Avant Garde method for Character Recognition for Devanagari Script using Hidden Markov Model and Neural Network

Dhara Gupta*
Dr. Udayan Ghosh**

Abstract

Devanagari script is a two dimensional composition of symbols. It is highly cumbersome to treat each composite character as a separate atomic symbol because such combinations are very large in number. The aim of this paper is to produce a system that classifies a given input as belonging to a certain class rather than to identify them uniquely, as every input pattern. The system performs character recognition by quantification of the character into a mathematical vector entity using the geometrical properties of the character image. The scope of the proposed system is limited to the recognition of a single character. The goal and objective of the proposed software is "Developing a system that helps in recognizing an unknown character presented to it.

Keywords: Devanagari handwritten character, script, text recognition, hidden markov model

Introduction

In the proposed paper, we shall be dealing with the problem of machine reading typewritten/handwritten characters. This corresponds to the ability of human beings to recognize such characters, which they are able to do us in little or no difficulty. The aim is to produce a system that classifies a given input as belonging to a certain class. The system performs character recognition by quantification of the character into a mathematical vector entity using the geometrical properties of the character image. The scope of the proposed system is limited to the recognition of a single character. We present a method for segmentation of text printed in Devanagari. Our segmentation approach is a hybrid approach, wherein we try to recognize the parts of the conjunct that form part of a character class. We use a set of letters that are robust and two distance based classifiers to classify the segmented images into known classes. We present a two level partitioning scheme and search algorithm for the correction of optically read Devanagari characters of text recognition system for Devanagari script. We use hidden Markov model for prediction of a character and back-propagation algorithm for training purpose.

Proposed scheme

Its different stages are as given below:

**Assistant Professor, VIET, Gautam Budh Nagar, Uttar Pradesh, India.*
***Associate Professor, GGSIPU, New Delhi, India.*

Authors are grateful to the anonymous referees for comments and suggestions for improving the text and contents of the paper. Authors alone are responsible for any errors/mistakes still remaining in the paper.

- **Input:** Samples are read by the system through a scanner.
- **Preprocessing:** Preprocessing converts the image into a form suitable for subsequent processing.
- **Segmentation:** The most basic step in CR is to segment the input image into individual glyphs. This step separates out sentences from text and subsequently words and letters from sentences.
- **Feature extraction:** Extraction of features of a character forms a vital part of the recognition process. Feature extraction captures the vital details of a character.
- **Classification:** During classification, a character is placed in the appropriate class to which it belongs.
- **Post Processing:** Combining the CR techniques either in parallel or series.

Input

The handwritten page is first scanned and from its scanned image, the images of three- character words are extracted. These images of the words are then saved in the database. The image scanner optically captures text images to be recognized.

