2015 International Conference on Computational Intelligence and Communication Networks

Removal of Obstacles in Devanagari Script for Efficient Optical Character Recognition

Vivek Kumar Verma,
Dept. of Computer Science & Engineering
Maniapl University Jaipur, India
vermavivek123@gmail.com

Pradeep Kumar Tiwari,
Dept. of Computer Science & Engineering
Maniapl University Jaipur, India
pradeeptiwari.mca@gmail.com

***Abstract*— There are different defining problems are still need to resolve for the high accuracy of Optical Character Recognition (OCR) of Devanagari script. The more prominent number of font availability in Devanagari script is a challenge for the character detection in India. Therefore to conquer this problem, here is proposed a system for identification of diverse font of characters which helps to improve the OCR system accuracy. The adequacy technique for recognition of character for number of font styles in Hindi script is shown by this methodology. The proposed methodology is suitable for handling script styles information for Devanagari script is indicated by the outcomes. A procedure for enhancing the recognition accuracy of Devanagari OCR System by creating idea for discovery emphasis words such as Bold, Italic and underline words is exhibited in this work. We have taken most generally utilized Devanagari font styles, for example, kruti Dev 714, DevLys 240, and Alekh for our benchmark testing. Recognition of text style in Hindi script record enhances the execution of Hindi OCR framework.**

***Keywords*—Devanagari OCR, font detection, character segmentation, projection profile**

I. INTRODUCTION

Text Optical Character Recognition has grown at vast speed during last few decades. It has emerged as very important technology in Computer science field especially in image processing and recognition. There are lots of systems that perform vast number of applications for OCR although the systems are not able to compete perfectly today also with human reading capabilities [1].

Optical Character Recognition (OCR) is a technique which converts image of printed, scanned text into a text understandable by machine and encode the character images into standards of ASCII or Unicode format. It enable human to feed the data directly into an electronic computer file, and allow making changes in the file using a word processor. Optical character recognition that works on optical tools like mirrors and lenses and digital character recognition that works on scanners and algorithms were earlier considered as the separate field. Early systems required training to read a specific font but now ICR systems using feature extraction method have a high degree of recognition and can read many fonts successfully [2]. Now in present scenario the systems can produce the output that is formatted and closely resembles the original image of scanned page. But this approach is sensitive to the type and size of different fonts. The process is complex for handwritten character. Now even the techniques of soft computing are used in OCR because it has good ability of recognition and mapping process.

Document analysis and document understanding is a great challenge in information processing. To make the computer understand the abstract meaning of a document perceived by human beings is not only the purpose of artificial intelligence but also the way to make the computer interface more friendly. Recently, various technologies of document information processing including optical character recognition and online index searching have been developed and utilized to transform a document into digital or electronic form. They can be utilized to realize office automation and digital library and provide lots of precious data for users. However, how to find out the claimed and useful data from the large database is the problem to be resolved. Information retrieval and data mining provide a solution by using the techniques of keyword matching and association rule. They can reduce the amount of searching candidates, but sometimes they will result in some ridiculous and useless searching.

The information of font type and font style not only can improve the accuracy of optical character recognition but also can be served as the reference in selecting feature words to analyze the contents of documents. The prominent font style or font type can be treated as the message to convey what the author wants to express. It is helpful especially for the content analysis. Latent semantic analysis represents the content of each document by the feature vector constructed from the appearing frequencies of feature words. To achieve the goal, principal component analysis is performed to calculate the priority of each element in the feature vector and the abstract meaning of each document is then transformed into a feature vector. In this way, the comparing of contents of documents is made to be possible.

Optical character recognition (OCR) method has been used in converting printed text into editable text form. OCR is very useful and popular technique in many applications. The output accuracy of OCR depends on text pre-processing and also segmentation algorithms. In some cases it is difficult to extract text from the image because of multiple sizes, style, orientation; complex background of image etc.One of the challenge is detection of font in Devanagari script. Problem is detection of bold and italic and underline character in Devanagari script for the purpose of use in OCR. This paper describes Boldface, italic and underlies font recognition method using stroke pattern analysis on segmented word

978-1-5090-0076-0/15 $31.00 © 2015 IEEE
DOI 10.1109/CICN.2015.90

images. The word images are extracted from scanned text documents containing word objects in various fonts and styles. In Earlier work font recognition methods are mainly focus on slanted texture or pattern analysis on single character or a text block; those are sensitive to noise for font and style. We are mostly focus on the most common fonts styles used in the Devanagari scripts such as DevLys 240, kruti Dev 714, Alekh etc.

