OCR For Handwritten Marathi Script

Mrs.Vinaya. S. Tapkir[1], Mrs.Sushma.D.Shelke[2]
[1]Maharashtra Academy Of Engineering, Alandi (D), Pune, India
[2]Pune Institutes Of Computer Technology, Dhankawadi, Pune, India
(sdshelke@pict.ac.in, vstapkir@entc.maepune.ac.in)

Abstract: **Optical Character Recognition which is the original method of character recognition many times gives poor recognition rate due to error in character segmentation. Segmentation is an important task of any OCR system. It separates the image text documents into lines, words and characters. The accuracy of OCR system mainly depends on the segmentation algorithm being used. Segmentation of Handwritten Devanagari text is difficult when compared with Printed Devanagari or Printed English or any other Printed document its structural complexity and increased character set. It contains vowels, consonants. Some of the characters may overlap together. The profile based methods can only segment non-overlapping lines and characters. This paper addresses the segmentation of Handwritten Devanagari text document, the most popular script of Indian sub-continent into lines, words and characters. The proposed algorithm is based on projection profiles. Experimental results it is observed that 100% line segmentation and about 98% character segmentation accuracy can be achieved with overlapping lines, words and characters.**

Keywords: **segmentation, projection profiles, features extraction, zoning features, projection histogram features, Euclidean Distance classifiers**

I. INTRODUCTION

Optical character recognition (OCR), is a program that translates scanned or printed image document into a text document. For certain language script today, it is not difficult to develop an optical character recognition (OCR) system that recognizes well-shaped characters with accuracy of 99% and above. In order to recognize the text contained in a document, it is usually segmented into lines, words, and characters.

The typical phases [2] of an OCR system are

1. Pre-processing
2. Segmentation
3. Feature extraction
4. Classification

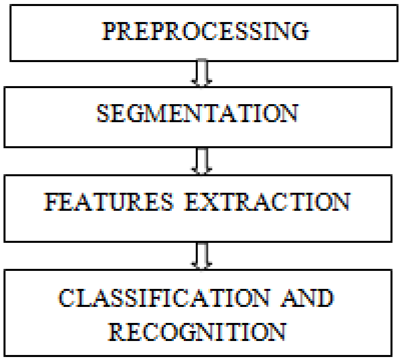


Fig.1. Steps of OCR System

Segmentation phase includes the segmentation text image into lines, word and characters. The final recognition phase consists of feature extraction, selection and classification. Actual processing takes place on the binary images. Binary image separates the foreground pixels from the background. For binarization Otsu [3] method can be used. In this a threshold value is selected and the intensity values above the threshold are converted into one intensity value (white) and below the threshold are converted into another intensity (black). The scanned documents may contain noise. To process the document noise is to be removed. This preprocessed image is given as input to the segmentation stage. In this stage the binary image is separated into lines, words and characters. There are several segmentation methods as discussed. The extracted characters are then given as input to the feature extraction and recognition phase to recognize and classify the characters. The Fig. 1 shows the steps of a typical OCR system.

II. METHODOLOGY

1. Preprocessing

The pre-processing stage takes a raw image then following operations are applied on it.

1.1Thresholding

Raw image either colour or grey is converted into binary image.

1.2Noise reduction

Various techniques like morphological operations are used to connect unconnected pixels, to remove isolated pixels, to smooth pixels boundary.

1.3Normalization

The character segmented image is normalized to 32*32 or 64*64 matrix.

III. RESULTS

1. Preprocessing

Preprocessing is the first step of OCR. Here results of preprocessing are shown on handwritten Marathi character 'A'.

1.1 Thresholding

This is the first preprocessing stage. Gray or color image is taken as a input to convert into binary image, based on threshold. The output image obtained replaces all pixels in the input image with luminance greater than threshold level with the value 1 (white) and replaces all other pixels with the value 0 (black). Inverted binary image performs inverse operation, replaces 1's with 0's and 0's with 1's.In proposed algorithm 0.9 threshold is found good for conversion. Fig. 2shows the inverted binary image of character 'A'.

