CS-299
Innovative Design Laboratory

Hindi OCR
Using Neural Networks

Abhishek Nautiyal
(1601CS02)
Piyush Singh
(1601CS30)

ABSTRACT

English Character Recognition has been extensively studied in the last half century and progressed to a level, sufficient to produce technology driven applications. But same is not the case for Indian languages which are complicated in terms of structure and computations. As there is no separation between the characters of texts written in Hindi as there is in English it further complicates the segmentation process, creating a major problem when designing an effective character segmentation technique.

Rapidly growing computational power may enable the implementation of Hindi OCR methodologies. Digital document processing is gaining popularity for application to office and library automation, banks and postal services, publishing houses and communication technology, to nursing homes and hospitals.

Hindi is the most widely spoken language in India, with more than 300 million speakers, should be given special attention so that document retrieval and analysis of rich ancient and modern Indian literature can be effectively done.

AIMS, OBJECTIVES & SIGNIFICANCE

Converting the image of a machine printed Hindi document into a editable format. This project holds great significance since it aims to assist in easing the conversion from physical to electronic type. Such capacity holds significant credibility and its advantages are limitless. This converts handwritten or printed symbols from simple pictures to helpful information that may be utilized in computers. These handwritten or printed documents do not stay in a large pile of pages in the workplace, instead they are now turned into digital information that can be easily interpreted by the computers. This also makes the process of searching these documents easy. Information is turned into digital format without any individual having to do the tedious work himself. Digital document processing has large applications in offices, banking services, communication technology.

Therefore, it becomes obvious that OCR platforms are incredibly useful for such functions and could be manipulated more to be employed in various other tasks of daily life that can be personal as well as commercial in nature

RELATED WORK

Most OCR techniques use split words into smaller units, though implementations vary on the level of segmentation. While some approaches use character as classification units others segment a character into components before classification. When characters used as classification units, horizontal and vertical profile of each word is examined to remove the header line. In Hindi words, the header line is very well-defined and is the basis for modelling classification techniques.

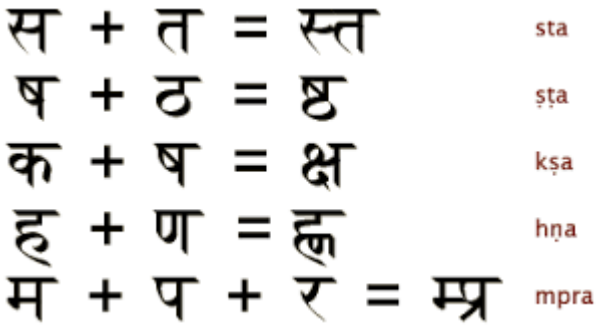
Some Popular reference works are cited below :

- Kailash S. Sharma,A. R. Karwankar, Dr. A.S.Bhalchandra," Devnagari Character Recognition Using Self Organizing Maps" ICCCCT'10
- H.Ma and D. Doermann, "Adaptive Hindi OCR using generalized Hausdorff image comparison," ACM Trans. Asian Lang. Inf. Process. vol. 2, no. 3, pp. 193–218, 2003.
- R.M.K. Sinha, and Veena Bansal, "On Automating trainer for construction of prototypes for Devnagari text recognition", Technical report TRCS-95-232, IIT Kanpur, India 1995.
- U. Bhattacharya and B. B. Chaudhuri, "Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals," IEEE Trans. Pattern Anal. Mach. Intell., vol. 31, no. 3, pp. 444–457, Mar. 2009.
- U. Pal and B. B. Chaudhuri, "Indian script character recognition: A survey," Pattern Recognit., vol. 37, pp. 1887–1899, 2004.
- R.M.K. Sinha, and Veena Bansal, "On Devanagari documentation processing", IEEE International Conference on Systems, Man and Cybernetics, Vancouver, Canada 1995.
- Veena Bansal, R.M.K. Sinha, "On How to Describe Shapes of Devanagari Characters and Use Them for Recognition," icdar, pp.410, Fifth International Conference on Document Analysis and Recognition (ICDAR'99), 1999
- Veena Bansal & R.M.K. Sinha, "Segmentation of Touching Characters In Devanagari", http://www.iitk.ac.in/ime/veena/PAPERS/stwo.pdf
- M.Babu Rao, Dr.B.Prabhakara Rao, Dr.A.Govardhan, "Content Based Image Retrieval using Dominant Color and Texture features" (IJCSIS) International Journal of Computer Science and Information Security,Vol. 9, No. 2, February 2011
- R. Jayadevan, Satish R. Kolhe, Pradeep M. Patil, and Umapada Pal "Offline Recognition of Devanagari Script: A Survey", ieee transactions on systems, man, and cybernetics—part c: applications and reviews, vol. 41, no. 6, november 2011.