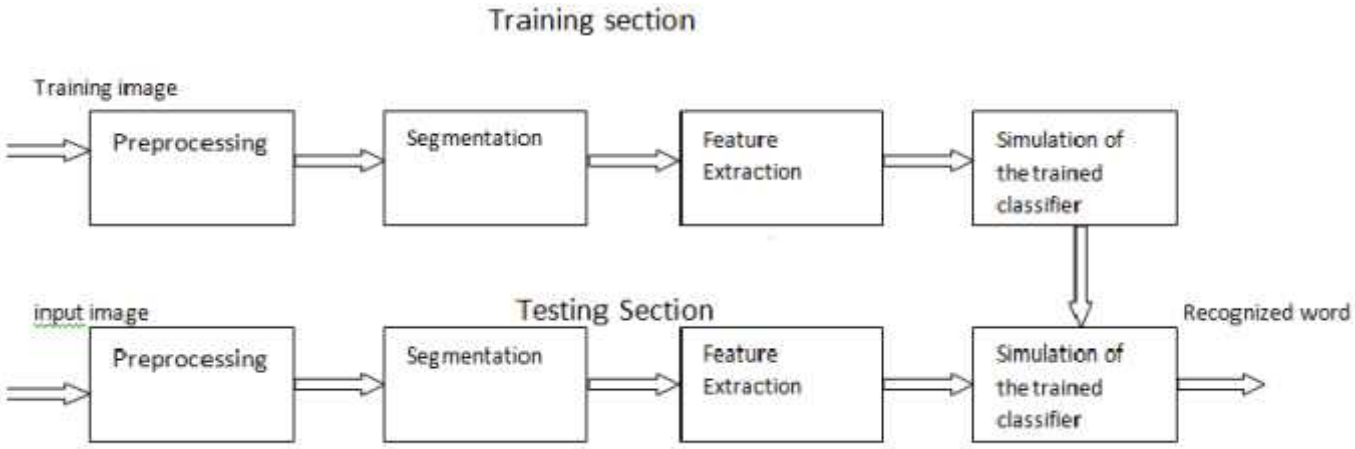


Figure 1: System Block Diagram: Offline Handwritten Character Recognition

Pre-Processing: These are the pre-processing steps often performed in Character Recognition:

Binarization: Usually presented with a grayscale image, binarization is then simply a matter of choosing a threshold value.

Morphological Operators: Remove isolated specks and holes in characters, can use the majority operator.

Segmentation

Check connectivity of shapes, label, and isolate. Segmentation is by far the most important aspect of the pre-processing stage. It allows the Recognizer to extract features from each individual character. In the more complicated case of Hand-written text, the segmentation problem becomes much more difficult as letters tend to be connected to each other. Extract sub-images that are vertically separate from their neighbours. These sub-images may contain more than one connected component. Select the sub-images that need further segmentation due to the presence of lower modifiers.

A selection criterion is required for selecting these:
- Separate the lower modifiers from the sub-images.
- Select the sub-images that contain conjuncts or shadow characters.
- Segment the conjunct sub-images into constituent consonant sub-images.
- Segment the sub-images of shadow characters into constituent character sub-images.

The Segmentation Process:
Segmentation is the most important step of analytical approaches employed to handwritten word recognition. Here, the cursive word is broken down into individual characters. Segmentation is the process of decomposing a word image into sub-images so that each image corresponds to a character. The process of segmentation is illustrated in figure below:

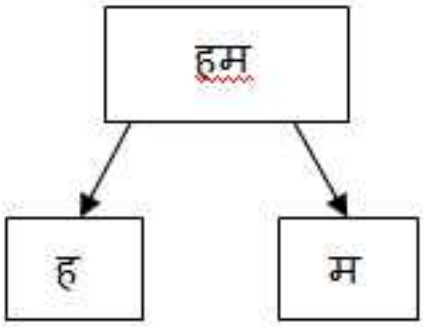


Figure 2: Segmentation - The process of dividing a word into its constituent characters

There are different methods for feature extraction or finding an image descriptor, these methods lie into two categories:
- One which uses the whole area of the image.
- Other that uses the contour or edges of the object.

All the above methods use the contour of the object to collect the object's features. Now segmentation for Hindi character:

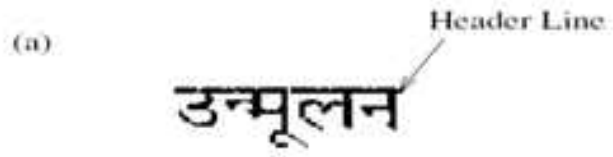


Figure 3 (a) Image of a Devanagari word

Figure 3 (b) Correct segmentation of the above word

Figure 3 (c) Image of the word after header line has been removed

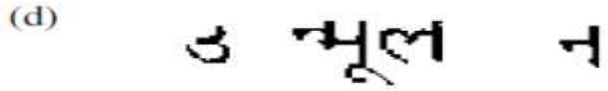


Figure 3 (d) The subimages that are vertically separate from their neighbours. These subimages may contain more than one connected component.