1.2 Noise Reduction And Normalization

Binary image exists some noise which has to be removed for further operations that is segmentation and features extraction. Different Morphological operations are required for noise reduction. In proposed system area open and clean operations are applied over image. Morphologically open binary image removes small objects or all connected components fewer than threshold pixels, brings accuracy in features extraction. Clean operation removes isolated pixels individual 1s that are surrounded by 0s that is small dots from image which is beneficial to bring accuracy in segmentation. After noise reduction unwanted 0's around the character are removed by cropping and image is normalized to 32*32 or 64*64 as per need and used for features extraction after segmentation. Fig.3 shows inverted cropped image of handwritten Marathi character 'A' which is normalized to 32*32 after noise reduction.

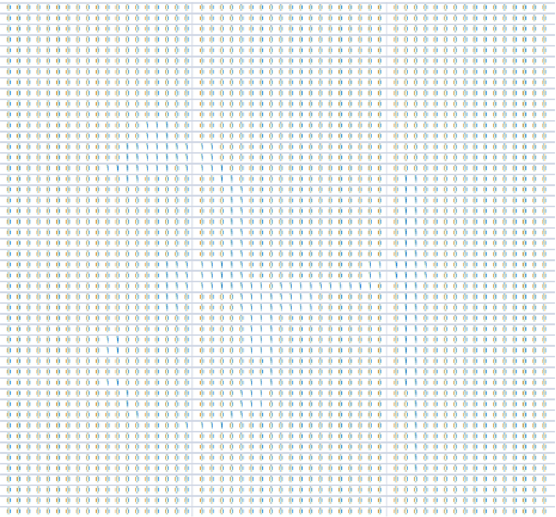
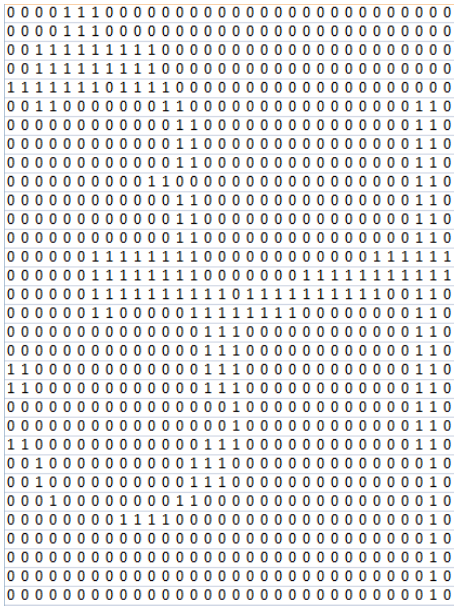


Fig. 2. Inverted Binary 48*54 Image of 'A'.



```
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1
0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1 0
0 0 0 0 0 0 1 1 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
```

Fig.3. Inverted Cropped Binary 32*32 Image Of Character 'A'

2. Segmentation

Segmentation results using projection profile are discussed here.

