PRINTED DEVANAGARI SCRIPT RECOGNITION USING DIFFERENT CLASSIFIERS

Sushama Shelke

Lead Data Analyst

Department of AI and ML
ARIN International, Pune, India.

Abstract : The development of character recognition is an interesting area in pattern recognition. Character Recognition is a process by which a computer recognizes letters, numbers, or symbols and turns them into a digital form that a computer can use and it is an active field of research today. It comprises of Pattern Recognition and Image Processing. Character Recognition is broadly categorized into Optical Character Recognition (OCR) and Handwritten Character Recognition (HCR). OCR system is most suitable for the applications like multi choice examinations, printed postal address resolution etc, while application of HCR is wider as compared to OCR. HCR system consists of a number of stages which are preprocessing, feature extraction, classification and followed by the actual recognition. Feature extraction and classification are essential steps of character recognition process affecting the overall accuracy of the recognition system. In this paper, system for Devanagari script recognition is discussed which separates the characters, extracts features and then recognizes the script.

Index Terms **- OCR, preprocessing, feature extraction, classification.**

I. INTRODUCTION

Devanagari script is one of the scripts in Brahmic family which is found in south east part of Asia. Languages derived from Devanagari are spoken in India, Tibet, and Nepal. Hindi, one of the most popular languages derived from Devanagari script is the national language of India. Hindi is the fourth most popular language in the world with 420 million speakers. Many ancient books and literature are also found in Devanagari script. Devanagari script is also used for official communication in few states instead of English. Therefore, there is a need to develop a system that converts them into machine readable documents. The character set in Devanagari has different fonts. Devanagari is written from left to right under a horizontal line. There are no upper and lower case letters in this script. It also includes vowels that take different shapes and forms, conjunct characters and similar characters that make this character recognition quite challenging [1].

The rest of paper is organized as follows: Section II discusses the literature survey in brief. Section III discusses the proposed system design. Section IV indicates features extracted and Section V indicated the classifiers. VI and VII discuss results obtained and conclusion.

II. LITERATURE SURVEY

Work on Devanagari script OCRs started with the recognition of printed script, where the characters were separated based upon the structural features in the characters [2-4]. Later, recognition attempts were made by extracting the features from printed and handwritten characters and scripts. Although some work is found for printed script recognition, there are no systems available for handwritten script recognition in Devanagari. Features play a crucial role in character recognition [5]. Different features are extracted from the segmented characters to obtain best representation of the characters under test [6]. Researchers applied various features to different classifiers to obtain a best combination of feature extraction and classification methods [7, 8]. Recently, deep learning techniques [9] are also applied which extract the features automatically.

III. PROPOSED SYSTEM

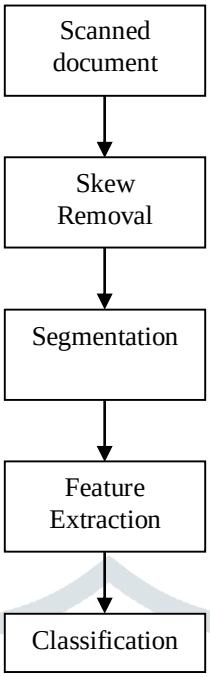

Fig. 3.1 System block diagram.

Figure 1 shows the system block diagram. The steps involved in implementing the proposed system to segment the words are given below:

Step 1: Read the image and store the binarized image for pre-processing and segmentation.

Step 2: For the correction of skew, the input image is first binarized and its edge is detected using Canny edge detection. Now radon transform of this image is taken which gives us the misalignment angle of the image. Finally, the image is rotated accordingly using the angle we obtained earlier and the skew is corrected.

Step 3: We find 8-connectivity regions present in the binarized, skew corrected image. 8-connected pixels are neighbors to every pixel that touches one of their edges or corners. These pixels are connected horizontally, vertically, and diagonally. Now, after getting these groups of 8-connected pixels, we remove all such groups which are below a certain threshold.

Step 4: All the remaining 8-connected groups left are labelled from 1 to n where n is the total number of labels present in the image. This is done by taking a copy of the binarized image and naming this as the label matrix. All pixels that belong to a particular 8-connected group are given value of the label number. For example, all the pixels in group 3 will have value 3.