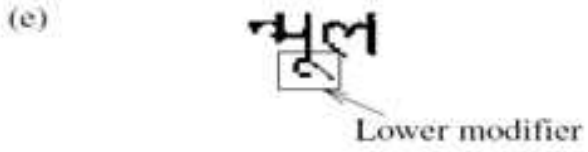


Figure 3 (e) The subimage that contains more than one connected component and needs further segmentation.

Figure 3 (f) The subimages after separating the lower modifier and extracting the subimages that are vertically separate from their neighbours.

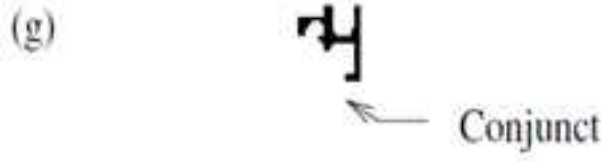


Figure 3 (g) The subimage contains two consonants that are touching and needs further segmentation

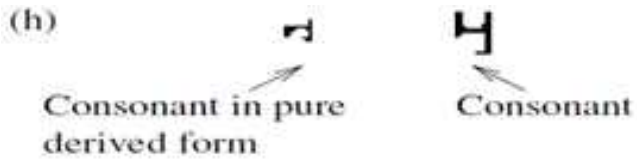


Figure 3 (h) The subimages after segmenting the touching consonants

Figure 3: Image of a Devanagari word and its segmentation

Extract subimages that are vertically separate from their neighbours. These subimages may contain more than one connected component.

Feature Extraction

Feature extraction is vital part of any recognition system and it is followed by segmentation process. In feature extraction stage each character is represented as a feature vector, which becomes its identity and also make the same character assume different appearance. The major goal of feature extraction is to extract a set of features, which maximizes the recognition rate with the least amount of elements. After detecting the individual symbols it is able to extract the general features such as width, height, closed shapes, diagonal lines, line intersections, special dots, etc Given a segmented (isolated) character.

Moment based features detection:
Think of each character as a PDF. The 2-D moments of the character are:

$$m_{pq} = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} x^p \; y^q \; f(x,y)$$

From the moments, we can compute features like:

- Total mass (number of pixels in a binarized character)
- Centroid - Center of mass
- Elliptical parameters
  - Eccentricity (ratio of major to minor axis)
  - Orientation (angle of major axis)
- Skewness
- Kurtosis
- Higher order moments
- Hough and Chain code transform
- Fourier transform and series

The segmentation constraint that has been taken is as follows:
2* Height / 3, where Height is the total height of the word image.Height is taken into consideration while deciding the segmentation points. Each word image is traced vertically after converting the gray scale image into binary matrix to find the first hit of a black pixel. If the black pixel lies below 2 * Height / 3 then it is a segmentation point. Thus we can say that segmentation points are decided depending on the row and column number of these black pixels.

Training of classifier:
Error Back Propagation algorithm
The operations of the network implementation in this project can be summarized by the following steps:

Training phase

i. Analyze image for characters
ii. Convert symbols to pixel matrices
iii. Retrieve corresponding desired output character and convert to Unicode Lineraize matrix and feed to network
iv. Compute output
v. Compare output with desired output Unicode value and compute error
vi. Adjust weights accordingly and repeat process until preset number of iterations

Testing phase

I. Analyze image for characters
II. Convert symbols to pixel matrices
III. Compute output
IV. Display character representation of the Unicode output