2.1 Line segmentation

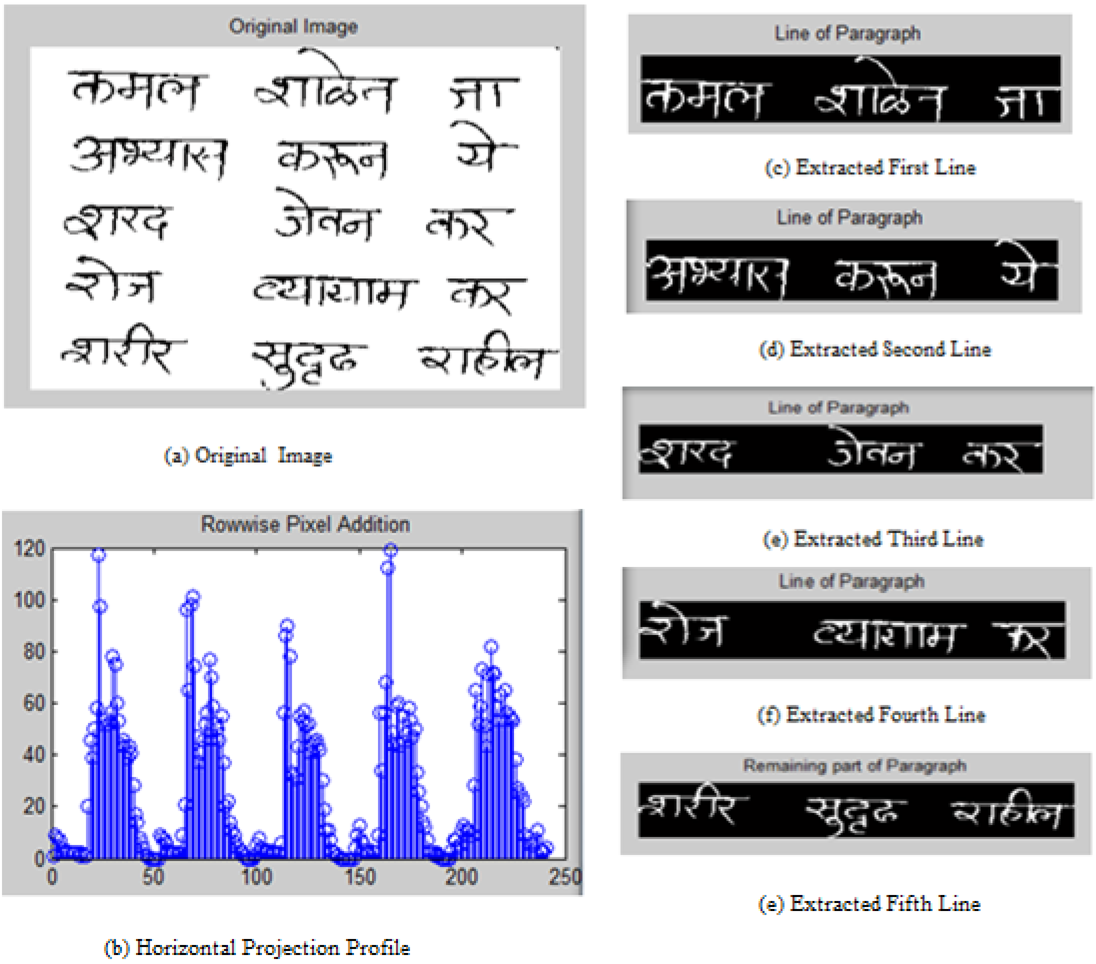
The text lines are detected by finding the valleys of the projection profile computed by a row-wise sum of black pixels. This profile shows large values for the headline of the individual text line. The position between two consecutive headlines, where the projection profile height is minimum , denotes the boundary between two text lines. The Fig. 4 shows sample text image, horizontal projection profile and the extracted text lines using Projection Profile.

2.2 Word segmentation

To segmenting the text line image into words, compute vertical projection profiles. The projection profile is the histogram of the image. In the profile, the zero valley peaks represent the word space. Segmented line from first stage is taken as input for second stage that is word segmentation. Line is then segmented into word in this stage. The Fig. 5 shows sample extracted Line image, Vertical projection profile and the extracted words using Projection Profile.

2.3 Character Segmentation

Character segmentation is done after the individual words are identified. To extract character from word removal of headline is essential. For this first horizontal projection of individual word is computed and the rows having highest projection is consider as a headline and removed for further character segmentation. After removal vertical projections of individual word is computed. Using these profiles separation of the base characters is done. The Fig. 6 shows sample word image, headline removed image, vertical projection profile and the extracted characters using Projection Profile.



