International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

HCR Using K-Means Clustering Algorithm

Meha Mathur[1], Anil Saroliya[2]

Amity School of Engineering & Technology
Amity University Rajasthan, India

Abstract: ***Hindi is a national language of India, there are about 300 million people in India who speak Hindi and write Devnagari script. A problem of Hindi character recognition is addressed and I propose a recognition mechanism based on k- means clustering. The large dataset of Hindi characters and their similarity makes the problem as there is no separation between the characters of texts written in Hindi as there is in English. K-means provides a natural degree of font independence and this is to reduce the size of the training database. In this paper I propose an OCR for Hindi characters, using K-means clustering. The major steps which are followed by a general OCR are preprocessing, character segmentation, feature extraction, classification and recognition. The paper introduce propose a two masks one is for horizontal projection and other for vertical projection of gray scale image to detect & eliminate shirorekha of word to decompose into individual characters from the words.***

Keywords: OCR, Hindi, Shrirorekha, Pre-processing, Segmentation, Feature Vector, Feature Extraction, Classification, and Devnagari.

1. Introduction

OCR finds wide applications as a telecommunication aid for the deaf people, postal address reorganization, and documents will directly processed foreign language recognition etc. However, there is not much reliable OCR software available for the Indian language Hindi (Devanagari), the third most spoken language in the world. The objective in this project is to design high performance OCR software for Devanagari script that can help in exploring future applications such as navigation, for example navigation used in traffics. To recognize the hindi character from the digital image, image segmentation is to be performed in computer vision. The automatic detection and recognition of hindi character or word in images, on the other hand, has been among the prime objectives of computer vision for several decades.

In order to determine the pattern K mean clustering algorithm is used. Clustering algorithm plays an important role in Hindi character identification. The currently best approach to recognition of Hindi character is K mean clustering algorithm.

In the pre-processing stage we have to select one image as per our interest which is colored one. We have to convert that image into gray scale image for better visualization of information stored in each pixel. Create two masks one is for horizontal projection and other for vertical projection of gray scale image to detect & eliminate shirorekha of word.

Now we have Segmented Image as our Binary Image. We have to crop each character from the Binary image word. Now we have cropped characters and have to find feature extraction of each character using K-Means clustering (For our Database this is a best suitable method except Contour Extraction, region growing....etc.).

Classification of the Hindi characters is performed by the Euclidean distance method. Where for this type of algorithm we estimate the future vectors for the each and every train data and store it in a data base. The Euclidean method is a simple method but powerful enough for the detection of the characters. For the case of test data we estimate the vectors and then these vectors are separated from the train data.

2. The Proposed Recognition Process

The proposed process of conversion of scanned image into a text document consist the following steps shown in the figure:

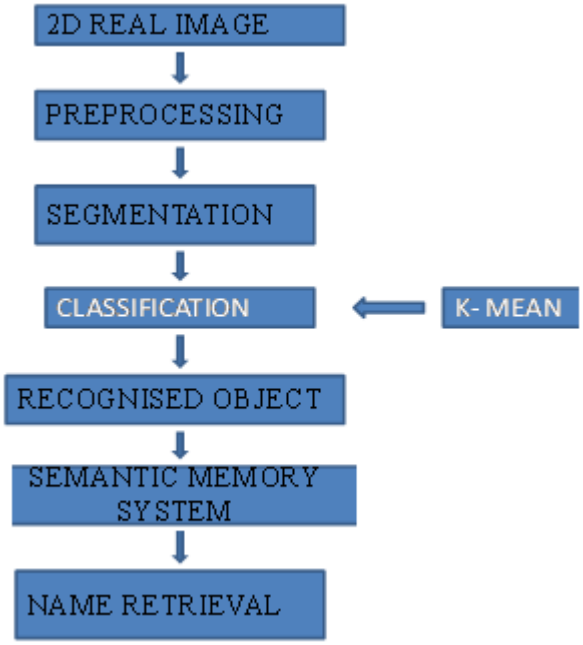

Figure 1: Flow chart for optical Hindi character recognition

2.1 Image Preprocessing

The scanned image was first converted from RGB scale to gray-scale. It was then splitted into individual character blocks using MATLAB script to obtain raw individual character samples. The following preprocessing and noise removal techniques were used on raw samples to obtain a clean dataset.