Step 5: The unwanted labels are discarded by removing all the labels who's label size is less than a minimum threshold or greater than a maximum threshold which classifies the label as either a word or unwanted noise. Label size is the total number of pixels present in the label. Also, the labels having width\height considerably bigger or smaller than the average label width \height are removed. All the remaining labels are numbered from 1 to n2 after removal of intermediate labels where n2 is final number of words in the document

IV. FEATURE EXTRACTION

Feature extraction involves reducing the amount of resources required to describe a large set of data. When performing analysis of complex data one of the major problems stems from the number of variables involved. Analysis with a large number of variables generally requires a large amount of memory and computation power; also it may cause a classification algorithm to overfit to training samples and generalize poorly to new samples. Feature extraction is a general term for methods of constructing combinations of the variables to get around these problems while still describing the data with sufficient accuracy. The deferent feature extraction methods utilized are: 1. Zoning and 2. Projection Histogram

4.1 Zoning

Zoning is a statistical feature extraction method in which the frame containing the character is divided into several overlapping or non-overlapping zones. The densities of the points or some features in different regions are analyzed. In this method, all the segmented characters are resized to 32*32 pixels which is then divided into 16 equal zones or blocks each of size 8*8 pixels. The features are extracted by counting the number of black pixels and dividing it by the total number of pixels in each zone. This procedure is repeated sequentially for all 16 zones which are stored in the form of signature array for each character. This array is then extrapolated to a length of 32. Thus, for each Devanagiri character we get a signature array of length 32 calculated from each zone. Fig. 2 shows the character 'tha' and its zoning features.

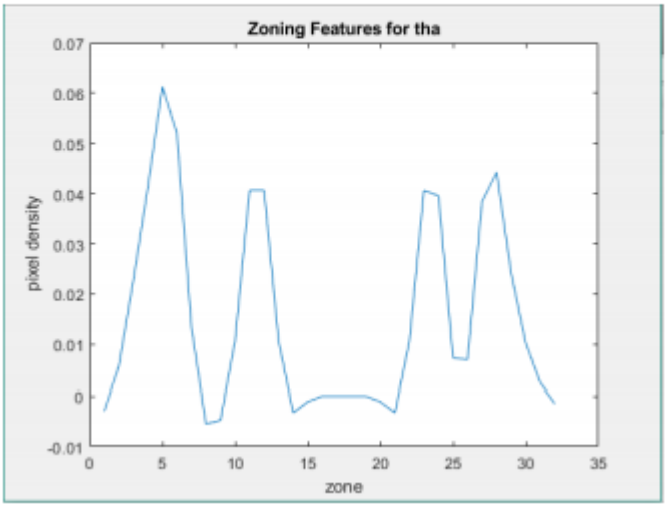

Fig. 4.1 Zoning features for character 'tha'.

4.2 Histogram Projection

Characters can be represented by projecting the pixel gray values onto lines in various directions. This representation creates one-dimensional signal from a two dimensional image, which can be used to represent the character image. The projection is divided into 3 categories: 1. Horizontal Projection 2. Vertical Projection 3. Diagonal Projection.

Binarization of the image leads to only two kinds of black and white color, the pixel gray value is 0 and 255, the convention here is that black pixels are recorded as 1, and the white pixels as 0. Horizontal projection and vertical projection profile of the image can be obtained by calculating the number of black pixels along the rows and 14 columns of an image respectively. To maintain uniformity, this number is normalized by dividing it with the maximum possible number of black pixels in the said directions. Similarly, for Diagonal projection the number of black pixels are counted diagonally and then again normalized. The segmented character image is of size 32*32 pixel. We obtain an array of length 32 for horizontal and vertical projection each and an array of length 63 for the diagonal projection. Thus, we get an overall array of length 127 which is then interpolated to a length of 32. Horizontal, vertical and diagonal features for character 'tha' are shown in Fig. 3, Fig. 4 and Fig. 5 respectively.

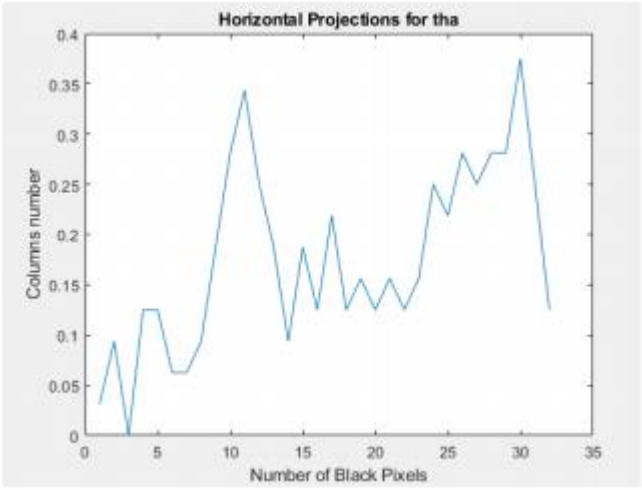

Fig. 4.2 Horizontal projections for character 'tha'.