(a) Original Image

(b) Horizontal Projection Profile

(c) Extracted First Line

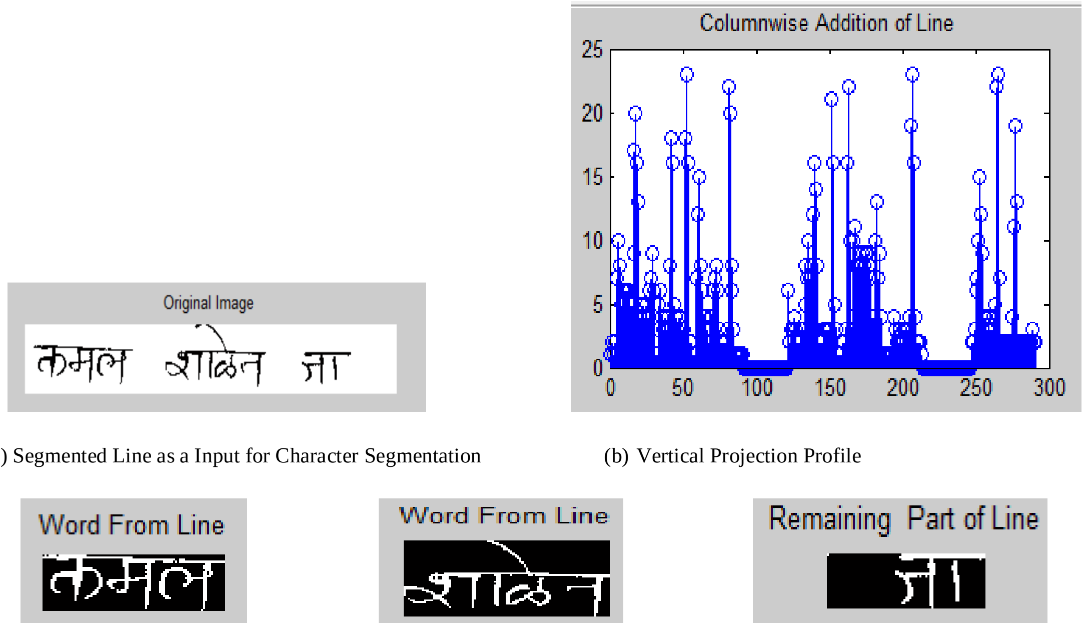
(d) Extracted Second Line

(e) Extracted Third Line

(f) Extracted Fourth Line

(e) Extracted Fifth Line

Fig. 4) Steps of Segmentation of Paragraph into lines



(a) Segmented Line as a Input for Character Segmentation

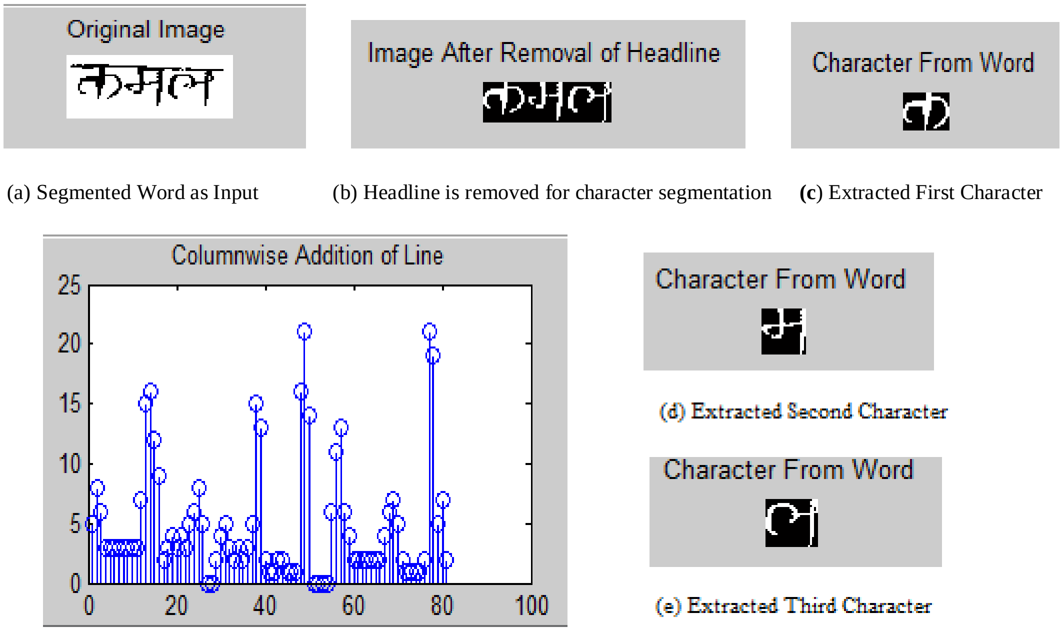
(b) Vertical Projection Profile

(c) Extracted First Word

(d) Extracted Second Word

(e) Extracted Third Word

Fig. 5) Steps of Segmentation of Line into Words



(a) Segmented Word as Input

(b) Headline is removed for character segmentation

(c) Extracted First Character

(d) Extracted Second Character

(e) Extracted Third Character

(f) Vertical Projection Profile

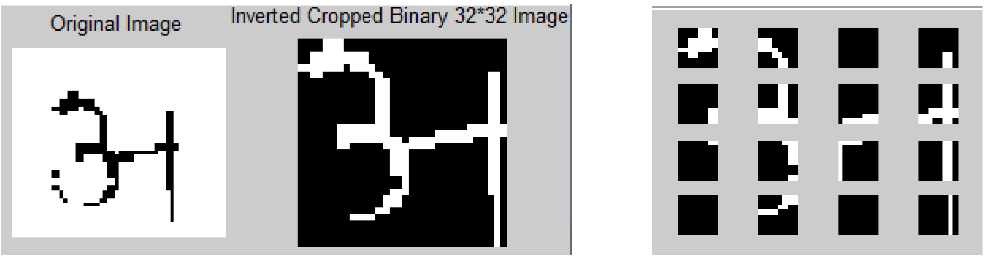
Fig. 6) Steps of Segmentation of Words into Characters

3. *Features Extraction*

Results of Features Extraction using Projection Histogram and Zoning are discussed here.

3.1 Pixel Density Features

Created 16 (4*4) zones of 32*32 sized sample. By dividing the number of foreground pixels in each zone by total number of pixels in each zone i.e. 64, the density of each zone is obtained as, pixel den = 0.4219, 0.2500, 0, 0.0938, 0.0938, 0.4688, 0.1563, 0.4531, 0.0313, 0.2031, 0.1875, 0.2500, 0, 0.1875, 0, 0.1250. Hence 16 zoning density features are obtained as shown in Fig. 7.



(a) Character Image of size 32*32

(b) 16(8*8) Zones of Character

Fig.7. Pixel Density Features

4. Euclidean Minimum Distance Classifier

At this step, in order to recognize the character, a distance between unknown or input feature vector X and the reference vectors, M from the training set must be computed. The distance will be computed based on Euclidean model as:

$$d(X,M_k) = \sqrt{\sum_{i=1}^{N}(x_i - m_i^k)^2}$$

The experimental result in Table 1 obviously shows that classification based on multiple feature gives more accuracy than using only one feature information.

