Detection of Devanagiri Text in Digital Images using Connected Component Analysis

Vishwanadha Kaushik and C. V. Jawahar
Indian Institute of Information Technology
Gachibowli, Hyderabad – 500 019
INDIA

Abstract

Detection and recognition of Indian language text in a digital image, will allow the indexing of Internet images for web serach. This paper describes an algorithm for detecting Devanagiri text in diverse categories of digital images. Input scenes are decomposed into a set of binary images, where connected components are analysed for the possible presence of Devanagiri characters. Structural characteristics of the Devanagiri script are statistically analysed for estimating the upper and lower bounds of the features employed. Results are demonstrated on real-life images comprising of internet images, video scenes and document pages.

1 Introduction

Detection and recognition of text in digital images is of paramout importance in document image analysis [1, 2]. Document image processing algorithms often consider the identification of text characters as a trivial problem since those images are rich in text characters. In relatively sparse documents, page segmentation is carried out as preprocessing before the character recognition stage. With the availability of large storage devices and higher bandwidth networks, images have become more popular to convey the information which otherwisei,text could only attempt to do. Also the presence and interpretation of text in these images can provide critical clues for machine vision algorithms. These images typically provide visual information in addition to the possible texts in the form of captions,sub titles or image objects. Processing this text in such digital images is a relatively new domain [3], and practically uncharted in the context of Indian languages.

Text is the most common and widely used medium of data exchange. It is rich in information and precise in description. People often read text in paper-based documents, television screens, and oflate, on the Internet and web pages. A machine aided processing of these texts can be very helpful in understanding the text and therefore content of the document. Text to be processed by machine can be in two forms: either coded or

images. Coded text is formed from a set of symbols that are defined as a part of the font library. These symbols have an ASCII or ISCII value associated with them. Pixel text are represented by image pixels. Often, both forms of text appear together as is in most of the web pages.

In the present day context, the challenge is to detect the presence of text in various documents like images, videos or scanned documents. An appropriate character recognition module can be later employed to recognise the text. The technology will play a vital role in applications like Web Search, Color Image indexing, Image Database organization, Video Indexing etc [4, 5, 6]. The process of locating text in a given image is the first step in the problem of automatic or semi-automatic text reading. This stage often does not depend on character recognition module. Most of the conventional document image analysis algorithms presume the detection process as trivial and starts from here [7]. The problem of text detection gets complex with the variations in fonts, font sizes, styles, and textures.

In literature, most of the work in this area was carried out for detection of English text in images, as is this the de-facto language used on the internet. Now there are many Indian web sites which are built in regional languages (eg. `www.sify.com`, `www.rediff.com`). Unlike in English, there are no standard fonts that are popular for these languages. Therefore most of these sites resort to the usage of images as a medium to convey text. This paper focusses on detecting Devanagiri text in images available over the Internet, digital videos and document pages. The approach can be extended to many other indian languages.

In the next section, we discuss about the earlier work done in this area and also describe the application domains. We then proceed to elaborate upon the conceptual framework for solving this problem in section 3. Specific characters of the Devanagari script and the statistical facts for the assumptions are described in section 4. Finally experimental results are discussed in section 5.

2 Background and Application Domains

Earlier attempts to solve this problem made use of either spatial or frequency domain information for detection of text. The frequency domain solution adopted texture analysis techniques such as Gabor filtering etc. for detecting the spatial frequency characteristics of the characters [8]. However, this approach is sensitive to font nature and is highly language specific. On the other hand, the spatial domain solutions were based on connected component analysis [9] which is fast and a font independent method. This approach works only for binary level images but fails in the case of multi coloured images.

Jain and Yu [9] provide an extensive study of detecting English text in images and video scenes. They decomposed the colour images into a set of binary images and detection is carried out independently on these images.The results provided were promising for synthetic images. Li *et al.* [10] describe a real-time algorithm for detecting and tracking text in video scenes. They exploit the parallelism of neural networks to track the text in video frames. Almost all the reported algorithms focus the attention on detecting

fronto-parallel text. Clark and Mirmehdi [11] proposed a novel method to detect non-fronto-parallel text using some local measures. Most of the algorithms in document image analysis process and recognise only English chacaters. Employing the algorithms for indian languages require a careful modification of various modules. OCRs are reported for Devanagiri script in literature [12]. There are also reports on segmentation of scripts in multi-script documents [13, 14]. Detection of Devanagiri text in Digital Images is still a new research domain.