Median Filtering: Scanning process introduces irregularities such as speckle noise" and salt and pepper noise" in the output image. Median Filtering was employed, to remove

such effects, where each pixel was replaced by the median of the neighboring pixels.

Background Removal: To model the background noise due to scanning, a white page was scanned with the same scanner and this image was subtracted from each of the character images, hence eliminating background and any residual background noise; highlighting only the character sample.

Thresholding: To remove any residual irregularities and to increase image contrast, all pixel values above 200 were scaled up to 255. Also, all pixels lying at the boundary within a 50 pixel wide strip were scaled up to 255 to ensure a clean boundary.

A. Firstly, image is import from any supported graphics image file format. I take the image which is an RGB image.

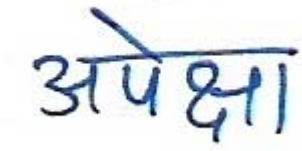

Figure 2: A Hindi word which is the original image taken as an input.

B. **Convert the image to gray scale image**
The input I had taken is a RGB image or a colored image I have to convert it into grayscale image for further processing. In computing a grayscale digital image in which the value of each pixel is consider as a single sample, which carries only information of intensity.

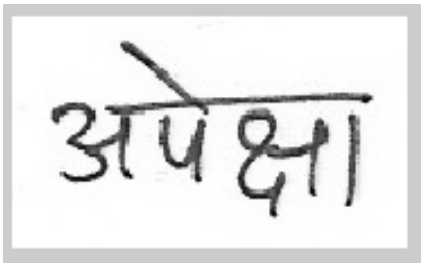

Figure 2: A gray scale image.

C. Defining image gradient
Image gradient is used for extract information from the image. Image gradient is of two types horizontal gradient and vertical gradient. These gradients are used for edge detection. Horizontal gradient detect horizontal edges and vertical gradient detect vertical edges. I use manual canny edge detector which uses image gradient for edge detection.

Noise reduction: Canny edge detector is susceptible to noise present in raw unprocessed image data; it will reduce the noise present in the input image. When we define vertical gradient mask, the matrix should be:

mask_V = [-1, 0, 1;
-2, 0, 2;
-1, 0, 1];

And for horizontal gradient mask, the matrix should be:
mask_H =[-1, -2, -1;
0, 0, 0;
1, 2, 1];

2.2 Shrirorekha Removing

After finding horizontal and vertical edges of the image the shirorekha can be removed. Process of removing shirorekha is done by identifying the rows with the maximum number of black pixels in a word. After locating the shirorekha, it is removed, i.e. converted to white pixels. Individual characters are separated from each zone by applying vertical scanning.

Figure 3: Detected shirorekha along vertical edge

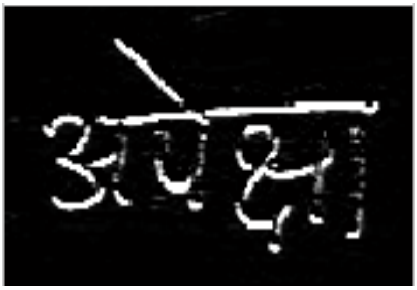

Figure 4: Detected shirorekha along horizontal edge

2.3 Convert gray image into binary image

Converting gray image into binary image using gray threshold. When we convert the grayscale image to the binary image the output image replaces all pixels in the input image with luminance greater than level with the value 1, i.e. white and replaces all other pixels with the value 0, i.e. black. Specify level on the range [0, 1]. This range is relative to the signal levels possible for the image's class. Therefore, a level value 0.5 is midway between black and white pixels, regardless of class. To find the level argument, we can use the function graythresh. If the input image is not a grayscale image, first convert it into grayscale image, and then converts this grayscale image to binary image by thresholding. Applying graythreshold to vertical and horizontal gradient.

