Segmentation of Devanagari Handwritten Characters

Ankita Srivastav
Student: Department of CSE and IT
North Cap University
Gurgaon, Haryana

Neha Sahu
Assistant professor:
Department of CSE and IT
North Cap University
Gurgaon, Haryana

ABSTRACT

The world is fast moving towards digitalization. In the age of super-fast computational capabilities, everything has to be made digitalized so as to make the computer understand and thereby process the given information. Optical character recognition is a method by which the computer is made to learn, understand and interpret the languages used and written by the human beings. It provides us a whole new way by which computer can interact with human beings, in their own languages. Hence OCR has been a topic of interest for researchers all around the globe in the past decade and research paper involving OCR is increasing day by day. It is seen that efficient algorithms have increased the speed and accuracy of character segmentation and recognition. A substantial amount of work has been done on foreign languages such as English , Chinese etc. but very few paper are there for Indian languages baring a few for Hindi and Bengali. Hence our research work was directed towards development of a novel algorithm for Devanagari character segmentation for Hindi. Hindi is one of the eighteen languages recognized by the Indian constituency. It is also one of the oldest languages and is spoken by millions of people in India. Segmentation and Recognition of this particular language is difficult because of the presence of complex connected characters and presence of shirorekha. A novel approach has been proposed for the segmentation of the connected character.

General Terms

Pattern recognition, image segmentation.

Keywords

OCR, Shirorekha, Devanagari character segmentation.

1. INTRODUCTION

Optical Character Recognition is an upcoming research topic. Numerous amount of work has been done on this filed. Character segmentation is necessary pre-processing step of OCR. The accuracy of OCR systems is directly proportional to the accuracy of Segmentation. Character segmentation is also the most well studied field over last few decades. The foremost purpose of this is to provide discrete characters to Optical Character Recognition. A tremendous advancement has been done in the fields of camera based applications and mobile phones, hence in order to increase the accuracy of character recognition the efforts must be done on character segmentation to segment the words in to individual characters to accelerate the results of recognition process.

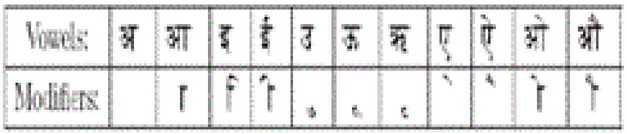
The segmentation has been done in many languages like English, Chinese etc. But the segmentation of Devanagari characters is still an intricate task. Devanagari is the script in

Which we represent Hindi, Nepali, Sanskrit languages. Hindi is the world's third widely used language all over world. This paper proposed a new approach to segment Devanagari handwritten characters. During segmentation, we examine the values stored in pixels and after that we perform the segmentation using bounding box technology.

2. FEATURES OF DEVANAGARI SCRIPT

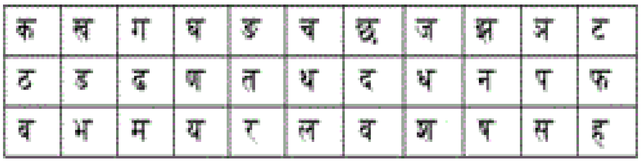
Devanagari script forms the base of many languages like Hindi, Nepali, and Sanskrit etc. The script is written from left side to right side and there is a header line which combines each character to form words. There are upper and lower modifiers that make the segmentation task more difficult. There are 33 consonants and 11 vowels present in Devanagari characters. Vowels can be written as independent letters or by using them above, below, before or after the consonant they belong to. When the vowels are written in this way they are known as modifiers and the characters so formed are known as conjuncts. Two or more consonants can be combined together to form compound characters [1] [2].

Table 1. Vowels And Corresponding Modifiers [1][2]



Vowels:	अ	आ	इ	ई	उ	ऊ	ऋ	ए	ऐ	ओ	औ
Modifiers:		ा	ि	ी	ु	ू	ृ	े	ै	ो	ौ

Table 2. Consonants [1][2]



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

The vowels of Devanagari characters can be written as independent letters. If we write vowel as independent letters, we call them "Modifiers". The characters that are formed by using modifiers are known as Conjuncts. Sometimes two or more consonants make new shaped characters that are known as compound characters. Devanagari script has rich set of conjuncts, hence the segmentation of these characters become a complicated task. The characters have two kinds of modifiers, Upper and Lower modifiers (as shown in figure 1). Due to these modifiers, also the task of segmentation becomes an intricate.