Advertisements and Monochrome Documents: Newspapers, a powerful medium of information exchange, having realised the increasing potential of the World Wide Web as the new medium, have put up their websites. These sites carry their online edition. A newspaper carry articles and classifed advertisements which need to be put on the online version. So there is a need for the extraction of text from the image which is scanned. Though the image is a binary image, there could be images in it in the form of a reversed representation (Logos and indian language text). Therefore there is a need for the identification of the text (so that it could go as the coded text in the web page). `www.Devanagarimilap.com, www.naiduniya.com, www.eenadu.net`. are examples of some sites which have an online edition of their newspapers in regional languages.

Web Search: The WWW is growing at a phenomenal rate. It is estimated that the number of websites are doubling in every three months. The number of Devanagari based sites has grown exponentially over the last 8 months. There is no dearth of information and the future would see more advancement. There would be more devanagiri based sites that would come up covering a spectrum of areas and aspects. In such a scenario, there is a need for a search engine that can locate and identify the text in the pages and use it for indexing and retrieving. Even in English the popular search engines are based on the coded text alone. In the context of the Devanagari language this is more relevant as the sites often use images to represent the text. Therefore a search engine for the Devanagari based languages would require the capability of locating Devanagari text in the images. The implementation would also be useful for multimedia search engines, with searching capabilities for Text documents,images and videos. The relevance of such search engines would become more obvious in the future when the information highway becomes a reality.

Video indexing and image database: The problems of Video indexing and image database, though different have a similar approach for the solution. Video infact is a set of images. These images normally make use of the RGB representation. The number of colours possible are 16M. A solution for this problem could prove useful in providing interactive video,indexing of video and building of image databases. These are the primary requirements to have a system that is capable of acting as a multimedia server.

3 A Framework for Detecting Devanagiri Text in Images

The spatial domain solution is a fast and excellently in the case of the binary images. In case we device a method that would reduce all the problems to the analysis of the binary images, we would be able to solve problems of varying nature.

The solution that we have tried to work out has focussed attention on three specific aspects:

1. Reduce a multicoloured image into an image of few colours.
2. Obtaining a set of foreground images from this reduced colour image.
3. Analyizing each of these images for the presence of the Devanagari text based on the statistical aspects.

A foreground image is a binary image with a specific colour pixel in the original image represented as black with the remaining as white.

The image that is fed as an input to the text locator could be a binary image, a pseudo colour image (as in the case of the synthetic images) or a true colour image as in the videos or real images. The analysis of the binary image is a relatively simple problem and that is our base case.

The analysis of each of the foreground image, makes use of the properties of the Devanagari text, which are unique for it. The unique attributes are determined after the statistical analysis of the Devanagari text. We present a description of these statistical aspects in the second part.

3.1 Image categories and pre-processing

The various categories of the images that could be fed as an input to the module fall into three categories. Binary level images,pseudo colour and true colour images. A detailed description of the preprocessing employed for each of the category is given below.

For a binary level image, the preprocessing required is minimal. There are two colours, a foreground and a background colour. The text is usually represented in the foreground. So we would need to look for the foreground components and perform the analysis. However it is possible to make use of the inverted text representation as well. That is a part of the image would be such that the text is in the background colour and that part would have the foreground as background.(commonly noticed in the newpapers). In such a context we need to do an inverted image analysis as well.

The next category of images is the pseudo colour images. The best examples of the pseudo colour images are GIFs. These images differ from the true colour images in the aspect that it makes use of only 256 different colours. The True colour images make use of

16M colours. So A gif colour set is a 256 elements subset in the 16M set of colours.However it is often observed that the gifs normally makes use of only a limited set of colours. They don't use the entire 256 colours possible. A sample gif is shown in fig 1. They normally make use of 4-8 colours. So we would need to analyze at the most 10 foreground images. A set of subimages obtained from the gif are as shown below.

Figure 1: A trypical synthetic GIF image

The problem of a true colour image is the most complex. There are 16M colours possible and it is difficult for us to identify and analyze each of the foreground image formed as a result of the decomposition. We would possibly obtain a very large set of foreground images, and none of them would contain any information. This brings out the need to reduce the number of colours in the true colour images. This makes the problem solvable. There are two mechanisms that are employed for this purpose.