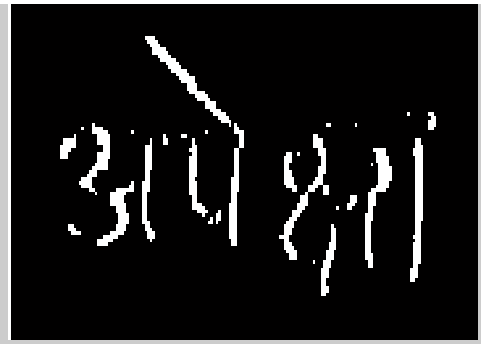

Figure 5: Binarized image vertical

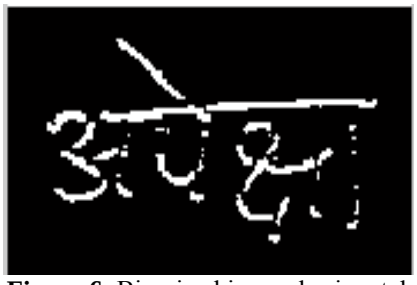

Figure 6: Binarized image horizontal

2.4 Image dilation

Dilation operation is one of the operations of morphological operations. Morphology is a set of image processing operations that process images based on shapes. To an input image, morphological operations apply a structuring element, creating the same size output image. In the output image the value of each pixel is based on a comparison of the corresponding pixel in the input image with its neighbors. Dilation adds pixels to the boundaries of objects in an image. The number of pixels added to the object in an image depends on the size and shape of the structuring element used to process the image. Dilation can be apply in both vertically and horizontally.

Figure 7: Dilated vertical image

Figure 8: Dilated horizontal image

2.5 Segmentation

Segmentation subdivides an image into its constituent objects. In segmentation, basic constituent of the word extract, which are characters. Segmentation phase is also crucial in contributing to this error due to touching characters, which cannot properly tackle by a classifier. Segmentation is the process of sub-diving the image into useful segments or components. We categorize the existing segmentation algorithm into region-based, data clustering, and edge-base segmentation. Region-based segmentation includes the seeded and unseeded region growing algorithms, and the fast scanning algorithm. All of them expand each region pixel by pixel based on their pixel value or quantized value so that each cluster has high positional relation. For data clustering, the concept of them is based on the whole image and considers the distance between each data.

Figure 9: Segmented image

2.6 Feature Extraction

A feature is an interesting part of an image, such as a corner, edge or line. Feature extraction enables to drive a set of feature vectors from a set of detected features. Feature extraction is most important step in developing a classification system. From the fig 9, I cropped a character which is the first character of the image.

Figure 10: Cropped character.

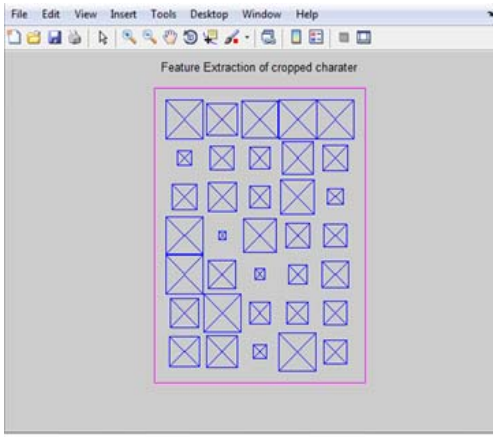

Figure 11: Feature extraction of cropped character

The feature extraction of a character is shown in 7x5 matrixes. The box contains pixels of the image and divided into the matrixes. The value of the character can be find by the matching the character with the ASCII value of character which is stored in the class library.

```
>> disp(char(2309:2350))
अआइईउऊऋऌऍऐऎऑओऒऔकखगघङचछजझञटठडढणतथदधनऩपफबभम
```

Figure 12: ASCII values of the Hindi characters.

Therefore we can find the value of 'a' from the ASCII value, so the value of 'a' is 2309. This value is stored in database for further processing for the same character in the different shape and the other characters were also recognize by the same procedure.