II. Properties of Devanagari Script

Devanagari script is written from left to right and in top-bottom manner. It contains total 11 vowels and 33 basic consonants. Except the first vowel, each one have corresponding modifier which is added to a consonant [3]. The word written in Devanagari script is a continuous line of black pixels which is called "Sirorekha". Each character is divided into three different parts based on sirorekha. The sub-part which is above sirorekha is known as upper modifier. The second part consists of characters and the third part consists of modifiers of vowels which are known as lower modifiers. Some of the characters also combine to form a new character set which we called it as, joint character. It may sometime happen that a character is in the shadow of another character. This disturbance can be because of lower modifier or the shapes of characters that are adjacent to each other, due to words showing header lines, words that have lower modifiers. Words that has shadow characters, Words that has composite character, and Character that has different height and width all are scanned and recognized by OCR system today.

Devanagari is highly complicated and the reason its high complexity is its conjuncts. This language is considered as partly phonetic language in which the words written in Devanagari script has a one way pronunciation , but it is not necessary that all the pronunciations can be written perfectly. A syllable or in general we say it as "Akshara" is made up of a vowel itself or it can be a combination of consonant and a vowel [4].

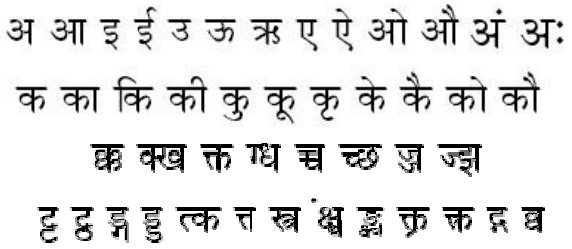

Figure1: Some of the vowels and consonants with modifiers and compound characters [4]

In the last few years, significant and noticeable work has been done in OCR field. Devanagari Optical Character Recognition is considered as one of the toughest step in the digitization of Indian literature. OCR is the process by which scanned images of characters are read electronically and they are converted into an editable text form or machine readable form. The document that is scanned is converted and stored in the form of bmp file extension or bitmap files when these images are recognized they are converted from text files in bmp form into corresponding ASCII or UNICODE format that is readable by compute. Generally the text that has been generated by OCR is considered as input into text search database which is used in reading the forms, manuscripts and their archival from old database which is also applied by searching in library [5].

First of all the words written in Devanagari script are divided into the composite characters and then each character is divided further into a set of symbols. The symbol consist of a composite Devanagari character, a modifier symbol – it can be upper or lower, or a Devanagari alphabet. These symbols are decomposed which are further recognized by using prototypes and are constitute a valid word of Hindi language. The symbols which are not recognized as valid symbols are ignored and considered as rejection and substitution errors.

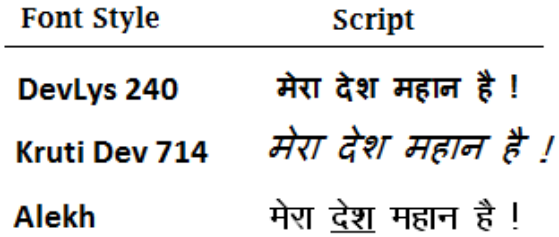

Figure2: Font styles of Devanagari script

OCR for Devanagari script becomes even more challenging when these compound characters and the characteristics of modifier are combined under 'noisy' situation. It is hard to identify the characters in different font styles. We are mostly focus on the most common fonts styles used in the Devanagari scripts as shown in above Fig.2. Here Devanagari script DevLys 240 font shown with Boldface, kruti Dev 714 shown with italic face and, Alekh with underline.

III. Literature Review

As Ravi Kant Yadav et al. [6] proposed an approach for the "Detection of Bold and Italic Character in Devanagari Script". The objective of this approach is to develop font detection system for machine printed characters. The approach based on the feature extraction of the various fonts. It is assumed that the document contains a single character and is good quality, noise free and less distortion. The process distinguishes between the normal and bold character accurately. In this way it is make a comparison of on pixel for vertical and horizontal line of the character. Each of the character has a characteristic that it has a horizontal line at its top; this line is called, Headline. It is examined that, in case of bold character, the thickness of headline is not the same as that of the rest part of the character. In this method pixels are compared and for this firsts scan the character and save it in .jpg format, Binarize the character for off & on pixel, Read the format of the character and store it in an array, for calculation of black pixels, Calculate number of on pixels in the head line of the character, Calculate number of on pixels in vertical line of the character and finally if the number of on pixel calculated in is more than previous one, then the character is bold. Now in this paper analyze bold and regular font as first check vertical line of regular & bold is always constant, Head line of regular & bold are always varied, For regular character the value of black pixel in header line is greater or equal to vertical line, For bold character the value of black pixel in vertical line is greater than header line. In this method few of the limitation there as the condition of the number of black pixels in vertical line of the character should be greater than the number of black pixels in the header line of the character" is not followed by some characters. So the conclusion, according to practical result, some characters can't be detected as a bold character on the basis of this