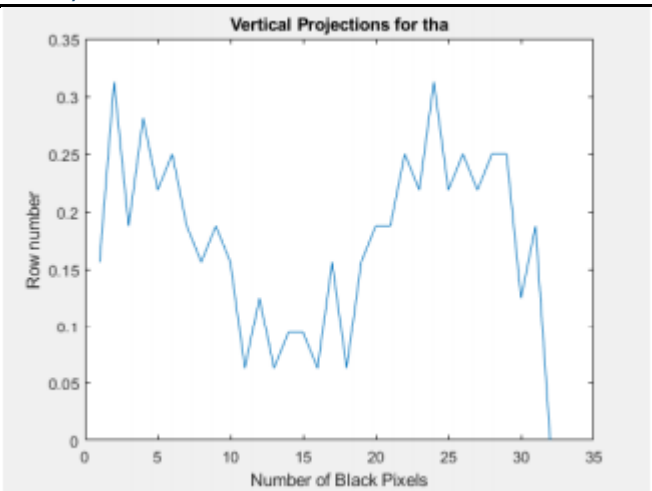

Fig. 4.3 Vertical projections for character 'tha'.

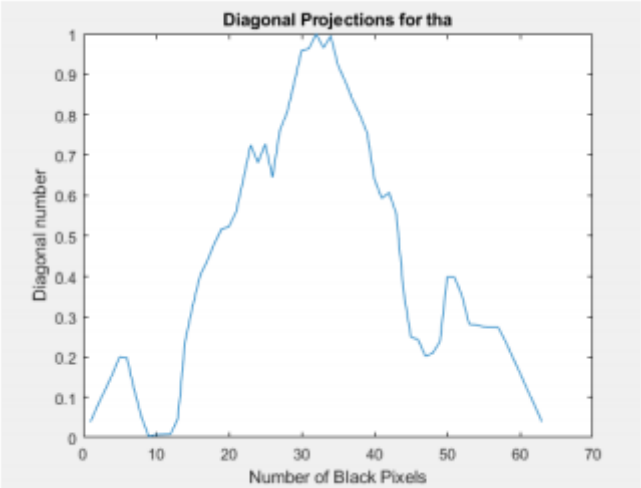

Fig. 4.4 Diagonal projections for character 'tha'.

V. CLASSIFICATION

Classification stage is to recognize characters or words. After features that represent the raw input data are extracted, classification stage would use the data to recognize the feature class based on the properties in the features. The different classification methods used are: 1. Template Matching or correlation 2. K-Nearest Neighbors, and 3. Convolutional Neural Network (CNN).

VI. RESULTS

For collection of data, we decided to use 3 different fonts i.e, 'Kiran', 'Prachi' and 'Amruta'. These three fonts were chosen after comparing the segmented output and finding out the most accurate and clear output. Different font sizes aren't considered because ultimately all the segmented character images are rezised to 32*32 pixels. Thus, different sizes don't make a difference. To create a training database, all the 46 characters of Devanagiri script consisting of 35 consonants and 11 vowels are considered. Thus, we obtain a data set of (46*3) = 138 characters. The characters are written sequentially on a word document interleaved with spaces and then printed. Later, photo of the printed page is taken and run through all the steps starting from the pre-processing stage. To create testing data set for entire script detection, random words are printed and taken a photo of. This image is taken as the input and again run through all the stages mentioned earlier. Furthermore, for testing the various classifiers and improving their accuracy to the requisite level, separate data sets were created for every classifier to optimize their outputs. These data sets were created by taking a few variations of every individual character present in the Devanagiri script. The accuracy of each classifier was calculated by comparing the detected script to the actual input script. Fig. 6, Fig. 7 and Fig. 8 shows the skew correction, word separation and character separation results respectively.

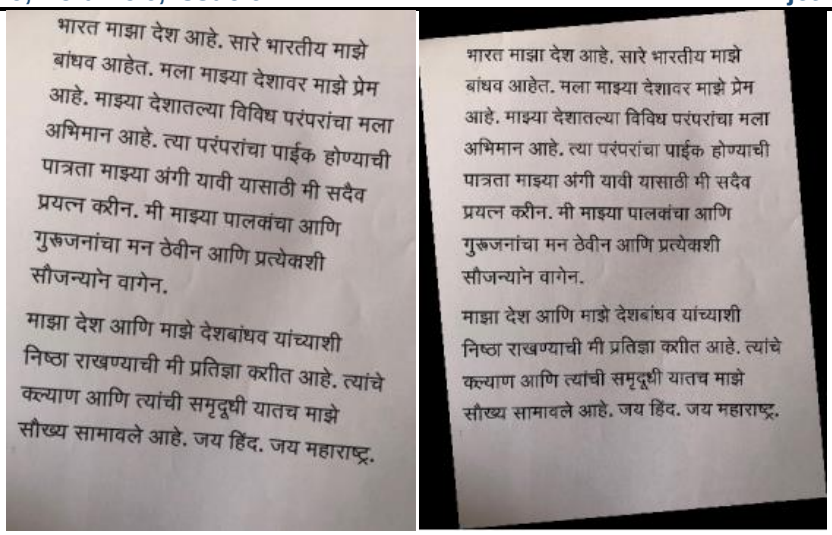

Fig.6.1 Skew correction.

