ओ, औ and अं ,अः respectively are very large.

**Table 1: Distribution of Vowels in Devanagari Database**

| Vowels | Training Set | Testing Set | Total |
|---|---|---|---|
| अ | 526 | 175 | 701 |
| आ | 535 | 177 | 712 |
| इ | 521 | 171 | 692 |
| ई | 518 | 170 | 688 |
| उ | 566 | 187 | 753 |
| ऊ | 555 | 180 | 735 |
| ऋ | 511 | 166 | 677 |
| ए | 528 | 175 | 703 |
| ऐ | 536 | 177 | 713 |
| ओ | 524 | 171 | 695 |
| औ | 534 | 177 | 711 |
| अं | 531 | 178 | 709 |
| अः | 525 | 177 | 702 |
| Total | 6910 | 2281 | 9191 |

## 3.1Pre-Processing

After getting, the database is pre-processed. Pre-processing deals with the procedure for the enhancement of contrast, removal of the noise and separating the regions whose glance indicate the possibility of vowel information. In this phase it is normalized and removal of all the redundancies errors as of the picture and sends to the subsequently phase. Figure 1 shows the pre processing of Devanagari vowel उ.

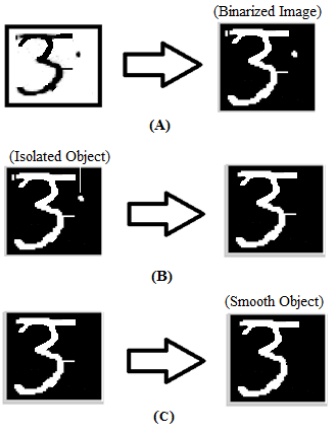


**Figure 1.(A) Devanagari Vowels उ after binarization process. (B) Devanagari Vowels उ after removing isolated object. (C) Devanagari Vowels उ after apply median filter.**

## 3.1 Feature Mining Algorithms

After pre-processing of Vowels, features of Vowels are mined. It's the core step of the method. In this step the vowels are classified on the basis of their features. Actually, the foremost dilemma in OCR system is the great variation within the shapes in a class of Vowels. This dissimilarity depends on the document noise, photometric effect, font styles, and poor image quality and documents skewness. The huge variation in the shapes makes it complicated to settle on the number of features, which are suitable earlier to the model building.

Suppose that im(x, y) is a handwritten Devanagari vowel image in which the foreground pixels are denoted by 1's and background pixels are denoted by 0's. Feature mining algorithm sub-divided the Vowel image recursively. At granularity level 0 the image is divided into four parts and gives a division point (DP) (x0, y0). The following algorithm shows that how x0 is calculated and likewise y0.

Algorithm: 1
Step 1: Let the input im(xmax, ymax) where the xmax and the ymax be the width and the height of the Vowel image.
Step 2: Let v0 [xmax] be the vertical projection of image (figure 2: b).
Step 3: Create v1 [2*xmax] array by filling a '0' before each element of v0 (figure 2: c).
Step 4: Find xq(index) in v1 that reduces the dissimilarity between the sums of the left partition [1, xq] and the right partition [xq, 2 *xmax] or left partition should be greater than right partition if not able to equally divide.
Step 5: x0=xq/2.
Step 6: if xq mod 2 = 0
Two sub-images will be [(1, 1), (x0, ymax) and (x0, 1), (xmax, ymax)] Else Two sub-images will be [(1, 1), (x0, ymax) and (x0+1, 1), (xmax,ymax)]

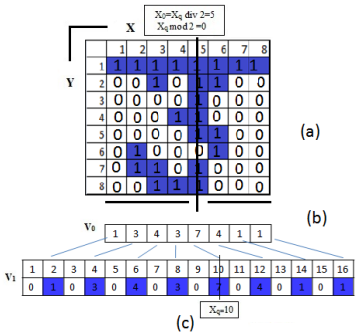


**Figure 2: Devanagari Vowel उ (a) Vertical division of an image array (xmax=8, ymax=8) (b) Vertical projection of image (c) v1 created from v0 to calculate xq**