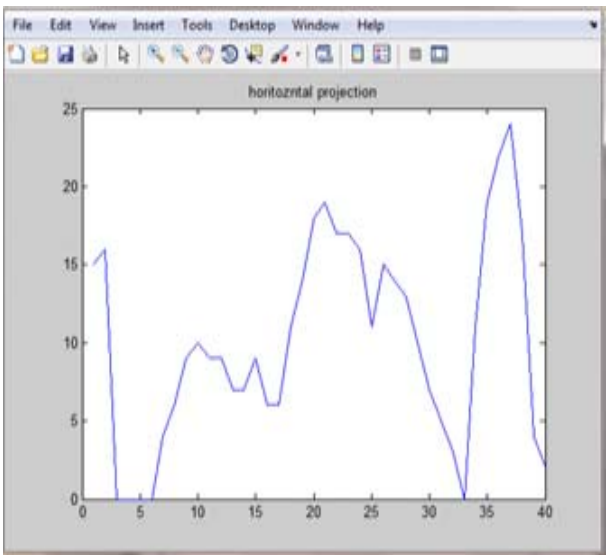

Figure 13: The horizontal projection of the character 'a'.

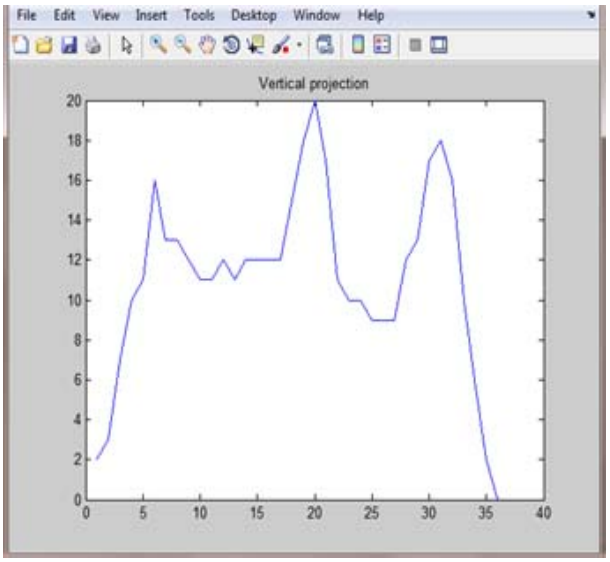

Figure 14: Vertical projection of the character 'a'.

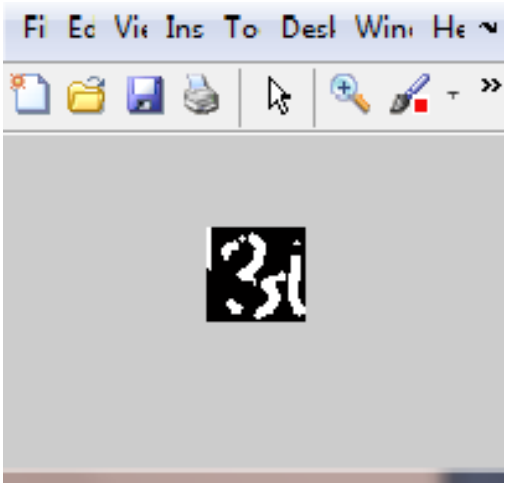

Figure 15: The final image of recognize character.

2.7 Classification

Classification of the Hindi characters is performed by the Euclidean distance method. Where for this type of algorithm we estimate the future vectors for the each and every train data and store it in a data base. The Euclidean method is a simple method but powerful enough for the detection of the characters. For the case of test data we estimate the vectors and then these vectors are separated from the train data. In classification k-means are used. K-mean algorithm is one of the simplest unsupervised learning algorithms that solve the well known problem of clustering. It follows a easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori.

The main idea is to define centroids of k numbers, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better way is to place them as much as possible far away from each other. The next step is to take each point which belong to a given set of data and associate it to the nearest centroid. When no point is remaining, the first step is completed. At this point k numbers of new centroids will be recalculated as bary centers of the clusters resulting from the previous step.

After we have these k numbers of new centroids, again a new binding has to be done between the same data set points and the nearest new centroid. Finally, the aim of this algorithm is to minimizing an objective function that is a squared error function.

3. Experimental Result and Discussion

The result obtained from this experiment is we can recognize Hindi characters with their four most similar shapes. The characters are recognized if they match 75% to the some other character it will consider as the same as other character by their horizontal and vertical projections. There are many shapes of a single character we can process as many shape of same character as possible. I work on another four more different shape of the same character.