- Bit dropping. In each of the 8 bits used to represent Red, Green and Blue, we retain only the 2 most significant . This is done keeping in mind, the fact there is no major loss of information in doing this. The contribution of the two MSB's alone effects the colour that we percieve(largely). In this process ,we reduce the number of possible colours from 16 M to 64.
- The second step is composed of a set of substeps which are as follows.

  1. Of the possible 64 colours, we look at the colour distance and draw up a 64 X 64 matrix. The colour distance is defined as:

     $$d(colour1, colour2) = \sqrt{(r-r')^2 + (g-g')^2 + (b-b')^2} \quad (1)$$

     where $colour1 = [r, g, b]^T$ and $colour2 = [r', g', b']^T$
  2. In the colour distance matrix, we choose a threshold after observing the psychovisual effect. In the matrix, all those colours which have their colour distance less than this threshold are merged up. The one with a higher number of pixels is choosen as the colour to represent the pixels. The threshold choosen by us was 0.45.(After observing the visual effects). At the end of this process, normally an image would consist of 4-6 colours. Thus making it feasible to form the subimages and analyze. A sample subimag is shown in Figure 2.

Figure 2: A sample two-level subimage

3.2 Analysis of the foreground images

The Foreground images after being formed are to be processed for the devanagari test. This phase makes use of the characteristics of the text. Each of the foreground image is processed and text is located. This phase makes use of the thresholds that are defined on the basis of the decisions made are statistical analysis. This process involves the identification of blocks based on the connected component. It is these blocks that we operate upon. These blocks may or may not contain text in it. The formation of these blocks is the segmentation process. The characteristics are described in the next section.

4 Characterisation of Devanagiri characters

The Blocks are rectangular sections. Some of them contain text while others do not. A few sample blocks are shown below.

The processing of these blocks is done based on these aspects.

All blocks of size less than 10x10 are discarded. The reason for doing this is that they are not perceivable to the human eye. So their presence is not of relevence. So this acts as the first threshold.

The characteristics that have been choosen to confirm the presence of the text in an image are based on the statistical parameters observed in the Devanagari text. A set of 7160 characters have been statistically analysed and the thresholds employed were chosen.

The Devanagari specific characteristics are as follows:

1. The presence of the horizontal bar connecting up the individual characters. All Devanagari based text have a horizontal bar at the top. This is referred to as sirorekha. This bar can be present only at two points. At the very beginning of the word or at the upper half. The statistical analysis reveals that this bar needs to be present in the 0-10So all the blocks, that contain the sirorekha in one of these regions of the block are the valid ones.The remaining blocks can be discarded.The results of the statistical analysis are presented in the form of graphs shown.