algorithm. Another side condition of the number of black pixels in vertical line of the character should be less than the number of black pixels in the header line of the character is varying for different size of characters. So the conclusion, according to practical result, detection of italic character on the basis of this algorithm is varying and not having proper value.

Yogendra et al. [7] used various morphological image processing techniques to detect bold text. The binarized image is thinned progressively, removing pixels alternatively from both the ends of the stems of a character. Each thinning operation is followed by opening the thinned image using a vertical structuring element. This leads to the reduction in the number of "ON" pixels present in the image after-each iteration. The ratio between the numbers of "ON" pixels in the present to the previous iteration is calculated after-each iteration. If this ratio is below 40%, the process is stopped and the resultant image is used to determine the positions of the bold text in the given page.

Harjit Singh et al. [8] proposed work for Italic Detection of English Character. This Approach Includes, the virtual strokes embedded in the considered character image. After extraction of characters, shear transformation is operated on the considered character image. The all alphabets of Latin script including upper and lower are classified into three classes based on the structural information of the extracted virtual strokes. Basically the italic and non-italic characters can then be distinguished based on the classification rule devised for each class of different characters. At last, the definite shear angle of the identified character (italic) is calculated to perform more accurate reverse shear transformation to rectify the italic style character into normal (non-italic) style character to facilitate the later OCR task.

IV. Proposed Work

The Main idea of detection includes slant line detection in the each word. For the slant line an angle calculation method is used here for the vertical line present in the word. Here system detect the vertical lines or slanted lines present in the word of a 25% height of the word and measure the slant angle and make a decision on whether this word is italic or not. It is possible to do rectification by a reversed shearing transform corresponding to the slant angle. Each character is composed by some basic units called strokes. The inputted character can then be classified into different character classes based on the structural information of its constituting strokes. We all admit that stroke extraction is a tedious and time-consuming process in image processing. In some applications, it is sufficient to merely extract the similar outlook of the character. The italic and non-italic style characters can then be distinguished based on the classification rule devised for each class of characters.

As we know, the boldface can be generated by extending the width of stroke in the normal style alphabet. Dilation and erosion are two basic operations of morphology which are useful in generating the boldface from normal alphabet and rectifying the boldface back to normal alphabet. The difficulty of boldface detection is that there is no suitable threshold which is selected depending on the kind of font type of the alphabet in classifying the width of stroke. In order to simplify the process of boldface detection, the following assumption is adopted: the font type in the same paragraph and the same word is the same. In this way, the task of boldface detection can be performed by checking the width of stroke in each word.

Algorithm for Italic Detection

1. *Initialize*
(Input Image= I; VS= Vertical Strokes; G =Gaussian filter; H=Line Height; S= slant angle; T (t1: t3) Angle Thresholds)
2. *Load & Read I ;*
3. *Pre-process I;*
4. *Canny Edge Detection (Input I; Output VS)*
   a. *Smoothing*
      Gaussian filter G = fspecial (Gaussian);
      Ig = imfilter (I, G);
   b. *Calculate gradients of I;*
   c. *Marked local maxima as edges*
   d. *Determined edges by thresholding*
   e. *Edge tracked by hysteresis*
5. *Calculate all VS > H/2*
6. *Check for all VS*
   If (t1<=S<=t2) then Italic Font
   else if (t2<S< t3) then Normal Font
   else (S< t1||S> t3) then Undecided
7. *End*

For bold detection of Devanagari script all the vertical and horizontal strokes are identified. In every stroke ratio of character height and stroke width are calculated and stores as bold ratio value.

