Journal of Advances in Engineering Science
Section C (3), January - June 2010, PP 47-56

SCRIPT IDENTIFICATION OF TEXT WORDS FROM A TRI-LINGUAL DOCUMENT

M. C. Padma[1]* and P. A. Vijaya[2]

1 - Dept. of Computer Science & Engineering, PES College of Engineering, Mandya-571401, Karnataka, India
2 - Dept. of Electronics & Communication Engg., Malnad College of Engineering, Hassan-573201, Karnataka, India

ABSTRACT

In a multi script environment, majority of the documents may contain text information in more than one script/ language forms. For automatic processing of such documents through Optical Character Recognition (OCR), it is necessary to identify different script regions of the document. With this context, this paper proposes to develop a model to identify and separate text words of Kannada, Hindi and English scripts from a printed trilingual document. The proposed method is trained to learn thoroughly the distinct features of each script. The binary tree classifier is used to classify the input text image. Experiments were conducted on manually created document images of size 600x600 pixels. The results are encouraging and prove the efficacy of the proposed model. The average success rate is found to be 99% for manually created data set and 98.5% for data set constructed from scanned document images.

Keywords: *Multi-lingual document processing, Script Identification, Feature Extraction, Binary Tree Classifier.*

1. INTRODUCTION

In recent years, the escalating use of physical documents has made to progress towards the creation of electronic documents to facilitate easy communication and storage of documents. However, the usage of physical documents is still prevalent in most of the communications. For instance, the fax machine remains a very important means of communication worldwide. Also, the fact that paper is a very comfortable and secured medium to deal with, ensures that the demand for physical documents continues for many more years to come. So, there is a great demand for software, which automatically extracts, analyses and stores information from physical documents for later retrieval. All these tasks fall under the general heading of document image analysis, which has been a fast growing area of research in recent years.

One important task of document image analysis is automatic reading of text information from the document image. The tool Optical Character Recognition (OCR) performs this, which is broadly defined as the process of reading the optically scanned text by the machine. Almost all existing works on OCR make an important implicit assumption that the script type of the document to be processed is known beforehand. In an automated multilingual environment, such document processing systems relying on OCR would clearly need human intervention to select the appropriate OCR package. This is certainly inefficient, undesirable and impractical. If a document has multilingual segments, then both analysis and recognition problems become more severely challenging, as it requires the identification of the languages before the analysis of the content could be made [10]. So, a pre-processor to the OCR system is necessary to identify the script type of the document, so that specific OCR tool can be selected. The ability to reliably identify the script type using the least amount of textual data is essential when dealing with document pages that contain text words of different scripts. An automatic script identification scheme is useful to (i) sort document images, (ii) to select specific Optical Character Recognition (OCR) systems and (iii) to search online archives of document image for those containing a particular script/language.

*padmapes@gmail.com

India is a multi-script multi-lingual country and hence most of the document including official ones, may contain text information printed in more than one script/language forms. For such multi script documents, it is necessary to pre-determine the language type of the document, before employing a particular OCR on them. With this context, in this paper, it is proposed to work on the prioritized requirements of a particular region- Karnataka, a state in India. According to the three-language policy adopted by most of the Indian states, the documents produced in Karnataka are composed of texts in Kannada- the regional language, Hindi – the National language and English. Such trilingual documents (documents having text in three languages) are found in majority of the private and Government sectors, railways, airlines, banks, post-offices of Karnataka state. For automatic processing of such tri-lingual documents through the respective OCRs, a pre-processor is necessary which could identify the language type of the texts words. So, in this paper, it is proposed to develop a model to identify and separate text words of Kannada, Hindi and English scripts In this paper, the terms script and language could be interchangeably used as the three languages - Kannada, Hindi and English belong to three different scripts.

This paper is organized as follows. The Section 2 briefs about the previous work carried out in this area. The database constructed for testing the proposed model is presented in Section 3. Section 4 briefs about the necessary preprocessing steps. In Section 5, complete description of the proposed model is explained in detail. The details of the experiments conducted and the states of results obtained are presented in section 6. Conclusions are given in section 7.