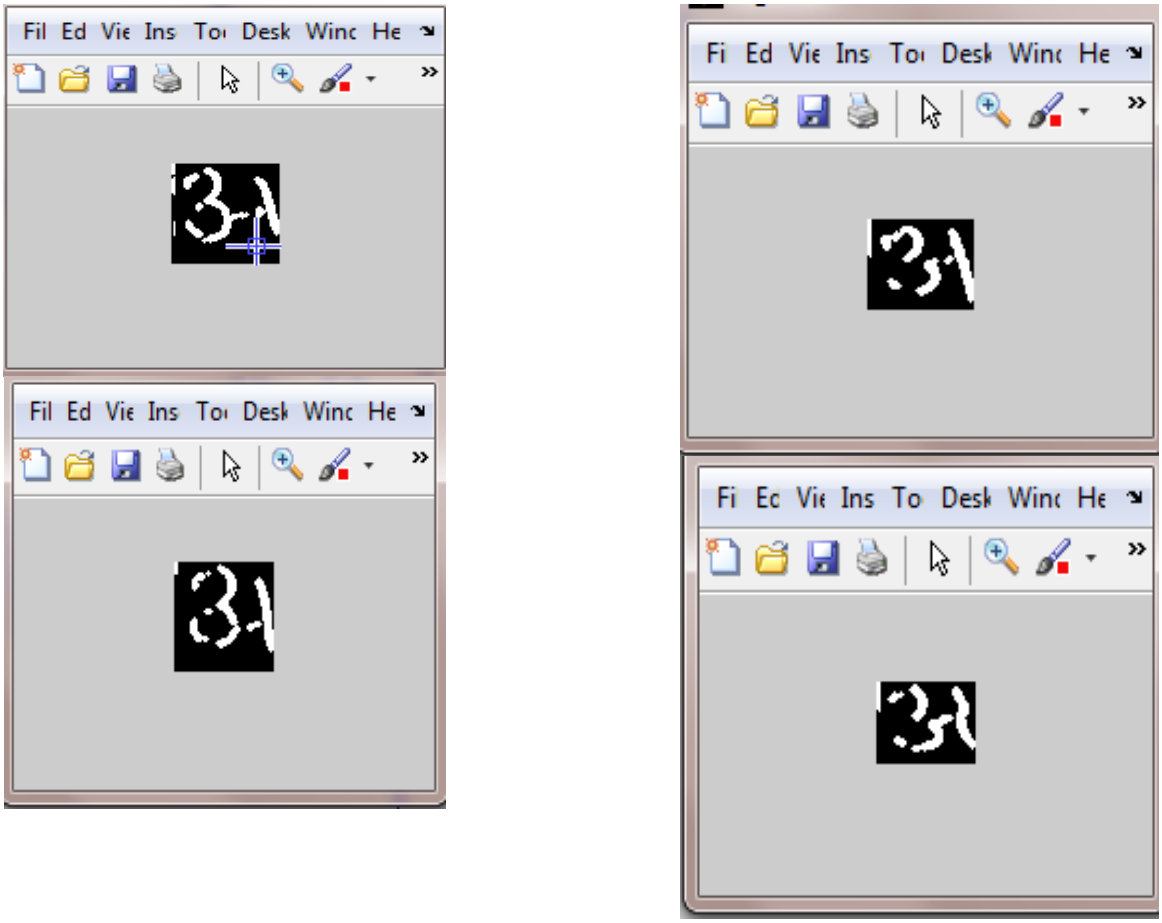

Figure 16: Outputs of different shapes of the same character

The classification values of the characters are stored in database. The database used in this work was trained with the four most standard fonts, and the accuracy brings by the k-means clustering. K-means clustering is an algorithm to cluster n data points specified in m dimensions or attributes into k clusters, considering that the point form a vector space. Each cluster is represented by a centroid of the data points in the cluster.

```
TA1=[0.770000000000000;0.590000000000000;0.480000000000000;1;0.860000000000000;0.280000000000000;0.970000000000000;0.340000000000000;1;1;0.950000000000000;
    0.700000000000000;0.490000000000000;1;0.400000000000000;1;0.200000000000000;0.760000000000000;0.440000000000000;0.340000000000000;1;0.950000000000000;
    0.420000000000000;0.570000000000000;0.400000000000000;0.750000000000000;0.970000000000000;0.810000000000000;0.480000000000000;0.370000000000000;
    0.770000000000000;0.800000000000000;0.220000000000000;0.860000000000000;0.460000000000000];
TA2=[0.640000000000000;0.450000000000000;1;0.890000000000000;0.950000000000000;0.720000000000000;0.580000000000000;1;0.930000000000000;0.590000000000000;
    0.960000000000000;0.350000000000000;0.960000000000000;1;0.520000000000000;0.850000000000000;0.860000000000000;0.480000000000000;0.680000000000000;
    0.330000000000000;0.720000000000000;1;0.430000000000000;0.970000000000000;0.520000000000000;0.640000000000000;0.500000000000000;0.710000000000000;
    1;0.830000000000000;1;0.900000000000000;1;1;1];
TA3=[0.860000000000000;0.790000000000000;0.570000000000000;1;1;0.530000000000000;0.990000000000000;0.400000000000000;0.970000000000000;0.570000000000000;
    1;0.960000000000000;0.490000000000000;1;0.420000000000000;1;0.700000000000000;0.430000000000000;1;0.430000000000000;0.570000000000000;1;0.580000000000000;
    0.720000000000000;0.350000000000000;0.530000000000000;0.800000000000000;0.410000000000000;0.970000000000000;0.450000000000000;1;0.930000000000000;
    0.910000000000000;1;0.800000000000000];
TA4=[0.790000000000000;0.270000000000000;0.850000000000000;0.750000000000000;0.900000000000000;0.330000000000000;0.760000000000000;0.620000000000000;
    0.800000000000000;0.660000000000000;0.930000000000000;0.550000000000000;0.800000000000000;0.870000000000000;0.480000000000000;1;0.880000000000000;
    0.470000000000000;0.610000000000000;0.390000000000000;0.910000000000000;1;0.660000000000000;0.740000000000000;0.410000000000000;0.680000000000000;