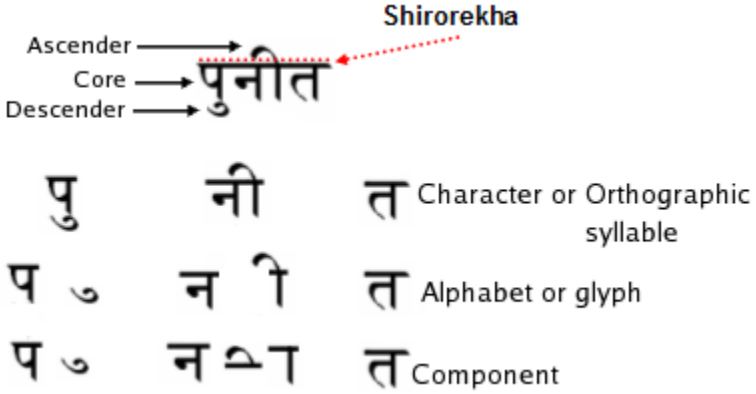
NOVELTY

As we know that recognition of character is not an easy task. Due to various font sizes and writing style it is difficult to recognize the character. Also in Devnagari script, many characters have similar shape, which creates trouble in recognition.

While in English language words are essentially isolated alphabets printed in close proximity, on the other hand in Hindi a large number of characters can be formed using existing characters and thus increasing the difficult several folds.



Also unlike alphabets in English, Hindi characters consists of three different layers listed as follows :



These additional attributes of the hindi language and the presence of more than 50 unique characters that can conjoin to form further more new characters makes this project exciting and hard at the same time!

The DATASET

Table 1 Class space of our system (components)

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

Consonants and vowels

Half-consonants

Ascenders

Descenders

WORK OUTLINE & PLAN OF WORK

The main steps involved in Character Recognition are as follows :

- Text Digitization
- Gray Tone to Two Tone Conversion
- Noise clearing
- Text Block Identification
- Skew Correction
- Line and Word Detection
- Character Segmentation