Algorithm for Bold Detection

1. *Initialize*
   (Input Image= I; HP= Horizontal Profile; P=Pixel Value (0/1); PW=Pen Width; MZ=Middle; Zone; T=Threshold)
2. *Load & Read I ;*
3. *Pre-process I;*
4. *Horizontal Projection Profile (Input I; Output= HP[i])*

```
For (Row=0 to Image Height)
 {
  For (Column=0 to Image Width)
   {
     P= Get Pixel Value (Row, Column)
          If (p==0)
           {
                  HP[i] =HP[i] +1;
           }
    }
}
```

5. *Calculate Row where HP[i] =HP[i] - HP[i+1] is Maximum*
6. *Calculate PW of the Character using HP[i]*
7. *Calculate MZ of the Character using HP[i]*
8. *If MZ/PW > T Character is Bold*
   Else Character is Normal
9. *End.*

For underline detection of Devanagari script lower zone of the word is selected as a region of interest (ROI). If this region consist maximum horizontal projection value it detects as underline word.

Algorithm for Underline Detection

1. Initialize
(Input Image= I; HP= Horizontal Profile; LH=Line Height; W=Word Width; P=Pixel Value (0/1); LZ=Lower Zone; HS= Horizontal Stroke T=Threshold)
2. Load & Read I ;
3. Pre-process I;
4. Horizontal Projection Profile (Input I; Output= HP[i])

```
For (Row=0 to Image Height)
    {
  For (Column=0 to Image Width)
      {
      P= Get Pixel Value (Row, Column)
        If (p==0)
                {
                  HP[i] =HP[i] +1;
                }
            }
          }
```

5. Select LZ using LH/2
6. Search HS using Hough Line
7. If Slope of HS==0 and HS> (W x 0.7) Word is Underline
Else Word is Normal
8. End.

V. Result Analysis

To verify the effectiveness of our approach, a benchmark test is performed over the entire system module. All testing images are single text line images, which further segmented to the word wise for individual processing of the module. Input images are extracted from different sources and scanned over 330 DPI scale. The setting of control parameters described in previous sections is used in all experiments. Our benchmark test results are summarized in single phase which indicates the performance of three font type scripts say Italic, Underline and Boldface. Another aspect is type of font styles indicates the absolute percentage of the correct detection over three different font style which is "Kruti Dev 714", "Devanagari New" and "DevLys 240 Regular". The overall testing procss demonstrates that our approach has, total 95 % accuracy. This confirms the effectiveness of our approach for Italic, underline and Boldface detection and removal.

VI. Conclusions

There are many techniques involving in document analysis, such as binarization, segmentation, block classification, noise removing, and so on. However, there are trade-offs in each group of techniques. There is no general method that is appropriate to all kinds of documents and various environments. The results generated by document analysis usually drastically influence the performance of later tasks. Hence, the devising of good document analysis techniques is the goal to be pursued in the future. As we know, the purpose of font type detection is to distinguish the normal style character from its corresponding. Currently, the rules generalized for the features are suitable to be employed for italic, Bold and underline font detection of Devanagari characters.

References

[1] G Gunvantsinh, Rekha Teraiya and Mahesh Goyan, "Chain code and holistic features based ocr system for printed devanagari script using ANN and SVM", International Journal of Artificial Intelligence & Applications (IJAIA), Vol.3, No.1, 2012.

[2] Zhen-Long BAI and Qiang HUO "Underline Detection and Removal in a Document Image Using Multiple Strategies" The 17th International Conference on Pattern Recognition, Cambridge, UK, vol. 2, PP 578-581, 2004.

[3] Tushar Patnaik, Shalu Gupta, Deepak Arya — Comparison of Binarization Algorithm in Indian Language OCR, 2011.

[4] Font identification - In context of an Indic script: Chanda, S.; Dept. of Comput. Sci. & Media Technol., Gjovik Univ. Coll., Gjovik, Norway ; Pal, U. ; Franke, K., IEEE Pattern Recognition (ICPR), 21st International Conference on 11-15 Pp 1655 – 1658, 2012.

[5] S L. Zhang, Y. Lu, and C. L. Tan. Italic font recognition using stroke pattern analysis on wavelet decomposed word images. In ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4, pages 835–838, Washington, DC, USA, IEEE Computer Society, 2004.

[6] Ravi Kant Yadav and ireshwar Dass Mazumdar "Detection of Boldand Italic Character in Devanagari Script", International Journal of Computer Applications by IJCA Journal, Volume 39 2012.

[7] Yogendra Bagoriya, Nisha Sharma, "Font type identification of hindi printed document", IJRET: International Journal of Research in Engineering and Technology, Volume: 03 Issue: 03 PP. 513-516, 2014.

[8] Harjit Singh, "Detection of Bold and Italic Character in Gurmukhi Script", IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661 Volume 1, Issue 6, PP. 28-31, 2012.