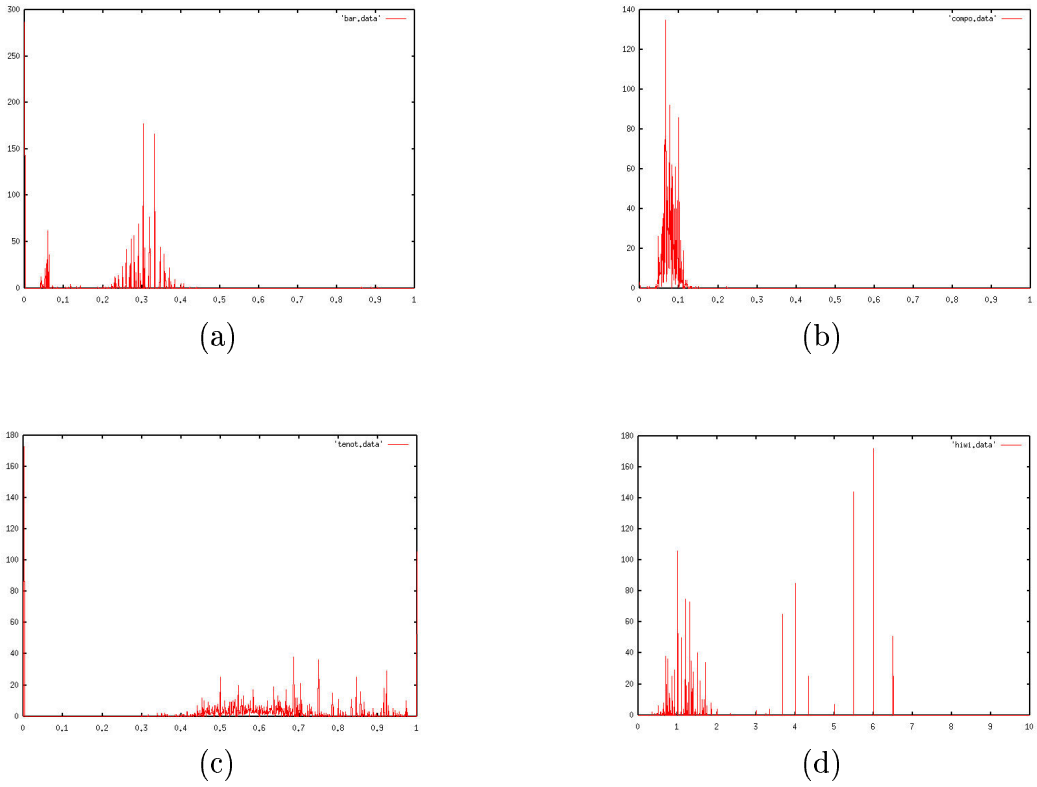

Figure 3: Graphs representing the characteristics of the Devanagari script

2. In The blocks that contain the bar,sirorekha, we count the number of components that are present. The count represents the number of characters present in the block. The number of components is always observed to be proportional to the width ofthe block. If the proportion is not maintained in the input block, it could be discarded. The ratios are taken with respect to the width of the block. It is observed from the results that the ratio should be 1 per hundred. or 3-15 per hundred.

3. The next Aspect that we look for in every image is the ratio of the text to the non text area is also constant. So we take a measure of thisin each of the character and decide upon each of the character as a valid character or not. If there are sufficient number of characters that satisify this condition then we accept it as a text. This ratio is ideally a 1:1. The textual area should be equal to the nontextual area.

4. The final characteristic that we look for is the height to the width ratio of each character. It doesn't have a fixed value but there are a set of values that it could take. The variation in this ratio can be attributed to the fact that the shapes of the characters vary a lot. For ex: In the case of a matra,the ratio comes to about 6:1.

Based on the output obtained from these four measures, we determine if the block under consideration is a valid block with text content or not.

At the end of this process, we could combine adjacent blocks, to from complete sentences.

5 Results and Discussions

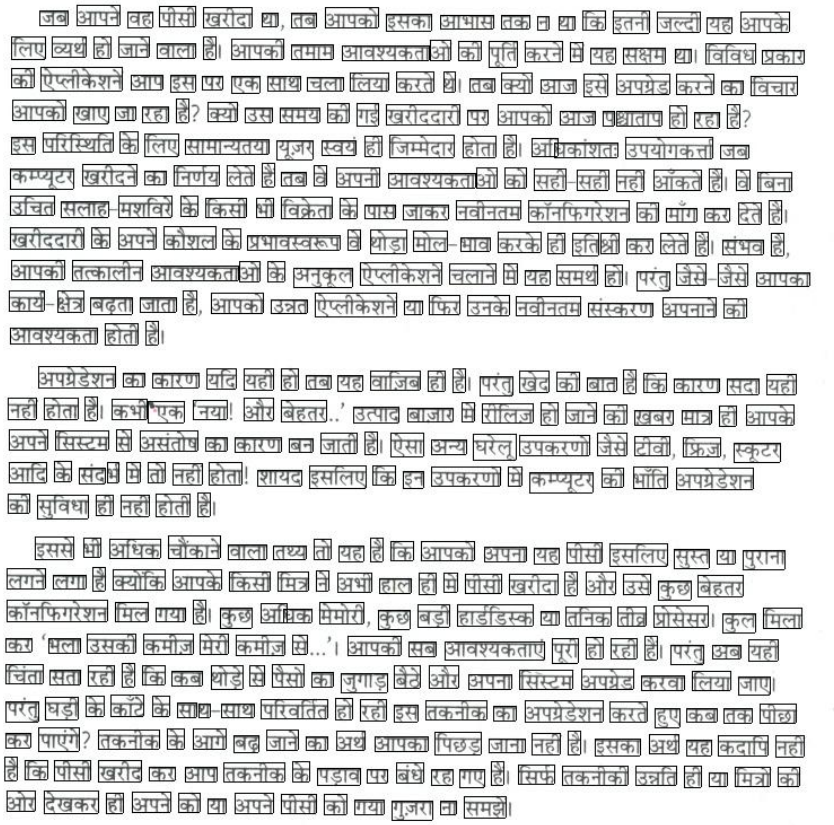

Figure 4: A trypical Devanagiri text document and the detected text blocks

In this section, we provide the results of the algorithm on various categories of images.