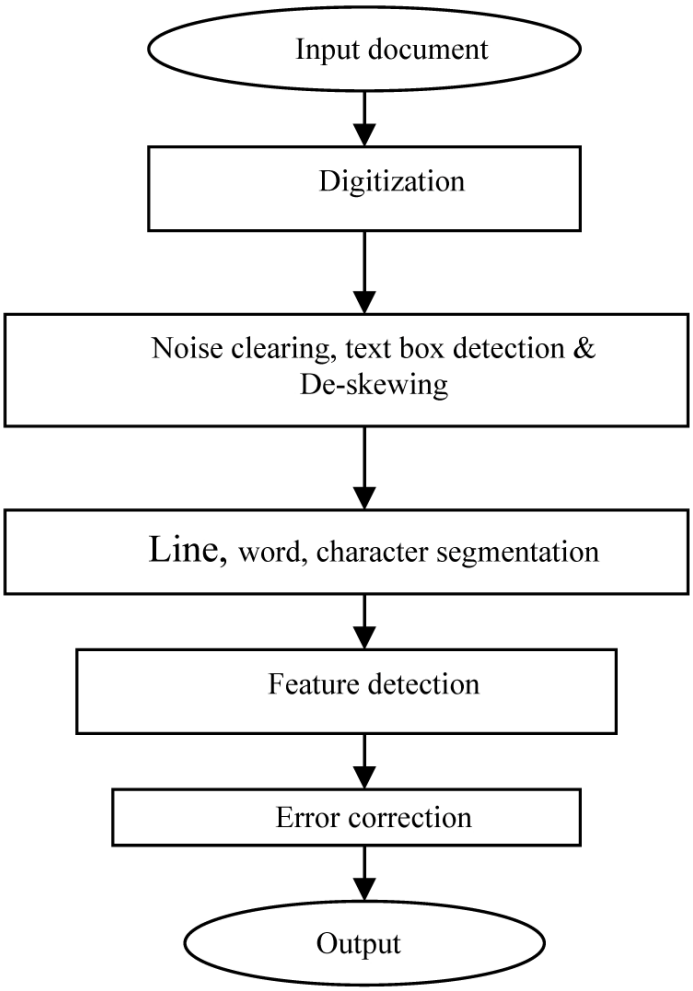


IMAGE PREPROCESSING

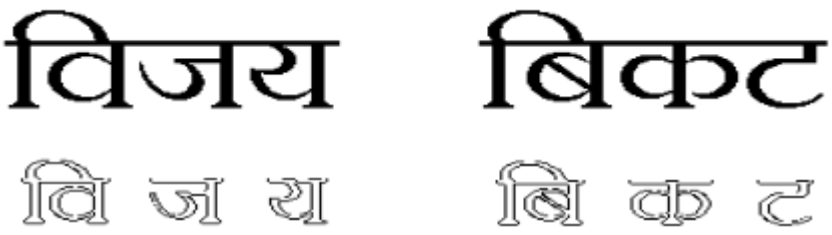
Data in a paper document are usually captured by optical scanning and stored in a file of picture elements, called pixels. These pixels may have values: OFF (0) or ON (1) for binary images, 0– 255 for gray-scale images, and 3 channels of 0–255 colour values for colour images. This collected raw data must be further analyzed to get useful information.
Such processing includes the following:

1. Thresholding :
   A grayscale or colour image is reduced to a binary image.

2. Noise reduction :
   The noise, introduced by the optical scanning device or the writing instrument, causes disconnected line segments, bumps and gaps in lines, filled loops etc. The distortion including local variations, rounding of corners, dilation and erosion, is also a problem. Prior to the character recognition, it is necessary to eliminate these imperfections.

3. Skew Detection and Correction :
   Documents may originally be skewed or skewness may introduce in document scanning process. This effect is unintentional in many real cases, and it should be eliminated because it dramatically reduces the accuracy of the subsequent processes, such as segmentation and classification. Skewed lines are made horizontal by calculating skew angle and making proper correction in the raw image.

4. Edge-Detection :
   The boundary detection of image is done to enable easier subsequent detection of pertinent features and objects of interest by using the canny edge detector.



SEGMENTATION

It is one the most important process that decides the success of character recognition technique. It is used to decompose an image of a sequence of characters into sub images of individual symbols by segmenting lines and words. Hindi words can further be splitted to individual character for classification and recognition by removing Shirorekha (header line). The Shirorekha is removed by applying line detection using Hough Transform method.Various vowel modifiers can be separated for structural feature extractions.

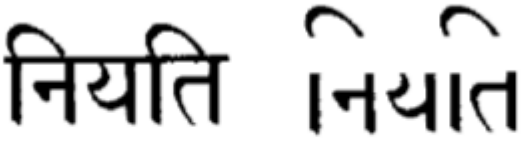


Figure 1. Sirorekha removal for a Hindi word

FEATURE DETECTION AND CLASSIFICATION

Feature extraction and selection can be defined as extracting the most representative information from the raw data, which minimizes the within class pattern variability while enhancing the between class pattern variability. For this purpose, a set of features are extracted for each class that helps distinguish it from other classes, while remaining invariant to characteristic differences within the class.