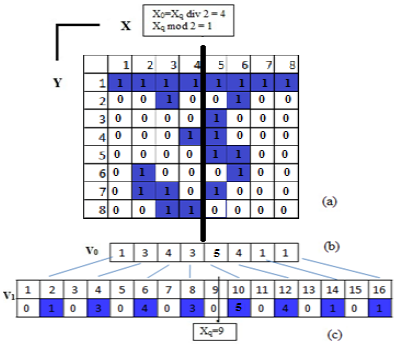


**Figure 3: Devanagari Vowel उ Example. where the xq mod 2 = 1**

From the figure 2 it is clear that the vertical division of handwritten Vowel image where the xq=10 and x0=5 and xq modulus 2 is 0 then the co-ordinates of two sub-images will be [(1,1),(5,8)] and [(5,1),(8,8)]. If the modulus of xq is not zero as in the case of an image which has xq =9 and xq modulus 2 = 1 is demonstrated in figure 3 and now the co-ordinates of two sub-images will be [(1,1),(4,8)] and [(5,1),(8,8)].

Algorithm: 2
This algorithm is just like the algorithm 1 but here we divided the image horizontally. This algorithm is applied after the algorithm 1 to find the division point (x0, y0) and portioned images.

Step 1: Let input im(xmax, ymax) where xmax and ymax be the width and the height of the Vowel image.

Step 2: Let v0 [ymax] be the horizontal projection of image (figure 4: b)

Step 3: Create v1 [2* ymax] array by inserting a '0' before each element of v0 (figure 4: c)

Step 4: Find yq(index) in v1 that minimizes the difference between the sums of the top partition [1, yq] and the bottom partition [yq, 2 *ymax] or top partition should be greater than bottom partition if not able to equally divide.

Step 5: y0=yq/2;

Step 6: if yq mod 2 = 0

If xq mod 2 = 0

Four sub-images will be [(1, 1), (x0, y0)] [(1, y0) (x0, ymax)] and [(x0, 1), (xmax, y0)][(x0,y0) , (xmax, ymax)]

Else

Four sub-images will be [(1, 1), (x0, y0)] [(1, y0) (x0, ymax)] and [(x0+1, 1), (xmax, y0)][(x0,y0) , (xmax,ymax)]

Else

ifxq mod 2 = 0

Four sub-images will be [(1, 1), (x0, y0)] [(1, y0+1) (x0, ymax)] and [(x0, 1), (xmax, y0)] [(x0, y0+1), (xmax, ymax)]

Else

Four sub-images will be [(1, 1), (x0, y0)] [(1, y0) (x0, ymax)] and [(x0+1, 1), (xmax, y0)] [(x0+1, y0+1), (xmax,ymax)]

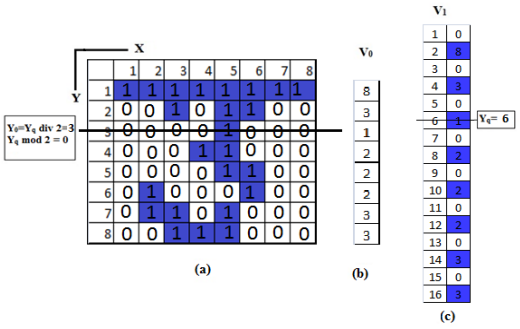


**Figure 4: Devanagari Vowel उ (a) Vertical and Horizontal division of an image array (xmax=8, ymax=8) (b) Horizontal projection of image (c) v1 created from v0 to calculate yq**

The number of sub-images, at the specified granularity level (L) will be 4(L+1). Let L=0 then the number of sub-images will be four and when the L=1 it will be 16. The number of DP (division point) equals to 4L (figure 5). At level L, the co-ordinates (xi, yj) of all DPs are stored as features. So for every L a 2*4L-dimensional feature vector is mined.

All feature vectors are scaled to (0, 1), by the help of normalized dimension value in our case it is 90. All the co-ordinates of feature vector are divided by 90.

f'= f/90 …….(1)

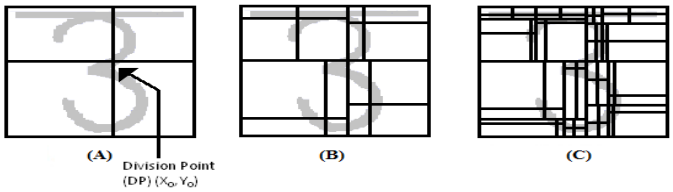


**Figure 5: Devanagari Handwritten Vowel उ segmentation at Level 0, 1, 2 shown in corresponding (A),(B) and (C)**

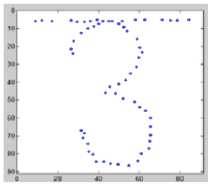

**Figure 6: Devanagari Vowel उ after applying algorithm (feature points)**

Above explanation is for the understanding of the algorithm. Experiments have been done on real image by applying same algorithm. The image is now normalised into 90 by 90. Figure 6 shows the feature points or Division Points getting after applying the algorithm. The feature vector values after scaling or dividing the feature points by 90.