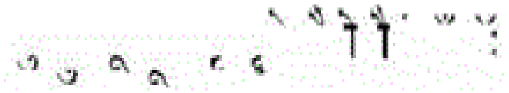


Figure 1. Upper And Lower Modifiers [1][2]

The important feature of Devanagari script is that it is written from left to right and the header line that we called as "Shirorekha" or "Maatra" appends all the letters to form a complete word (as shown in figure 2).

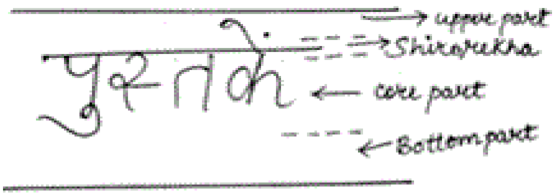


Figure 2. Figure Showing Devanagari Characters Different Parts

The above figure shows the various parts of Devanagari characters. There is a Shirorekha, which appends every character to form a complete word. The upper and core part is separated by the shirorekha and there is an un-seeable baseline which separates the core part and lower modifiers.

3. PROPOSED SEGMENTATION APPROAH

The various algorithms involved in each step of proposed system are discussed in this section. The proposed approach used for the segmentation of the handwritten Devanagari words is shown in figure 3.

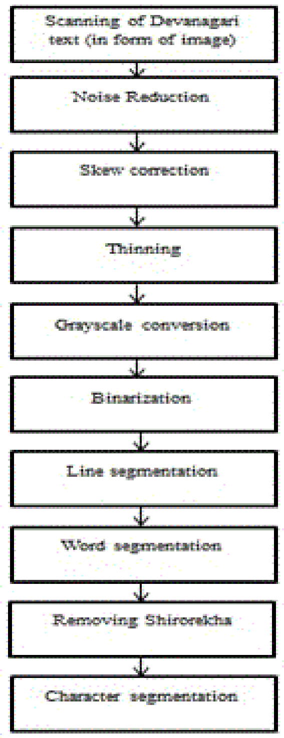


Figure 3. Proposed Methodology

3.1 Scanning of Documents

In the first step, the handwritten Devanagari text is scanned by using scanner. While scanning of the image few deviations get added like skew, noise. For the accurate segmentation these deviations need to be removed.

3.2 Noise Reduction

During scanning of the document, noise gets added which needs to be removed to improve the segmentation process. Filled loops, disjointed lines and gaps between the lines, rounding of corners, etc. gets added which make the task troublesome. For better segmentation, noise should be removed from the image. Noise is removed by using medfilt2. Medfilt2 filtering is used to remove salt and pepper noise and preserve edges. It performs two dimensional median filtering [1] [2].

3.3 Skew Correction

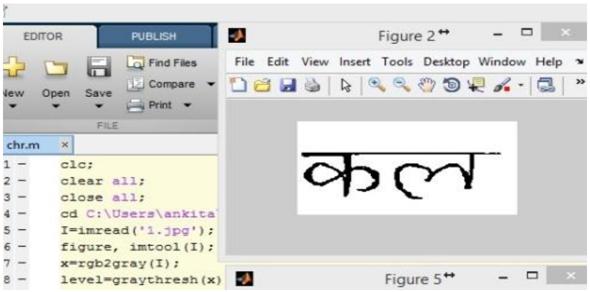
Skew must be detected and corrected to improve the segmentation process as it radically diminishes the precision of segmentation. Skew is the twist that is introduced while scanning the document. If a text line makes an angle with reference to the horizontal line then it is known as skew angle. It is corrected by calculating the skew angle and then rotating the image in reverse direction by the same amount of skew angle.

3.4 Thinning

It is a morphological operation through which we remove selected foreground pixels from binary images. It's behavior is determined by structuring elements. It is done to detect pertinent features and objects of interest [1] it is corrected using bwmorph morphological operation. It is used to remove unnecessary pixels.

3.5 Grayscale conversion

The image is scanned as color image or RGB image; it needs to be converted into the grayscale image. A predefined function is available in MATLAB, rgbtogray to convert color image into grayscale image. The image scanned may be a color image so it needs to be converted into graysacle image of 2X2 matrix by using the function RGBtoGRAY. The conversion is shown in figure 4



```
clc;
clear all;
close all;
cd C:\Users\ankita
I=imread('1.jpg');
figure, imtool(I);
x=rgb2gray(I);
level=graythresh(x)
```

Figure 4. Grayscale conversion

3.6 Binarization

After grayscale conversion, the grayscale image is again converted into binary image or two-tone image. The binary image has only has two values 0 or 1. 0 represents object, and 1 represents background, Or 0 is for representing white and 1 is for representing black.