Procedure for classification of pattern

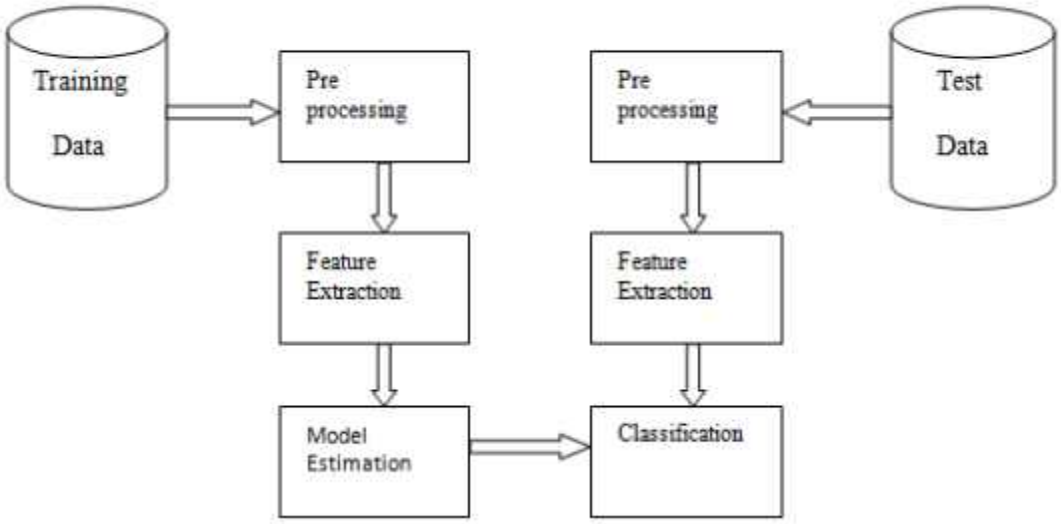


Figure 4: pattern classification

Steps include in classification:

I. Pre-processing
II. Feature extraction
III. Classification - Compare feature vectors to the various models and find the Closest match. One can use a distance measure

Algorithm of HMM Model for classification

There are two steps in building a classifier: Training and testing. These steps can be broken down further into sub-steps.

Traininig:

Various steps follow in training process:

1. In the training process firstly Gathered training examples, and images of words as a input and found instances of character images such as:
   क, क, क, क
2. **Feature extraction:** Input, Instances of character images and Output will be a sequence of 20-dimensional vectors per character instance
3. **Clustering and character model:** Input a sequence of observation symbols and character instance and will be a separately trained character HMMs for each character and get various transition probabilities for each character.
4. **Word model:** character transition probabilities, character HMMs and output word HMMs.

Prediction using a hidden markov model:

An image of word taken that is to be recognized and perform feature extraction and get a sequence of a 20-dimensional vector and computed centroid vector C during the training phase and perform discretization a sequence of symbols from a finite alphabet for the image of the word then viterbi decoding perform Input is a word HMM and a sequence of observation symbols and output,a sequence of states that most likely emitted the observation.

Prediction:

Input: sequence of states
क क क क

Output: sequence of characters
'क', 'क', 'क', 'क'

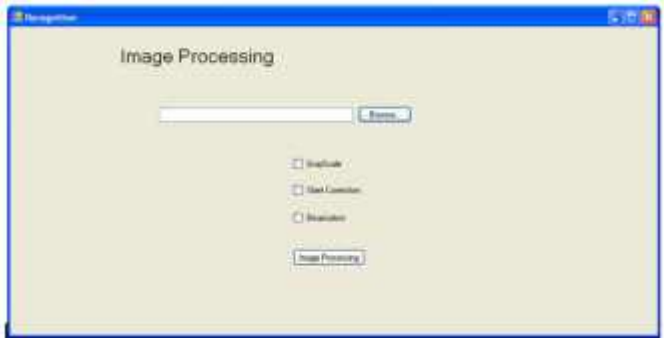
Postprocessing

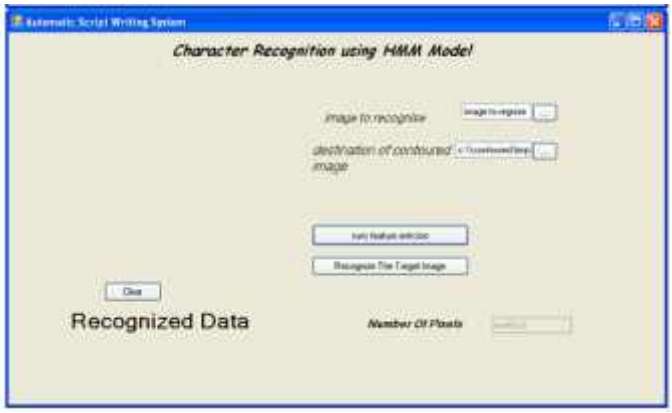
In this each vector is classified to one of the 64 classes. For each character the probability for each state to contain the vectors that it contains and only these, is first calculated according to its HMM model, by the relationship:

