Line Based Robust Script Identification for Indian Languages

Bhupendra Kumar, Aniket Bera, and Tushar Patnaik

Abstract—**In this paper a line based script identification using a hierarchical classification scheme is proposed to identify the Indian scripts includes Hindi, Gurumukhi and Bangla. We model the problem as topological, structural classification problem and examine the features inspired by human visual perception. Our basic algorithm uses different feature set at different level of classifier to optimize the tradeoff between accuracy and speed. The feature extraction is done on the subsets of image which in turn increases the performance of algorithm. The proposed system attains overall classification accuracy of 90% over the 2500+ text image data set.**

Index Terms—**Feature extraction, hierarchical classification, script identification.**

I. Introduction

In country like India where 18 official languages are used, multilingual data or documents are often found, also the amount of multimedia data captured and stored is increasing rapidly with the advances in computer technology. So, there is a great demand for software, which automatically extracts, analyses and stores information from physical documents for later retrieval. The techniques to solve these types of tasks are grouped under the general heading of document image analysis, which has been a fast growing area of research in recent years.

The ability to reliably identify the script type using the least amount of textual data is essential when dealing with document pages that contain different scripts. Automatic identification of scripts in document facilitates 1) Automatic archiving of multilingual documents, 2) Searching online archives of document images, 3) Selection of script specific OCR in a multilingual environment.

The Script Identification is a pre-requisite to the Consortia based OCR which presently needs manual feed of the Script. It will also be beneficial for the Web Based Version of the OCR where users can directly use a web image for OCR without the need of knowing the script.

In the proposed system we are focusing on the problem where an OCR system clearly needs human intervention to select an appropriate OCR package which is inefficient, undesirable and impractical.

The solution to script identification can be broadly classified into two categories local approach and global approach. Global approaches uses texture based features comprising sub-patterns and textons.

After a detailed study of various techniques[1]-[4], we have come up with a fast and robust line based script identification.

Manuscript received January 10, 2012; revised February 28, 2012.

Bhupendra Kumar is from IIIT Allahabad with the specialization in wireless communication and computing. India.

Tushar Patnaik is leading the consortium based project "Development of Robust Document Analysis and Recognition System for Printed Indian Scripts". India.

Aniket Bera is a graduate student of Jaypee Institute of Information Technology. India.

II. About the Scripts

The Bengali script (বাংলালিপি) is the writing system for the Bengali language. It is also used, with some modifications, for Assamese, Meitei, Bishnupriya Manipuri, Kokborok, Garo and Mundari languages. All these languages are spoken in the eastern region of South Asia. Historically, the script has also been used to write the Sanskrit language in the same region. From a classificatory point of view, the Bengali script is an abugida, i.e. its vowel graphemes are mainly realized not as independent letters like in a true alphabet, but as diacritics attached to its consonant graphemes.

Gurmukhi (ਗੁਰਮੁਖੀ) is the most common script used for writing the Punjabi language. An abugida derived from the Laṇḍā script and ultimately descended from Brahmi. Modern Gurmukhi has forty-one consonants (vianjan), nine vowel symbols (lāga mātrā), two symbols for nasal sounds (bindī and ṭippī), and one symbol which duplicates the sound of any consonant (addak). In addition, four conjuncts are used: three subjoined forms of the consonants Rara, Haha and Vava, and one half-form of Yayya. Use of the conjunct forms of Vava and Yayya is increasingly scarce in modern contexts

Devanagari (देवनागरी), is an abugida alphabet of India and Nepal. It is written from left to right, does not have distinct letter cases.

All the three scripts are recognizable by a horizontal line that runs along the top of full letters known as Shiro Rekha or the Matra line.

III. Preprocessing

A. Binarization/Noise Cleaning/Skew Correction

The document images were Binarized, Noise Cleaned and Skew Corrected before extracting the required components.

Adaptive binarization method extends Otsu's method to a novel adaptive binarization

Scheme. The first step of our method is to divide images into NxN blocks, and then Otsu's method is applied straightaway in each of the blocks. Then each and every pixel is applied with a nonlinear quadratic filter to fine tune

all the pixels according to the local information available.