The ostu's global thresholding method is used to convert grayscale image into binary image. The binary conversion is shown in figure 5.

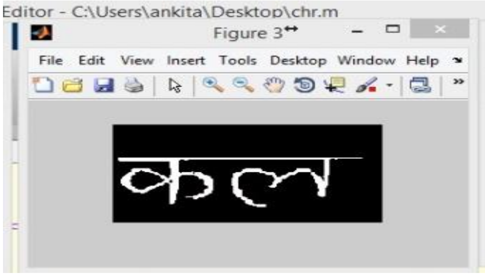

Editor - C:\Users\ankita\Desktop\chr.m
Figure 3
File Edit View Insert Tools Desktop Window Help

Figure 5. Binarization

3.7 Line Segmentation

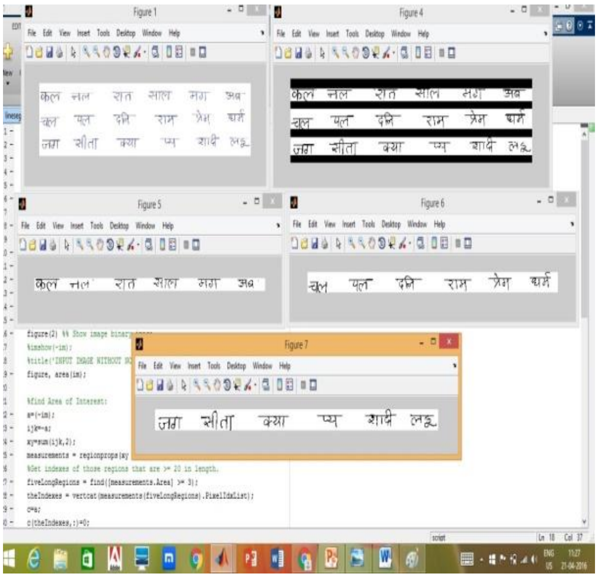
In this step the image is scanned horizontally, pixel – row by pixel – row from left to right and top to bottom. And for each pixel intensity is evaluated [6]. The space among the text lines are calculated and bounding boxes are placed for each text line.

Following algorithm has been proposed for line segmentation.

i) Invert the binary image and count the no of black pixels in each row.

ii) Find the rows that contains maximum no of black pixels and replace all those rows with 0.

iii) Apply bounding box for text lines using MATLAB functions, region props and rectangle functions.



```
figure(2) %% Show image binary
%imshow(~im);
%title('INPUT IMAGE WITHOUT NO
figure, area(im);

%find Area of Interest:
a=(~im);
ijk=~a;
xy=sum(ijk,2);
measurements = regionprops(xy
%Get indexes of those regions that are >= 20 in length.
fiveLongRegions = find([measurements.Area] >= 3);
theIndexes = vertcat(measurements(fiveLongRegions).PixelIdxList);
c=a;
c(theIndexes,:)=0;
```

Figure 6. Line segmentation

3.8 Word Segmentation

Afterwards the image is scanned vertically, pixel – row by pixel – row from left to right and top to bottom. The spaces among words are calculated and bounding box is placed for each word [6].

The proposed algorithm for performing word segmentation is as follows

i) Invert the binary image and count the no of black pixels in each column.

ii) Find column that contains maximum no of black pixels and replace all those rows with 0.

iii) Apply bounding box for text lines using MATLAB functions, region props and rectangle functions.

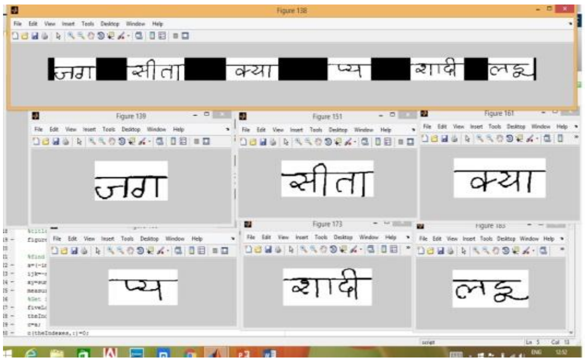


Figure 7. Word segmentation

3.9 Character segmentation

This is the last step of segmentation process. Words are separated into characters and bounding box is plotted for each different Devanagari character.

The proposed algorithm for character segmentation is as follows

i) Invert the binary image and calculate the sum of number of pixels containing row wise. In addition, store the sum column wise.