In this case, we will be going to apply the HOG image de-scriptor and a Linear Support Vector Machine (SVM) to learn the representation of characters.For this system, we used python, openCV and sklearn to run classification and read the dataset.

The sample of the dataset consists of 30000 data points, each with a feature vector of length 784, corresponding to the 28 × 28 grayscale pixel intensities of the image.

Support Vector Machine (SVM)

SVM are a set of supervised learning methods used for classification, regression and outliers detection. In SVM, each data item will be plot as a point in n-dimensional space (n = number of feature) with the value of each feature being value of a particular coordinate. The classification is done by finding the hyper-plane that differentiate the two (2) classes.

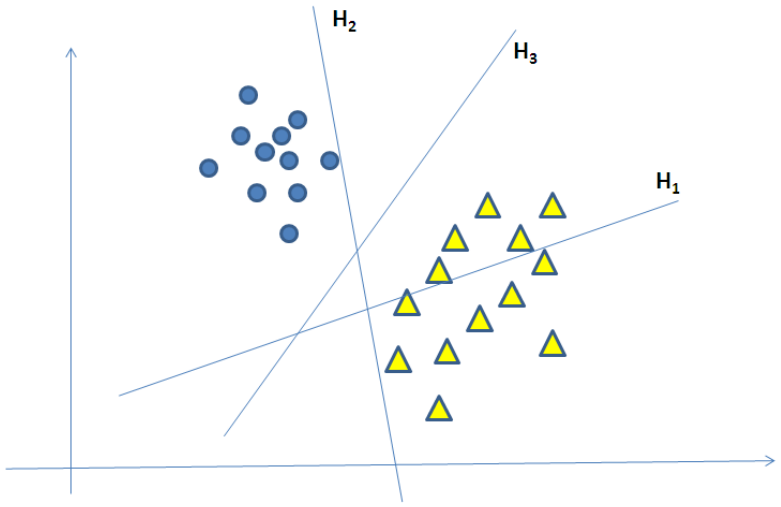


FIGURE: Separation hyperplanes. H1 does not separate the two classes; H2 separates but with a very tiny margin between the classes and H3 separates the two classes with much better margin than H2.

The advantage of SVM classifier over other classifiers can be explained as follows. An Indian language OCR system generally has large number of classes and high dimensional feature vectors. Variability of characters is also very high at each occurrence. SVMs are well suited for such problems since they have excellent generalization capability. They always result in a global solution of the problem.

The SVM in scikit-learn support both dense numpy.ndarray and convertible to that by numpy.asarray) and sparse (any scipy.sparse) sample vectors as input. In scikit-learn have three (3) classes that capable of performing multi-class classification on a dataset which is SVC, NuSVC and LinearSVC. In this system we will use LinearSVC class to perform the classification on our dataset. LinearSVC or Linear Support Vector Classification that use linear kernel and implemented in terms of liblinear that has

more flexibility in the choise of penalties and loss functions and should scale better large numbers of samples.

In this classification, we will have two (2) file which is classify.py for classification and predict.py for testing the classification. In the generateClassifier.py we will perform three (3) step such as:

- Calculate the HOG features for each sample in the database.
- Train a multi-class linear SVM with the HOG features of each sample along with the corresponding label.
- Save classifier in a file.

The dataset images of the digits will be save in a numpy array and corresponding labels. Next we will calculate the HOG features for each images and save them in another numpy array.

Computing the HOG descriptor is handled by the hog method of the feature sub-package of scikit-image. We pass in the number of orientations, pixels per cell, cells per block, and whether or not square-root transformation should be applied to the image prior to computing the HOG descriptor.