First, we consider binary images. This algorithm is the base case for all the other images. Images are basically segmented and blocks are obtained. The blocks are analyzed for the presence of Devanagiri chacters as described in the previous section. The statistical thresholds that we have computed are excellent for images of this nature. The alogirthm works perfectly and is able to distinguish between Devanagari text, and even punctuation marks and digits. Results on a document page is shown in Figure 4. Clearly the Roman script can be distinguished from Devanagiri. We could achieve 100% accuracy in this case. There are no false calls. The sizes of the image considered is 752x737. This image has been scanned at a resolution of 100dpi. The images has been obtained from a popular hindi computer magazine.

The second category of images that we deal with are the pseudo colour images. The algorithm performs excellently for this class of images. There is no need for color reduction and the subimages formed contain all formation. The sample gifs shown in Figure 1 has been processed and the output image is shown in Figure 5. Ther are no false calls in this case. We also tried with the gifs from various web sites like `http://www.dailymilap.com` and `http://hindi.sify.com`. The set of subimages obtained are finite without any

Figure 5: Output for the image shown in Fgure 1

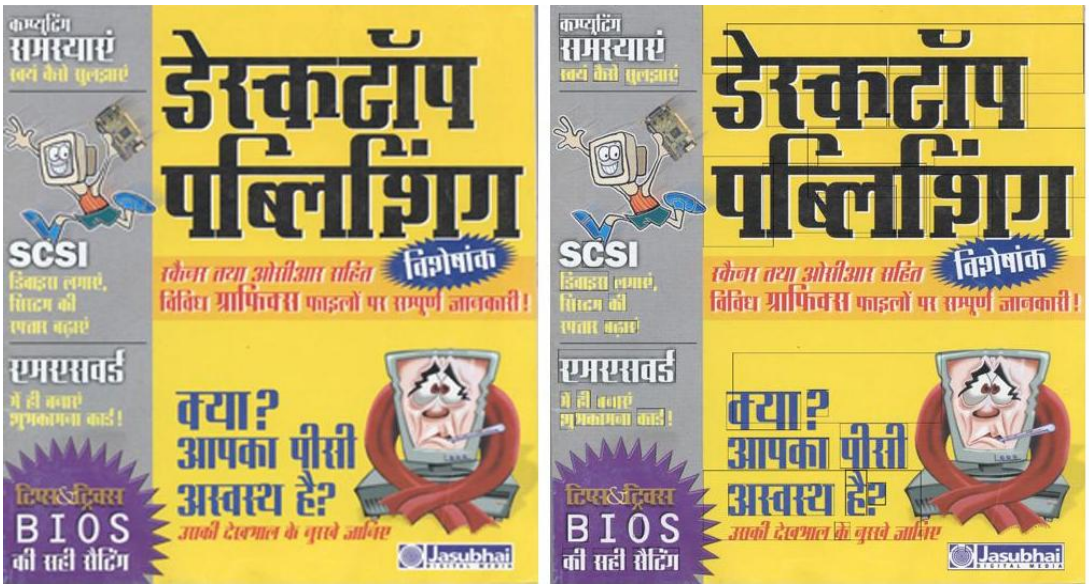

Figure 6: The true colour image and the detected text blocks

preprocessing being employed. The procedure for the text detection remains the same as in the first example. The image that has been shown below is from the web site, `www.dailymilap.com`. This image is of size 633x204 pixels.

We have also tested the algorithm on true colour images. These images require preprocessing for reducing the number of colours. The subimages obtained from preprocessed image are analysed for the presence of Devanagiri characters. The thresholds have been chosen to prevent missing of possible text regions while allowing for false alarms. The procedure due to its inherent lossy nature sometimes fails to locate the regions. There is a particular band that has not been detected in the image. The particular band has not been detected because none of the subimages formed have the region represented with its textual properties intact. Figure 6 depicts a true colour image obtained from a magazine cover and scanned at 75dpi. The detected text blocks are also shown. Except for a particular colour pair, our algorithm provided excellent results. We also tried with frames obtained from video sequence (Figure 7) Results are quite encouraging even in

Figure 7: A Video Scene and the detacted text possible blocks

noisy environment.