2. LITERATURE SURVEY

Automatic script identification is a challenging research problem in a multi script environment over the last few years. Major work on Indian script identification is by Pal, Choudhuri and their team [1, 3, 5]. Pal and Choudhuri [1] have proposed an automatic technique of separating the text lines from 12 Indian scripts (English, Devanagari, Bangla, Gujarati, Tamil, Kashmiri, Malayalam, Oriya, Punjabi, Telugu and Urdu) using ten triplets formed by grouping English and Devanagari with any one of the other scripts. This method works only when the triplet type of the document is known. Script identification technique explored by Pal [3] uses a binary tree classifier for 12 Indian scripts using a large set of features. The method suggested in [3] segments the input image up to character level for feature extraction and hence complexity increases. Lijun Zhou et. al. [9] has developed a method for Bangla and English script identification based on the analysis of connected component profiles. Santanu Choudhuri, et al. [4] has proposed a method for identification of Indian languages by combining Gabor filter based technique and direction distance histogram classifier considering Hindi, English, Malayalam, Bengali, Telugu and Urdu. Gopal Datt Joshi, et. al. [6] have presented a script identification technique for 10 Indian scripts using a set of features extracted from log-Gabor filters. Ramachandra Manthalkar et.al. [19] have proposed a method based on rotation-invariant texture features using multichannel Gabor filter for identifying seven Indian languages namely Bengali, Kannada, Malayalam, Oriya, Telugu and Marathi. Hiremath et al. [20] have proposed a novel approach for script identification of South Indian scripts using wavelet based co-occurrence histogram features. Though global approaches are faster, they are applicable and well suited only when the whole document or a paragraph or a text line are in one and only one script. But, in majority of the documents one text line itself may contain texts in different languages. For such documents, it is necessary to identify the script type at word level.

Sufficient work has also been carried out on non-Indian languages [2, 17, 18]. Tan [2] has developed a rotation invariant texture feature extraction method for automatic script identification for six languages: Chinese, Greek, English, Russian, Persian and Malayalam. Lijun Zhou et. Al. [9] has developed a method for Bangla and English script identification based on the analysis of connected component profiles. Peake and Tan [10] have proposed a method for automatic script and language identification from document images using multiple channel (Gabor) filters and gray level co-occurrence matrices for seven languages: Chinese, English, Greek, Korean,

Malayalam, Persian and Russian. Wood et al. [14] have proposed projection profile method to determine Roman, Russian, Arabic, Korean and Chinese characters. Hochberg et al. [15] have presented a method for automatically identifying script from a binary document image using cluster-based text symbol templates. Andrew Bhush [17] has presented a texture-based approach for automatic script identification. Spitz has [18] proposed method to discriminate between the Chinese based scripts and the Latin based scripts.

Some considerable amount of work has been carried out on specifically the three languages - Kannada, Hindi and English. Basavaraj Patil et. al. [7] have proposed a neural network based system for script identification of Kannada, Hindi and English languages. Vipin Gupta et. al. [13] have presented a novel approach to automatically identify Kannada, Hindi and English languages using a set of features-cavity analysis, end point analysis, corner point analysis, line based analysis and Kannada base character analysis. Word level script identification in bilingual documents through discriminating features has been developed by Dhandra et. al. [8]. Padma et. al. [11] have presented a method based on visual discriminating features for identification of Kannada, Hindi and English text lines. Though a great amount of work has been carried out on identification of the three languages Kannada, Hindi and English, very few works are reported in literature at word level. Also, the great demand for automatic processing of tri-lingual documents shows that much more work needs to be carried out on word level identification. So, this paper focuses on word wise identification of Kannada, Hindi and English scripts.