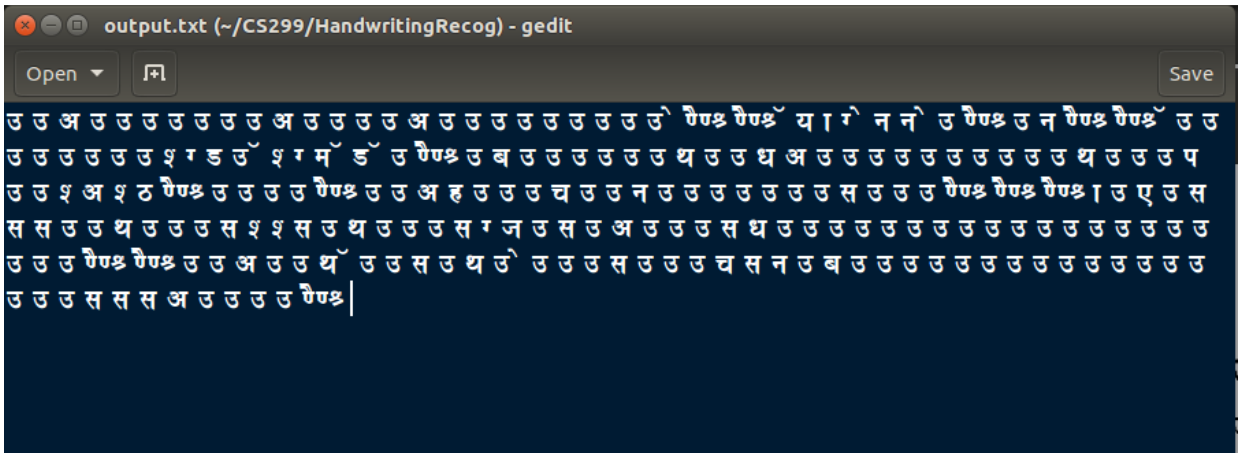
Then we will create Linear SVM object and perform the training for the dataset then we will save the classifier in a file.

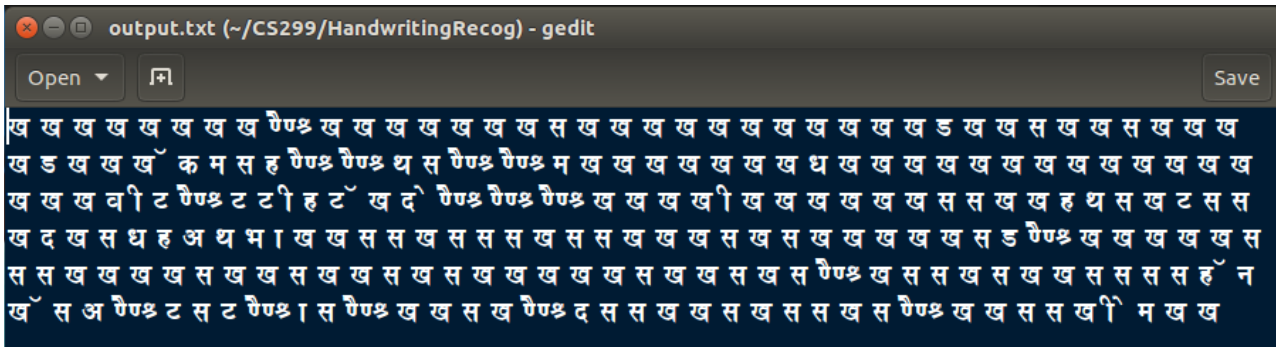
Now to classify a test image, we compute the HOG feature vector of the thresholded ROI(Region of Interest) by calling the describe method of theHOG descriptor.
The HOG feature vector is then fed into the LinearSVC's predict method which classifies which character the ROI is, based on the HOG feature vector.

RESULTS & OBSERVATIONS

The results reported here are character level while in real-life applications of the system we will be comparing with recognized sentences and at paragraph levels.