True colour image processing is lossy and the loss of information can possibly be compensated only by allowing false alarms. We are presently working on a lossless mode of analysis of structural characteristics of Devanagari using fuzzy notions.

The features employed remain invariant under the transformations, resulting in a robust algorithm for text detection. Rotation Invariance. Since most of the text in an image is aligned along the horizontal direction, we have implemented and suggested methods for tackling only the horizontal text. But the rotated text problem is solvable by addition of a procedure, that would rotate it and place the text horizontally to make it feasible for this module to handle.

The algorithm does not depend on the absolute coordinates of the pixels. It is only the shape/structure of a set of pixels, which provides the clue for presence or absence of Devanagiri text. Thus the location (translation) of the text does not influence the algorithm. This algorithm is not prone to varying font size. We employ the ratios as features instead of taking tha absolute values. And thus the robustness to scaling. As of now, we detect only horizontal texts. But the method can be made rotationally invariant by identifying the principal direction of the text block and compensating the rotation.

The performance of the algorithm can be improved by adding an OCR for recognising the detected text and providing a feed back based on the confidence with which the classifier recognises the character. This can result in an intelligent text detection algorithm which can adapt to new langauges/fonts and styles easily. A neural architexture may be appropriate for this.

Though the processing of binary image is simple, we loose information in the colour reduction process. We are presently working on using fuzzy notions to provide a comprehenisve representation of the colour variation while allowing the shape and structual features to be evaluated.

6 Conclusions

In this paper, we have presented a solution for locating Devanagari text in binary images, pseudo colour images, paper based documents and video frames. This proves to be useful for extracting text information from images. We have also discussed about the relevance of the approach in the contect of image databasing, video indexing, web searching etc. Structural characteristics of the Devanagiri characters are analysed to identify the possible range of features. Based on the experimentation, we have suggested 5 measures that help us in locating Devanagiri text. The method described here can be extended for other Indian languages with appropriate modification. We are presently working on this.

References

[1] G. Nagy, "Twenty years of document image analysis in PAMI," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, pp. 38–62, January 2000.

[2] Y. Zhong, K. Karu, and A. K. Jain, "Locating text in complex color images," *Pattern Recognition*, vol. 28, no. 10, pp. 1523–1535, 1995.

[3] J. Zhou and D. Lopresti, "Extracting text from www images," *In Proceedings of ICDAR*, pp. 248–252, 1997.

[4] Y. Dai, N. Zheng, X. Zhang, and G. Xuan, "Automatic recognition of province name on the license plate of moving vehicle," *n Proceedings of ICPR*, pp. 927–929, 1988.

[5] H.-K. Kim, "Efficient automatic text location method and content-based indexing and structuring of video database," *Journal of Visual Communication and Image Representation*, vol. 7, pp. 336–344, 1996.

[6] J. Shim, C. Dorai, and R. Bolle, "Automatic text extraction from video for content-based annotation and retrieval," *In Proceedings of ICPR*, pp. 618–620, 1998.

[7] I. Guyon, "Applications of neural networks to character recognition," pp. 353–382, 1991.

[8] A. Jain and S. Bhattacharjee, "Text segmentation using gabor filters for automatic document processing," *Machine Vision and Applications*, vol. 5, pp. 169–184, 1992.

[9] A. K. Jain and B. Yu, "Automatic text detection in images and video frames," *In Proc. of ICPR*, pp. 1497–1499, 1998.

[10] H. Li, D. Doermann, and O. Kia, "Automatic text detection and tracking in digital video," *IEEE Tran. Image Processing*, vol. 9, pp. 147–156, 2000.

[11] P. Clark and M. Mirmehdi, "Finding text region using localised measures,"

[12] B. B. Chaudhuri and U. Pal, "An OCR system to read two indian language scripts: Bangla and Devanagiri," *Proc. ICDAR, Bangalore*, pp. 1011–1015, 1997.

[13] U. Pal and B. B. Chaudhuri, "Automatic separation of different script lines from indian multi-script documents," *Proc. ICVGIP 98*, pp. 141–146, 1998.

[14] S. C. *et al.*, "Identification of scripts of indian languages by combining trainable classifiers," *Proc. ICVGIP 2000*, pp. 194–199, 2000.