## 3.2 Selection of Granularity Level

One question is raised here that what the granularity level or how much time we will be subdivide the image. For this firstly we have to done some experiment to select a right value for granularity level that is denoted by Lbest.In this phase, gradually increase the higher levels of granularity starting with level 1, features are mined and the recognition rate is calculated at particular level and drawn a graph (figure 7) that shows the level of granularity and the recognition rate. By the help of graph examine the highest recognition rate at corresponding level (Lbest).

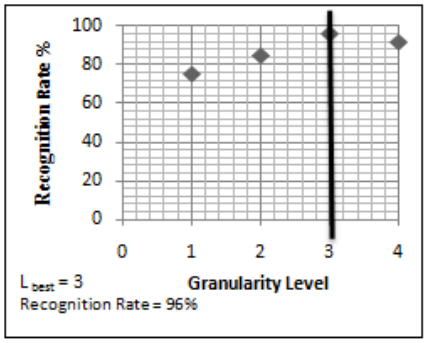


**Figure 7: Example finding the best level (Lbest)**

# 4. RESULT

Here the training dataset is a matrix of 6910*170 which shows that there are 6910 training samples and each sample have 170 feature values and the size of testing dataset is 2281*170 which shows there are 2281 testing samples and 170 features for each testing sample. K-NN classifier returns the class level of testing samples identified by the help of training dataset. Table 2 shows the no of wrongly identified vowel corresponding each vowel, 98 vowels are total wrongly identified vowels out of 2281. The recognition rate comes out to be 95.70%.

**Table 2: Number of wrongly identified or misclassified Vowels at default argument of K-NN**

| Vowel | अ | आ | इ | ई | उ | ऊ | ऋ | ए | ऐ | ओ | औ | अं | अः | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No of wrongly identified Vowels | 10 | 7 | 5 | 7 | 6 | 7 | 4 | 7 | 5 | 10 | 9 | 11 | 10 | 98 |

Now the value of K is changed from 1 to 3 and all the arguments remain same as above experiment and run the K-NN classifier again. Table 3 shows the results after changing the value of K. The total no of misclassified Vowels are reduced from 98 to 92 and the recognition rate is 95.97% which is improved from the previous experiment.

**Table 3: Number of wrongly identified or misclassified Vowels at K=3**

| Vowel | अ | आ | इ | ई | उ | ऊ | ऋ | ए | ऐ | ओ | औ | अं | अः | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No of wrongly identified Vowels | 9 | 6 | 5 | 7 | 6 | 7 | 4 | 7 | 5 | 9 | 8 | 10 | 9 | **92** |

In the next experiment, the distance metric is changed from Euclidean distance to Correlation Distance and remaining arguments keep same and repeat the experiment again. Table 4 shows the results after simulation. The recognition rate is 96.14%.

**Table 4: Number of wrongly identified or misclassified Vowels at K=3 and correlation**

| Vowel | अ | आ | इ | ई | उ | ऊ | ऋ | ए | ऐ | ओ | औ | अं | अः | Total |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No of wrongly identified Vowels | 9 | 6 | 5 | 6 | 6 | 6 | 4 | 6 | 5 | 9 | 8 | 9 | 9 | **88** |

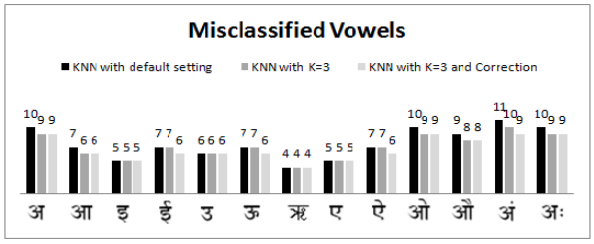


**Figure 8: The result for different values of k and Rule of K-NN**

So the highest recognition rate is 96.14 %. Figure 8 shows the combined result for different settings of K-NN classifier. Table 5 shows the recognition rate for K-NN classifier. For the recognition of Devanagari handwritten vowels, the K-NN with K =3 and correlation distance metric gives the highest recognition rate so we can say that it is better for recognition. The experiments are also done with different values of k and distance metric here we mentioned only a few with which we get higher accuracy.