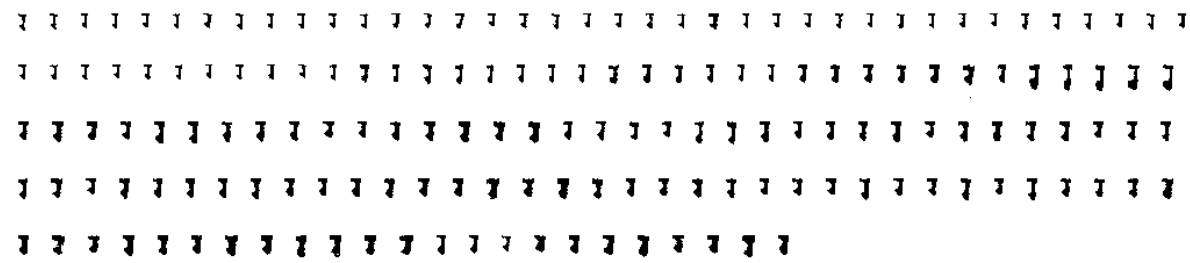
Accuracy rate is calculated by first calculating the total sum of the given letter and then finding the percentage of given letter where it's false. As you see,some letters have a very low accuracy
whereas the accuracy of other letters is satisfactory.
After training the given letters with more data sets, results are improved.
However, there has been a small decline in some of the letters. The main reason for the decline is because of the increasing similarity of 2 letters and this problem can be solved by writing them in different styles. It's still not guaranteed that the accuracy rate will reach 100% but it will slightly increase if the image is well processed by the system.

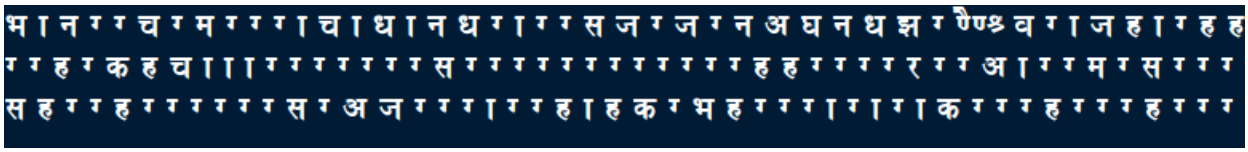




In this example the programs gets confused between two similar characters i.e., and as a result there is a good number of times that it gives that character as an output.

While in other cases due to low number of training images the accuracy is low.
For such cases we need to drastically imporve the number and quality of the trainig dataset, one such example is:





CONCLUSION

The experimental results illustrate that the Machine Learning concept can be applied successfully to solve the Hindi optical character recognition problem. There are many variations of factors that affect the performance of the developed Hindi OCR software.

However, other kinds of preprocessing and neural network models may be tested for a better recognition rate in the future. The character segmentation method can be improved to handle larger variety of touching characters that occur often in images obtained from inferior-quality documents. The test set used in this experiment is of 77 characters of different types of fonts. This can be tested for a greater number of fonts. The toughest phase in the experiment is getting a good set of characters for classification. This highlights the need for generation of a large ground-truthed set of characters of various resolutions so that more research can be performed for recognition of languages from Indian subcontinent.

Additional work can also be completed in order to improve the outcomes and solve the problems that were confronted with by the training of a bigger set of images.

The system cannot segment certain type of characters that are formed by adding small modifiers to existing consonant or fomed by addition of 3 or more characters, this is because they comprise a small percentage of the of the language and hence sufficient sample cannot be used.

FUTURE WORK

The other future enhancement that can be incorporated in the work is to use a dictionary of words to correct the output. Certainly this will improve the performance. Further speech synthesizer can be integrated with the OCR with the aim of making a system for reading aids to the blind. We can also implement the neural network method for classifying hand-written texts.

In hand-written documents, the fragmentation of characters and the variation in shape of characters are considerably greater compared to printed documents. The higher levels can be used to provide clues for a hypothesization system, which learns from the text it recognizes. We have not dealt with punctuation marks and numerals in this work. The set of the punctuation marks and the numerals need special treatment right from the point of preliminary segmentation of words into characters and symbols. For Devanagari script, the header line is removed from the character before an attempt is made to classify it. However, a numeral or punctuation mark cannot be dealt with the same manner. There is no header line to be removed, even though some such marks may have a horizontal line which resembles the header line. We implemented the work for the scripts with only Hindi characters. However, we can extend it to classify the document with characters of more than one script.