3. DATA COLLECTION

Standard database of documents of Indian languages is currently not available. In this paper, it is assumed that the input data set contains text words of Kannada, Hindi and English languages. For the experimentation of the proposed model, three sets of database were constructed, out of which one database was used for learning and the other two databases were constructed to test the system. The proposed model was made to learn the features of the three languages using manually constructed data set of 800 text words from each of the three languages. The text words of Kannada and English languages were created using the Microsoft word software. These text words were imported to the Micro Soft Paint program and saved as black and white bitmap (BMP) images. The font type of Times New Roman, Arial, Bookman Old Style and Tahoma were used for English language. The font type of Vijaya, Kasturi and Sirigannada were used for Kannada language. The font size of 14, 20 and 26 were used for both Kannada and English text words. However, the performance is independent of font size. The text words of Hindi language were constructed by clipping only text portion of the document downloaded from the Internet.

To test the proposed model, two different data sets were constructed. One dataset was constructed manually similar to the dataset constructed for learning and the other data set was constructed from the scanned document images. The printed documents like newspapers and magazines were scanned through an optical scanner to obtain the document image. The scanner used in this research work for obtaining the digitized images is HP Scan Jet 5200c series. The scanning is performed in normal 100% view size at 300 dpi resolution. The test document image of size 600x600 pixels were considered such that each text line would contain text words in mixture of the three languages. Manually constructed dataset is considered as good quality dataset and the data set constructed from the scanned document images are considered poor quality data set. The test data set was constructed such that 600 text words were incorporated from each of the three languages - Kannada, Hindi and English.

4. PREPROCESSING

Any script identification method used for identifying the script type of a document, requires conditioned image input of the document, which implies that the document should be noise free, skew free and so on. In this paper, the preprocessing techniques such as noise removal and skew correction are not necessary for the manually constructed data sets. However, for the datasets that were constructed from the scanned document images, preprocessing steps such as removal of non-text regions, skew-correction, noise removal and

binarization is necessary. In the proposed model, text portion of the document image was separated from the non-text region manually. Skew detection and correction was performed using the existing technique proposed by Shivakumar [16]. Binarization can be described as the process of converting a gray-scale image into one, which contains only two distinct tones, that is black and white. In this work, a global thresholding approach is used to binarize the scanned gray scale images where black pixels having the value 0's correspond to object and white pixels having value 1's correspond to background.

The document image is segmented into several text lines using the valleys of the horizontal projection profiles computed by a row-wise sum of black pixels. The position between two consecutive horizontal projections where the histogram height is least denotes the boundary of a text line. Using these boundary lines, document image is segmented into several text lines. Each text line is further segmented into several text words using the valleys of the vertical projection profile computed by a column-wise sum of black pixels. From the experimentation, it is observed that the distance between two words is greater than two times the distance between two characters in a word. So, the threshold value for inter word gap is decided as two times the inter character gap. Using this inter word gap, each text line is segmented into several text words. Then, a bounding box is fixed for the segmented text word by finding the leftmost, rightmost, topmost and bottommost black pixels. Thus, the image of the bounded text word is prepared ready for further processing such as feature extraction.

5. OVERVIEW OF THE PROPOSED MODEL

Script identification may seem to be an elementary and simple issue for human in the real world, but it is difficult for a machine. The reason is that, the different languages are made up of different shaped patterns to produce different character sets. Individual text patterns of one script are collected together to form meaningful text information in the form of either a word, a text line or a paragraph. The collection of the text patterns of the same script exhibits a distinct visual appearance due to the presence of the segments like – horizontal lines, vertical lines, upward curves, downward curves, descendants, hole-like structures and so on. It was inspired to use distinct characteristic behaviour of a particular script as supporting features to identify the type of script.

A. Properties of the three scripts

It could be observed that most of the Kannada characters have either horizontal lines or hole-like structures present at top portion of the characters. Also, it could be observed that majority of Kannada characters have upward curves present at their bottom portion. In addition, Kannada text lines have combination of basic and compound characters. The compound characters are obtained by combining basic characters - vowels and consonants. So, some compound characters have descendants called 'vothakshara' found below the basic characters. Thus, the presence of the structures like – horizontal lines, hole-like structures, upward curves and descendants could be used as the supporting features to identify Kannada scripts.