**Table 5: Recognition rate for K-NN**

| Classifier | Misclassified Vowels | Recognition Rate |
|---|---|---|
| K-NN (Default) | 98 | 95.70 % |
| K-NN (with K=3) | 92 | 95.97 % |
| K-NN (with K=3 & correlation ) | 88 | 96.14% |

## 5. FUTURE SCOPE

Over the past three decades, many different methods have been explored by a large number of scientists to recognize Vowels. A variety of approaches have been proposed and tested by researchers in different parts of the world, including statistical methods, structural and syntactic methods and neural networks. No OCR in this world is 100% accurate till date. The recognition accuracy of handwritten Devanagari vowels proposed here can be further improved. The number of vowels set used here for training is reasonably enough but the recognition accuracy can be improved by taking more samples. In handwritten Devanagari vowel, some Devanagari vowels are similar in shape when they are written. So the recognition accuracy can be improved by solving such confusion by left or right profile or such other technique. Today's there are number of classifiers are used in recognition for example MLP (Multilayer Perceptron), MIL (Mirror Image Learning) and MQDF (Multi Quadratic Discriminate Function) etc. This classifier can be used to increase the recognition accuracy

## 6. ACKNOWLEDGMENTS

Authors would like to thanks to Indian Statistical Institute (ISI) Kolkata, for providing the extremely essential data base of handwritten Devanagari vowels for performing experiment on just a single mailing request.

## 7. REFERENCES

[1] Bremner D, Demaine E, Erickson J, Iacono J, Langerman S, Morin P, Toussaint G"Output-sensitive algorithms for computing nearest-neighbor decision boundaries". Discrete and Computational Geometry 33 (4): pp. 593–604, 2005.

[2] Nigsch F, Bender A, van Buuren B, Tissen J, Nigsch E, Mitchell JB, "Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization" Journal of Chemical Information and Modeling 46 (6): pp. 2412–2422, 2006.

[3] Hall P, Park BU, Samworth RJ, "Choice of neighbor order in nearest-neighbor classification". Annals of Statistics 36 (5): pp.2135–2152, 2008.

[4] U. Pal, T. Wakabayashi, F. Kimura, "Comparative Study of Devanagari Handwritten Character Recognition using Different Feature and Classifiers", 10th Intl. Conf. on Document Analysis and Recognition, pp. 1111-1115, 2009.

[5] R. Jayadevan, S.R. Kolhe, P.M. Patil, U. Pal, "Database development and recognition of handwritten devanagari legal amount words" Conference Proceeding: 10/2011;DOI:10.1109/ICDAR.2011.69 In proceeding of: 2011 International conference on Document Analysis and Recognition (ICDAR),

[6] Reena Bajaj, LipikaDey ,SantanuChaudhury,"Devanagari Vowel recognition by combining decision of multiple connectionist classifiers", Sadhana Vol. 27,Part 1, pp. 59–72, February 2002.

[7] A. Elnagar and S. Harous, "Recognition of handwritten Hindi Vowels using structural descriptors," Journal of Experimental & Theoretical Artificial Intelligence, Vol. 15, no. 3,pp. 299–214, 2003

[8] R. J. Ramteke, S. C. Mehrotra, "Recognition of Handwritten Devnagari Vowels", International Journal of Computer Processing of Oriental Languages, 2008.

[9] N. Sharma, U. Pal, F. Kimura, and S. Pal, "Recognition of offline handwritten Devnagari characters using quadratic classifier," in Proc. Indian Conference ComputerVision Graph. Image Process, pp. 805–816, 2006.

[10] C. V. Lakshmi, R. Jain, and C. Patvardhan, "Handwritten Devnagari Vowels recognition with higher accuracy," in Proc. International Conference Computer Intelligence Multimedia", pp. 255–259, 2007.

[11] M. Hanmandlu, A. V. Nath, A. C. Mishra, and V. K. Madasu, "Fuzzy model based recognition of handwritten hindi Vowels using bacterial foraging," in Proc. International Conference Computer Information Science,pp. 309–314, 2007.

[12] M. Hanmandlu, O.V. Ramana Murthy, Vamsi Krishna Madasu, "Fuzzy Model based recognition of handwritten Hindi characters", Digital Image Computing Techniques and Applications, pp. 7695-3067-IEEE. Feb-2007.