$$b_j(X) = \pi p_m * \pi(1 - p_n)$$

(where m the vectors that it contains and n the others) Then the probability of occurrence is calculated for each character by multiplying the probabilities for each state. Finally we choose the

character with the highest probability, in the case that we know for a field of the application form what we should expect, for example: Greek letters, English letters, numbers, three letters and four numbers etc, we choose the Greek letter with the highest probability or the English letter and so on then combining the CR techniques either in parallel or series, simulate the result and recognize the word as follows:





Conclusion

The BPN network designed proposed has the ability to recognize stimulus patterns without affecting by shift in position not by a small distortion in shape of input pattern. It also has a function of organization, which processes by means of Supervised Learning. If a set of input patterns are repeatedly presented to it, it gradually acquires the ability to recognize these patterns. It is not necessary to give any instructions about the categories to which the stimulus patterns should belong. The performance of the network has been demonstrated by simulating on a computer. We do not advocate that the network is a complete model for the mechanism of character recognition in the brain, but we propose it as a working design for some neural mechanisms of visual pattern recognition One of the largest and longstanding difficulties is in designing a pattern recognizing machine has been the problem how to cope with the shift in position and the distortion in the shape of the input patterns. The network proposed in this paper gives a partial solution to this difficulty .but the proposed work to recognize a single character of capital font or legible handwritten one For the sake of simplicity only BMP file is used to store the character to be recognized, It is limited to the DOS environment i.e. it cannot be executed in VC++ and Noisy and distorted character cannot be considered for recognition. Using a single HMM for each Hindi character (i.e., the full-character configuration) provided the best overall performance but at a much higher computational cost than sub-character models.

References

1. *P. Natarajan, Z. Lu, I. Bazzi, R. Schwartz, and J. Makhoul (2001), "Multilingual Machine Printed OCR," International Journal of Pattern Recognition and Artificial Intelligence, 15:1, pp. 43-63.*
2. *Z. Lu, R. Schwartz, P. Natarajan, I. Bazzi, and J. Makhoul (1999), "Advances in the BBN BYBLOS OCR System," Proc. of Intl. Conf. Doc. Analysis and Recognition, Bangalore, India, pp. 337–340.*
3. *P. Natarajan, R. Schwartz, M. Decerbo, T. Keller (2003), "Porting the BBN BYBLOS OCR System to New Languages," Symposium on Document Image Understanding Technologies, pp. 47-52.*
4. *G. R. Arce and R. E. Foster (1989), "Detail-preserving ranked-order based filters for image processing," January. IEEE Truns. Acoust., Speech, Signal Processing, vol. 37. pp. 83-98.*
5. *L. R. Rabiner (989), "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proc. IEEE, 77(2), pp. 257-286.*
6. *Veena Bansal and R. M. K. Sinha, "Segmentation Of Touching And Fused Devanagari Characters", Department of Computer Science and Engineering, Indian Institute of Technology, Kanpur 208016 India.*
7. *Su Liang, M. Shridhar and M. Ahmad (1994), Segmentation of Touching Characters in Printed Document Recognition, Pattern Recognition, 27, pp. 825-84.*
8. *R. G. Casey and E. Lecolinet (1996), A survey of Methods and Strategies in Character Segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 18, pp. 690-706.*
9. *Freund, Y. & Schapire, R. E. (1996) Experiments with a new boosting algorithm. in Proceedings of the Thirteenth International Conference on Machine Learning.*