In Otsu's method we exhaustively search for the threshold that minimizes the intra-class variance, defined as a weighted sum of variances of the two classes:

$$\sigma_{\omega}^2(t) = \omega_1(t)\sigma_1^2(t) + \omega_2(t)\sigma_2^2(t)$$

Weights ω_i are the probabilities of the two classes separated by a threshold t and σ_i^2 variances of these classes. Otsu shows that minimizing the intra-class variance is the same as maximizing inter-class variance

$$\sigma_b^2(t) = \sigma^2 - \sigma_{\omega}^2(t) = \omega_1(t)\omega_2(t)[\mu_1(t) - \mu_2(t)]^2$$

which is expressed in terms of class probabilities ω_i and class means μ_i which in turn can be updated iteratively.

The morphological opening and closing operators not only remove image noise but also connect discontinuities that are caused in the thresholding stage, in the character images that we have. The opening and closing operators are as follows:

$$A \circ B = (A \otimes B) \oplus B$$

and

$$A * B = (A \oplus B) \otimes B$$

Which $\oplus$ and $\otimes$ are respectively the morphological erosion and dilation operators and B is the related structure element. The algorithm implemented is a two-pass algorithm to find the individual connected components in the document. While finding the component, the size of the component, ie. number of pixel in them are also computed. An elongation parameter for each component is also computed i.e. whether the component is elongated or not is decided. The users are the maximum size of components that will be considered as noise and hence deleted. All components with 4 or less pixels are already considered as noise

Next we used the mlskew algorithm for correcting the skew in page which is generally caused due to incorrect scanning.

IV. Selection of Best Line in Page

For our analysis we need to use the line with maximum number of characters for the best results in script identification. For achieving instead of dividing the line into individual characters and counting then which would have been a slow module, we have devised a unique method.

We used horizontal profiling throughout the page and then used the data to segment the line. After line segmentation is complete we have used the following conditions to determine whether the line is the best line of the document

1) Using Vertical Profiling we found out the line width and compared it with the Page Width. To be accepted as a valid line, it should be at least 0.6 times the page width.
2) We neglect the first 2 lines of every page as they are generally heading lines or pages no's and have very less characters.
3) Comparing line width with the average line width of the document. This is necessary as in many languages the lower zone of one line the upper zone of the line below don't have any space in between them or sometimes they get joined due to incorrect printing. This fails when horizontal projection is used for line segmentation as there will not be any line where is the sum of the horizontal pixels is 0.
4) Using Vertical Profiling to determine the no of white space in the line. More the white space lesser are the actual characters. The white space in a line should not be more than 0.4 times the line width to be classified as a valid line.

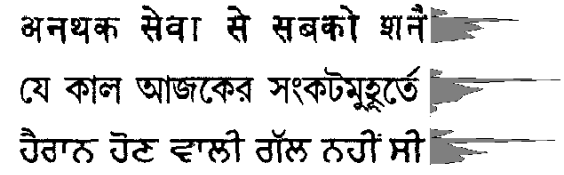


Fig. 1. Horizontal projection

If a line passes through all these conditions then the line is judged to be fit for script detection.

V. Local Skew/Alignment Correction by Image Reconstruction

Many old printed documents suffer from the problem of local skew/alignment [5, 6].

We have handled local skew/alignment in a unique way. After selecting the line we check for the matra line (Shiro Rekha or the Matra line is present in 3 Indian Scripts namely Hindi, Bangla and Gurumukhi).

If Matra Line is not detected it is passed through the Local Skew Correction Module

A line with local skew and deformed alignment

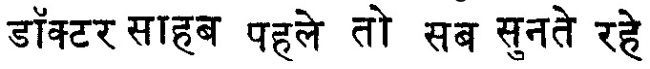


Fig. 2. Line with local skew

A. Steps we Have Taken to Correct These Types of Lines -

We performed vertical profiling again on the extracted line and segmented the line into words by analyzing the change in projections [7]. Word breaks will have 0 projection sum value. (All White Pixels)

Problem with the above method is that in old type writers and printers, there can be a gap between every character too, like in the above line.

To solve this we have analyzed many samples and found that in almost all cases the inter-character gap is less than 10% of the line width and hence after calculating the threshold we are able to separate words.

Inter-word Distance > Line Width*0.1 > Inter-character Distance

After word-level segmentation we have again applied the horizontal projections on each word to see if Matra line exists or not by the similar process we did for a line.

In this case we have set the threshold as 85% of line-filling for Matra detection, meaning atleast 85% of the word boundary (word width) should be occupied by the matra.

The logic behind keeping a higher value for words is that in words there will be very less inter-character gaps/white spaces.

After detecting Matra line for each word we have calculated the centroid of the every matra and its distance from the line border. To correct the alignment to reconstructed the image into a new line after normalizing the centroid locations of every word and hence the line will be nearly straight.

