OCR of Devanagiri Characters

Interim report for CS 6350

Yogesh Mishra Nishant Agarwal

1 Current State

The problem we are trying to solve is OCR (Optical Character Recognition) of Devanagiri Characters. Devanagiri script contains 13 vowels and 36 consonants in its character set. The script is quite complex such that basic characters can couple to form modified characters, hence we will be classifying and recognizing the basic characters itself.We found a ready made dataset for Devanagiri characters from HP labs[6]. The dataset is divided into training and test sets. Each character is already segmented into separated images. The image format is TIFF(Tagged Image File Format). Our training set contains handwritten samples of the 49 characters from 87 users. Each user has given two sample of each character. Similarly our test set contains handwritten samples of each character from 19 users. Each user has given a single sample for each character.

We searched extensively and read many papers published on OCR. In our search to understand an approach to the problem we came across [1][2][3][4][5][7].Here we learned that our work needs a mixture of image processing and computer vision to process the image and extract features for classification. The steps for OCR are:

a. Pre-processing of image: This step requires skew detection and correction, noise removal, etc. to get a clean data to work on.

b. Segmentation: In this step characters are extracted from words and the image is normalized.

c. Feature extraction: In this step computer vision algorithms are applied to get feature vectors so that we can classify the characters.

d. Classification and Recognition: In this step we train the classifier and then recognize the characters from the test set.

   As our data is already segmented and filtered we have normalized all the images to a fixed size of 32X32 pixels using OpenCV[8]. This will vary based on future experiments. For feature extraction we have found that methods like shadow detection, Code chain histograms of character contours, histograms of oriented gradients, intersection/junctions in a character,etc. are used. We have yet to decide on the feature extraction method we will use.

Based on [1][2][3][4][5][7] we have found that SVM(Support Vector Machines) give a better accuracy than other classifiers like Naive Bayes, kNN and Ada Boost.

2 Plan

Our future plan of action is as follows:

a. We will decide and implement a feature extraction method.

b. We will implement SVM and kNN to perform a comparison between the classifiers.

c. If time permits we will perform a comparison using neural networks, Ada Boost and Naive Bayes.

d. If time permits we will try implementing multiple feature extraction methods and perform classifications.

References

[1] Sandhya Arora, Debotosh Bhattacharjee, Mita Nasipuri, Dipak Kumar Basu, and Mahantapas Kundu. Combining multiple feature extraction techniques for handwritten devnagari character recognition. In *Industrial and Information Systems, 2008. ICIIS 2008. IEEE Region 10 and the Third international Conference on*, pages 1–6. IEEE, 2008.

[2] Sandhya Arora1 Debotosh Bhattacharjee, Mita Nasipuri, L Malik, M Kundu, and DK Basu. Performance comparison of svm and ann for handwritten devnagari character recognition. *IJCSI*, page 18, 2010.

[3] Deepali R Birajdar and Manasi M Patil. Recognition of off-line handwritten devanagari characters using combinational feature extraction. *International Journal of Computer Applications*, 120(3), 2015.

[4] Vikas J Dongre and Vijay H Mankar. Devnagari handwritten numeral recognition using geometric features and statistical combination classifier. *arXiv preprint arXiv:1310.5619*, 2013.

[5] Aditi Goyal, Kartikay Khandelwal, and Piyush Keshri. Optical character recognition for handwritten hindi, 2010.

[6] HP Labs. Hp labs india indic handwriting datasets. `http://programmingcomputervision.com/downloads/ProgrammingComputerVision_CCdraft.pdf`.

[7] Neural Networks. Neural networks for ocr. `http://www.codeproject.com/Articles/11285/Neural-Network-OCR`.

[8] Jan Erik Solem. Programming computer vision with python. `http://programmingcomputervision.com/downloads/ProgrammingComputerVision_CCdraft.pdf`.