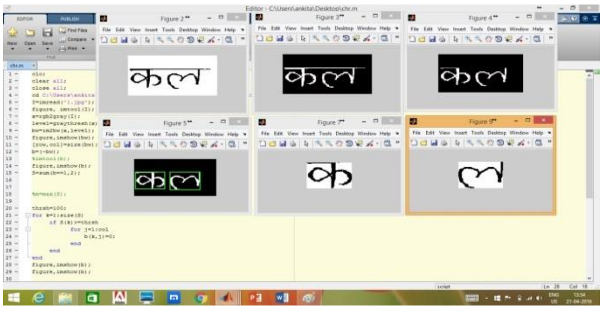
ii) To remove shirorekha, set a threshold value.

iii) Compare the sum that is stored column wise to the threshold value.

iv) If shirorekha is present then the threshold value will have lesser value as compared to the sum of the pixels along that row.

v) The rows that contain values greater than threshold is changed with 0.

vi) Apply the bounding boxes for each character using MATLAB functions region props and rectangle.



```
clc;
clear all;
close all;
cd C:\Users\ankita
I=imread('1.jpg');
figure, imtool(I);
a=rgb2gray(I);
level=graythresh(a)
bw=im2bw(a,level);
figure,imshow(bw);
[row,col]=size(bw);
b=(~bw);
%imtool(b);
figure,imshow(b);
S=sum(b==1,2);


%m=max(S);

thresh=100;
for k=1:size(S)
    if S(k)>=thresh
        for j=1:col
            b(k,j)=0;
        end
    end
end
figure,imshow(b);
figure,imshow(b);
```

Figure 8. Character segmentation

4. RESULTS

Different documents consisting of different number of Devanagari lines, words and characters were assembled and examined. The observations include that line and words segmentation are performed with accuracy of approximately 100%. But when character segmentation is performed using the algorithm explained above, it is able to segment characters and modifiers easily but not able to segment connected characters properly, so character segmentation is done with approximately 90% accuracy.

Table 3. Net Segmentation Result of document used in above figures

Line Segmentation	**Lines in Document**	**Recognized Lines**	**Accuracy**
	5	5	100%
Word Segmentation	**Words in Document**	**Recognized Words**	**Accuracy**
	25	25	100%
Character Segmentation	**Characters in Document**	**Recognized Characters**	**Accuracy**
	98	108	90%

5. CONCLUSIONS

This paper has proposed algorithms for segmenting Devanagari handwritten script consisting of lines, words and characters. The line segmentation and word segmentation have achieved successful 100% accuracy. But in character segmentation, the accuracy achieved is about 90% this is because the algorithm is not able to properly segment some of the connected words. Besides connected characters the algorithm is efficient in segmenting characters, modifiers and purnviram (full stop in Devanagari).

6. FUTURE SCOPE

The following issues comprise the Future Work in this area:

i) Segmentation of connected components

Different approaches will be used in future to deal with above mentioned issues and solve them with 100% accuracy.

7. REFERENCES

[1] Ambadas B. Shinde and Yogesh H. Dandawate “Shirorekha Extraction in Character Segmentation for Printed Devanagari Text in Document Image Processing”, 2014 Annual IEEE India Confeernce (INDICOM).

[2] Vijay J Dongre and Vijay H Mankar, “Segmentation of Devanagari Documents”, Communications in Computer and Information Science, Springer, Vol. 198, pp. 211-218, 2011

[3] N.Vishwanath, S.Somasundaram, M.R.Rupesh Ravi and N.Krishnan Nallaperumal “Connected component Analysis for Indian Lisence Plate Infra-Red and Color Image Character Segmentation”, IEEE International Conference on Computational Intelligence and Computing Research, 2012.

[4] Vijay Kumar and Pankaj K. Senegar, “Segmentation of Printed Text in Devanagari Script and Gurumukhi Script”, International Journal of Computer Applications(IJCA), Vol.3, pp. 24-29, 2010.

[5] Rapeeporn Chamchong, Chun Che Fung,” A Combined Method of Segmentation for Connected Handwritten on Palm Leaf Manuscripts”, 2014 IEEE International Conference on Systems, Man, and Cybernetics October 5-8, 2014, San Diego, CA, USA.

[6] Namrata Dave, “Segmentation methods for Handwritten Character Recognition”, International Journal of signal processing, image processing and pattern rcognition, vol 8no4(2015),pp.154-164 http://dx.doi.org/10.4257/ijsip.2015.8.4.14