Before Local Skew Correction

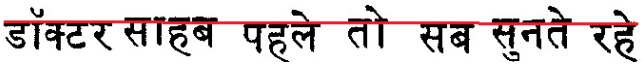


Fig. 3. Before local skew correction.

After Local Skew Correction

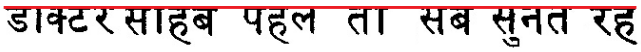


Fig. 4. After local skew correction

B. Results for Local Skew Correction

TABLE I: ACCURACY IN LOCAL SKEW CORRECTION

	Skewed Lines	Corrected	% Correction
Hindi	486	460	94.6
Gurumukhi	491	466	94.9
Bangla	503	492	97.8

C. Problems which are Still Existent after Pre-Processing

Lines which angular skew at word level will not give proper output

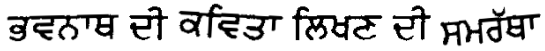


Fig. 5. Angular skew at word level

These lines will will not be skew corrected globally nor locally and hence matra line cannot be detected.

VI. REMOVAL TO DOTS IN MIDDLE ZONE

Using connected components we separated dots in the image and then deleted it. This is a pre-requisite for both Bangla and Gurumukhi Detection and enhances our match percentage.

For removing dots in code we found out all the components not connected so the main line and they were contained in the component boxed boundary of 0.5 of the Line Width

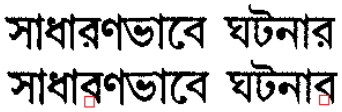
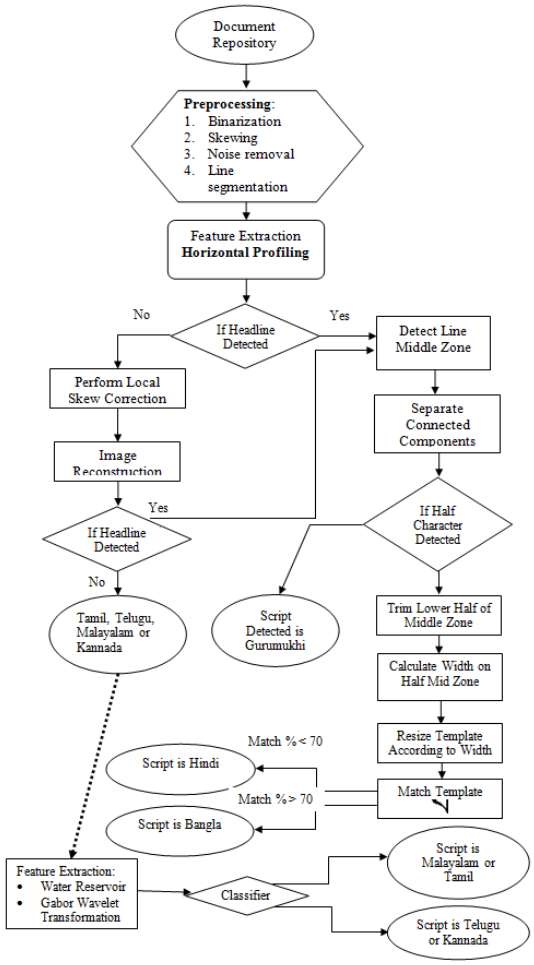


Fig. 6. Before and after dot removal

VII. TEMPLATE MATCHING

We have followed a hierarchical flow for identifying the script.



Flowchart. 1. Hierarchy for script identification

A. Matching for Gurumukhi

For Gurumukhi Matching we have used 8-Connected Components after removal of the Matra line to separate the characters.

A set of black pixels, **P**, is an ***8-connected component*** [8] if for every pair of pixels $\mathbf{p}_i$ and $\mathbf{p}_j$ in **P,** there exists a sequence of pixels p_i, ..., p_j such that:
a) all pixels in the sequence are in the set **P** i.e. are black, and

b) every 2 pixels that are ***adjacent in the sequence*** are ***8-neighbours***

After sepating the connected components we try to find the Gurumukhi Vowel Sign AA (*Kanna*)

Fig. 7. Gurumukhi *kanna*

This character is different from the Hindi AA Matra as the Hieght of the *Kanna* is Half of that if the Hindi AA Matra [9].

Fig. 8. Half-character detected in gurumukhi

B. Matching for Bangla

For matching for Bangla we have split the middle-zone and taken into consideration only the lower half of it.

Fig. 9. Bangla template

The Template [10] is opt-repeating part of many Bangla Characters and its occurrence is very frequent.