Table 1. Experimental Results

Feature Extraction	Percentage of Recognition
Pixel Density(zoning)	92.77

V. CONCLUSION

In this experiment, the proposed algorithm is tested with several document images. Some of the documents contained overlapping lines and characters. Even though it could segment all the documents in a robust way and gave good results. But, it couldn't segment the touching lines and characters. The broken characters have been over segmented. Segmentation of the touching lines and characters may require some heuristic approaches.

REFERENCES

[1] U. Pal, B.B. Chaudhuri. (2004): "Indian script character recognition: a survey, Pattern Recognition", 37,1887 – 1899.

[2] B. Anuradhaand, Arun Agarwal and C. Raghavendra Rao. (2008): "An Overview of OCR Research in Indian Scripts", IJCSES, Vol.2, No.2.

[3] N. Otsu. (1979): "A threshold selection method from gray-level histograms", IEEE transactions on systems, Man and Cybernetics, Vol. Smc-9, No. 1.

[4] Satish Kumar, "An Analysis of Irregularities in Devanagari Script Writing – A Machine Recognition Perspective", (IJCSE) International Journal on Computer Science and Engineering Vol. 2, No. 2, 2010, 274-279,ISSN:0975

[5] C. V Lakshmi, C. Patvardhan. (2004): "An optical character recognition system for printed Telugu text, Pattern Analysis & Applications", Volume7,pp.190-204.

[6] optical character Recognition", Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition

[7]R.C. Gonzalez and R.E. Woods. (2004): Digital Image Processing, Pearson Education. Nitin Bhatia and Vandana. (2010): "Survey of Nearest Neighbor Techniques", IJCSIS

[8]C V Lakshmi, C PAtardhan "A Multi-font OCR System for printed Telugu Text.", Proceeding of LEC'02, IEEE, 2002

[9] Sushama Shelke, Shaila Apte, "Multistage Handwritten Marathi Compound Character Recognition Using Neural Networks" , Journal of Pattern Recognition Research 2 (2011) 253-268