It could be noted that many characters of Hindi script have a horizontal line at the upper part called sirorekha [1], which is generally called a headline. It could be seen that, when two or more basic or compound characters are combined to form a word, the character headline segments mostly join one another and generates one long headline at the top portion of each text word. These long horizontal lines are present at the top portion of the characters. Another strong feature that could be noticed in Hindi script is the presence of vertical lines.

It could be observed that the upward-curve and downward-curve shaped structures are present at the bottom and top portion of majority of English characters. Also, it could be observed that majority of the English characters have vertical line like structures. The presence of the distinct characteristic structures of each script are used as supporting features in the proposed model.

The discriminating features of the three scripts - Kannada, Hindi and English are well projected by partitioning the text word. Four reference lines obtained from the top-profile and the bottom-profile of each text word are used to partition the text word. The

top-profile (bottom-profile) of a text word represents a set of black pixels obtained by scanning each column of the text word from top (bottom) until it reaches a first black pixel. Thus, a component of width N gets N such pixels. The row at which the first black pixel lies in the top-profile (bottom-profile) is called top-line (bottom-line). The row having the maximum number of black pixels in the top-profile (bottom-profile) is called the attribute top-max-row (bottom-max-row). Using these four reference lines each text word is partitioned into three zones – top-zone, middle-zone and bottom-zone. A typical partitioned portion of Kannada text word is shown in Figure 1. The attribute 'x-height' represents the difference between top-max-row and bottom-max-row.

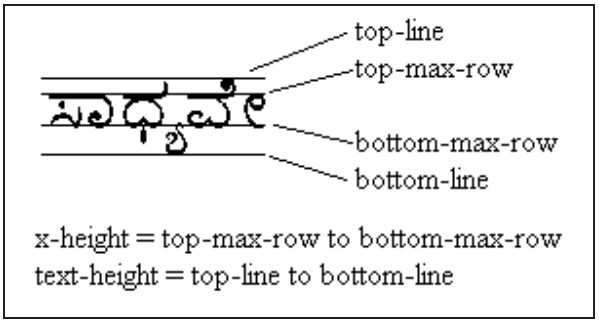

Figure 1 : Partitioned Kannada Text Word.

The distinct features useful for identifying the three scripts - Kannada, Hindi and English are shown in Table 1. The entry 'Y' in the Table 1 means that the feature in the corresponding row is used for identifying the script in the corresponding column. Thus, four features for Kannada, two features for Hindi and three features for English are used.

Table 1 : Features of Kannada, Hindi and English languages.

	Features	Kannada	Hindi	English
F1	Bottom-components	**Y**	—	—
F2	Bottom-max-row	**—**	**Y**	—
F3	Top-horizontal-line	**Y**	**Y**	—
F4	Vertical lines	**—**	—	**Y**
F5	Holes	**Y**	—	—
F6	Top-downward-curves	—	—	**Y**
F7	Bottom-upward-curves	**Y**	—	**Y**

B. Feature Extraction

The method of extracting the distinct features, which are used in the proposed model, is explained below:

Feature 1: Bottom-component

The presence of votthaksharas or descendants found at the bottom portion of Kannada script could be used as a feature called bottom-component. The feature named 'bottom-component' is extracted from the bottom-portion of the input text line. Bottom-portion is computed as follows:

Bottom-portion= f(x,y) where x=bottom-max-row to m and y=1to n ; where f(x,y) represent the image of the input text line.

Through experimentation, it is estimated that the number of pixels of a descendant is greater than 8 pixels and hence the threshold value for a connected component is fixed as 8 pixels. Any connected component whose number of pixels is greater than 8 pixels is considered as the feature bottom-component. Such bottom-components extracted from Kannada script are shown in Figure 2.