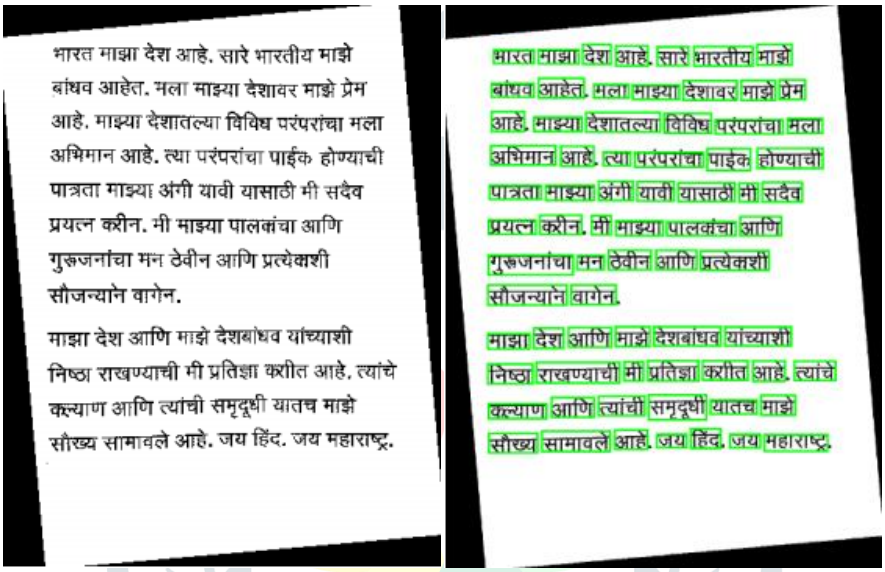

Fig. 6.2 Word separation.

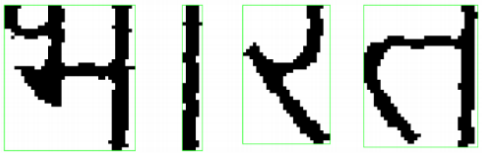

Fig. 6.3 Character separation.

Table 6.1 Recognition Result

Classifiers	Database Used For Training	Database Used For Testing	Features	Accuracy
Template Matching	1056	264	Projections & Zoning	73
KNN	1056	264	Projections & Zoning	75
CNN	1056	264	Projections & Zoning	96

The recognition results for various classifiers are indicated in Table 1. As seen CNN gives best recognition accuracy amongst the three classifiers.

VII. CONCLUSION

In this paper, a comprehensive recognition system for Devanagari script is discussed. The system, addresses skew correction and character separation from the entire script. Various classifiers are applied to the extracted features and recognition accuracy is tested. Results indicate that CNN gives better results than Template matching and KNN.

REFERENCES

[1] U. Pal and B. B. Chaudhuri. 2004. Indian script character recognition: A survey. Pattern Recognit., 37: 1887–1899.

[2] R. M. K. Sinha and H. Mahabala. 1979. Machine recognition of Devnagari script. IEEE Trans. Syst. Man Cybern., 9(8): 435–441.

[3] V. Bansal and R. M. K. Sinha. 1999. On How to Describe Shapes of Devanagari Characters and Use them for Recognition. Proc. 5th International Conference on Document Analysis and Recognition, 410-413.

[4] V. Bansal, and R.M.K. Sinha. 1999. Partitioning and Searching Dictionary for Correction of Optically Read Devanagari Character Strings. Proc. 5th International conference on Document Analysis and Recognition, 653-656.

[5] Oivind Due Trier, Anil K. Jain and Torfinn Taxt. 1996. Feature Extraction Methods for Character Recognition – A Survey. Pattern Recognition, 29: 41-62.

[6] Sushama Shelke, and Priti Rege. 2018. Rotation Invariance in Transform Features for Handwritten Devanagari Character Recognition. Proc. 4th IEEE sponsored International Conference for Convergence in Technology, 1-5.

[7] S. Kumar. 2009. Performance comparison of features on Devanagari handprinted dataset. Int. J. Recent Trends in Engineering, 1(2): 33–37.

[8] S. Shelke, and S.Apte. 2010. A Novel Multi-feature Multi-classifier Scheme for Unconstrained Handwritten Devanagari Character Recognition. Proc. 12th Int. Conf. Frontiers Handwrit. Recognit., 215-219.

[9] Prasad Sonawane, and Sushama Shelke. 2018. Handwritten Devanagari Character Classification using Deep Learning. IEEE International Conference on Information, Communication, Engineering and Technology, 1-5.