    0.880000000000000;0.310000000000000;0.920000000000000;0.470000000000000;1;1;0.850000000000000;1;0.700000000000000];
```

Figure 17: Classification values of the characters

4. Conclusion and Future Scope

The result of the experiment show that the problem of Hindi character recognition solves successfully with the use of K-means. By the use of OCR we can recognize all the Hindi characters and their matras which are use in the Hindi words. From the result we can see that by the segmentation method we can separate the touching characters and remove the shrirorekha from the word. We conclude the horizontal and vertical gradient for edge detection using manual canny edge detector and take the vertical and horizontal projection of each character for matching with other character and their classification. Therefore this experiment illustrate that we recognize all the Hindi character and classify them successfully.

Future scope of this experiment is we can recognize a full Hindi word or text instead of a character, we can recognize the Hindi numbers. Another scope is speech recognition of the Hindi character using OCR for blind people.

References

[1] Karthik Sheshadri, Pavan Kumar T Ambekar, "An OCR system for Printed Kannada using k-means clustering"
[2] E.R. Davies and A.P. Plummer, "Thinning Algorithms: A critique and new Methodology" ,Pattern Recognition 14, [1981]: 53-63
[3] S. Arora, D.Bhattacharya, M. Nasipuri, L.Malik, "A Novel Approach for Handwritten Devanagari Character Recognition" in IEEE –International Conference on Signal And Image Processing, Hubli, Karnataka, Dec 7-9, 2006
[4] Vedgupt Saraf, D.S. Rao, "Devnagari Script Character Recognition Using Genetic Algorithm for Get Better Efficiency" in International Journal of Soft Computing and Engineering (IJSCE), April 2013
[5] Rahul KALA1, Harsh VAZIRANI2, Anupam SHUKLA3 and Ritu TIWARI4, "Offline Handwriting Recognition using Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 2, No 1, March 2010.
[6] Shabana Mehfuz1, Gauri Katiyar2, ''Intelligent Systems for Off-Line Handwritten Character Recognition: A Review" International Journal of Emerging Technology and Advanced Engineering, Volume 2, Issue 4, April 2012

Author Profile

Meha Mathur received the B.Tech degrees in Information Technology from Vyas Institute of Engineering and Technology in 2012 and pursuing M.Tech degree from Amity University Rajasthan. She is now working on the implementation of HCR.