Fig. 10. Bangla characters which contain the Template we have used in the lower middle zone

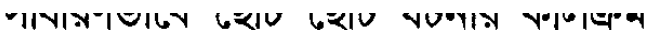


Fig. 11. Lower half of middle zone

If the match percentage

(No of Black Pixels Matched + No of Surrounding White Pixels Matched / Total Pixels in Block) is greater than 0.7 then it is classified as Bangla, else it is classified as Hindi

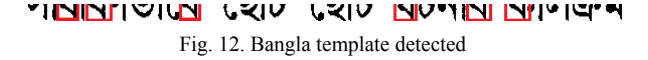


Fig. 12. Bangla template detected

VIII. Results

TABLE II: Script Detection Accuracy

Script	No. of Pages	Correct Classification	Accuracy %
Hindi	872	826	**94.7**
Gurumukhi	1094	916	**83.7**
Bangla	700	624	**89.1**

IX. Identified Problems

- Gurumukhi Half-Character not detected not due broken script.
- Bangla/Hindi Template Threshold is variable for font styles and fails if broken characters are present.
- High failure rate with fancy fonts.
- Pages with shorter lines like pages with poems have higher error rate due to the possible lack of the matching character.
- Threshold (Percentage match) for template matching is invariably dependent on the quality of scanned documents, for very poor quality scanned documents there is a possibility that none of the templates match.

References

[1] U. Pal and B. B. Choudhuri, "Script Line Separation From Indian Multi-Script Documents, 5th Int.Conference on Document Analysis and Recognition," *IEEE Comput. Soc. Press*, pp. 406-409, 1999.

[2] A. L. Spitz, "Multilingual document recognition Electronic publishing, Document Manipulations, and Typography," R. Furuta ed. Cambridge Uni. Press, pp. 193-206, 1990.

[3] B. B. Chaudhuri and U. Pal, "A complete printed Bangla OCR," *Pattern Recognition* vol.31, pp 531-549, 1998.

[4] B. B. Chaudhuri and U. Pal, "An OCR system to read two Indian language scripts: Bangla and Devnagari (Hindi),", in *Proc. of 4th ICDAR*, Uhn. 18-20 August, 1997. *A document skew detection method using the Hough transform. Pattern Analysis and Applications*. vol. 3. 243-253.

[5] B. V. Dhandra, V. S. Malemath, M. Hangarge, R. Hegadi, "Skew detection in Binary image documents based on Image Dilation and Region labeling Approach," in *Proceedings of ICPR*, vol. 2, no. 3, pp. 954-95, 2006

[6] G. Ciardiello, G. Scafuro, M. T. Degrandi, M. R. Spada, and M. P. Roccotelli, "An experimental system for office document handling and text recognition," In Proceedings of the ninth international conference on pattern recognition, Milano, pp. 739-743, 1988.

[7] K. Wu, E. Otoo, and A. Shoshani, "Optimizing connected component labeling algorithms," *Medical Imaging 2005: Image Processing*, vol. 5747. 1965-1976.

[8] L. D. Stefano , S. Mattoccia, and F. Tombari, "ZNCC-based template matching using bounded partial correlation," *Pattern Recognition Letters*, vol. 26 no.14, pp. 2129-2134, 15 October 2005

[9] S. Mattoccia, F. Tombari, and L. Di Stefano, "Fast full-search equivalent template matching by enhanced bounded correlation,"*IEEE Trans. Image Process*., vol. 17, no. 4, pp. 528-538, May 2008.

Mr. Bhupendra Kumar (Sr. Scientific Officer) joined CDAC in 2005, he received his M.Tech degree from IIIT Allahabad with the specialization in wireless communication and computing. His interest areas are Advanced Image processing, pattern recognition, computer network, wireless network, MANETs. Currently he is involved in project Development of "Document Analysis and Recognition System for Printed Indian Scripts"

Mr. Tushar Patnaik (Sr. Lecturer/Sr. Project Engineer) joined CDAC in 1998. He has eleven years of teaching experience. His interest areas are Computer Graphics, Multimedia and Database Management System and Pattern Recognition. At present he is leading the consortium based project "Development of Robust Document Analysis and Recognition System for Printed Indian Scripts"

Aniket Bera is a graduate student of Jaypee Institute of Information Technology. He is a developer and an avid programmer. His interests fields are in Computer Vision and Machine Learning. He has had work experience at C-DAC, Microsoft and many start-up organisations.