Feature 2: Bottom-max-row

It is observed through experimentation that for Kannada and English script, the attribute bottom-max-row lies closer to the bottom-line than the top-line, whereas for Hindi script, the bottom-max-row occurs very close to the top-max-row than the bottom-max-row. So, the attribute bottom-max-row could be used as a strong feature to separate Hindi script from the other two scripts.

Feature 3: Top-horizontal-line

It could be noted that the horizontal line like structures are present at the top-max-row of Kannada and Hindi scripts. The connected components present at the top-max-row of the text word are analyzed. If the number of pixels of these connected components is greater than the ¾ of the middle-portion of the text line, then such components are used as the feature top-horizontal-line. The probability of presence of this feature is calculated from the complete Kannada character set. Also, the distribution of this feature is analyzed using 500 text

words from all the three languages Kannada, Hindi and English. From the experimental analysis, it is observed that the presence of top-horizontal-line is more in Kannada and Hindi script and it is almost absent in the case of English script. So, using the feature named top-horizontal-line, English script could be separated from Kannada and Hindi script. If the length of the horizontal line (length of the horizontal line is measured with the number of pixels of that component) is greater than two times the x-height (height of the middle-portion of the text word), then Hindi text word can separated from Kannada word. So, using the length of the feature top-horizontal-line, Hindi can be well separated from Kannada script.

Feature 4: Vertical lines

It is noticed that the Hindi and English scripts have vertical line segments. To extract these vertical lines, the middle-zone of the text line is extracted as below:

Middle-zone = g(x,y) where x = top-max-row to bottom-max-row and y = 1 to n

where g(x,y) is the text line image of size (m,n). By convolving a vertical line filter over the image of the middle-zone, vertical lines are extracted. Typical vertical lines extracted from English script are shown in Figure 4.

Feature 5: Top-holes

Hole is a connected component having a set of white pixels enclosed by a set of black pixels. By thoroughly observing the words of Kannada scripts, it is noticed that hole like structures are found at the top portion. Presence of holes at the top-thick-zone gives clue to identify the text word as Kannada script as this feature is not present in the other two anticipated languages.

Features 6 & 7: Top-downward-curves and Bottom-upward-curves

The attribute upward-curve (downward-curve) is used for the connected component which have two runs of black pixels that appear on a single scan line of the raster image and if there is a run on the line below (above) which spans the distance between these two runs.

By thoroughly observing the structural shape of the scripts like Kannada and English, it is observed that the upward and downward shaped components are present at the region of top-max-row and bottom-max-row. This inspired us to extract the two attributes top-pipe and bottom-pipe as follows:

Top-pipe = g(x,y) where x=top-max-row – t to top-max-row+t and y=1 to nt and

Bottom-pipe = g(x,y) where x=bottom-max-row – t to bottom-max-row + t and y=1 to nb

where g(x,y) and nt represents the input image and number of columns of the input image. The variable 't' is used as a threshold value and t=round(x-height/3).

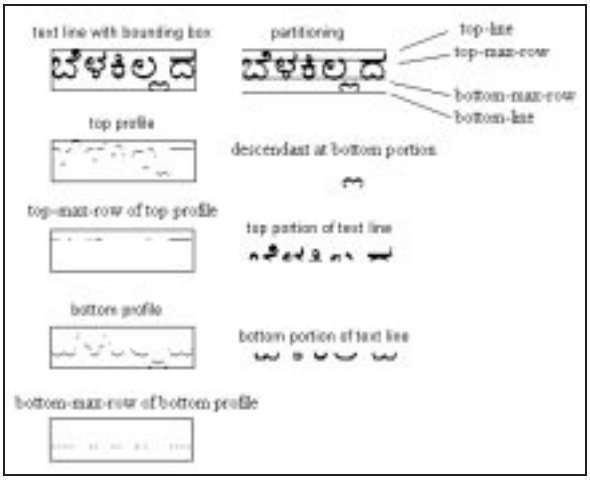

Figure 2 : Output Image of Kannada Text Word.

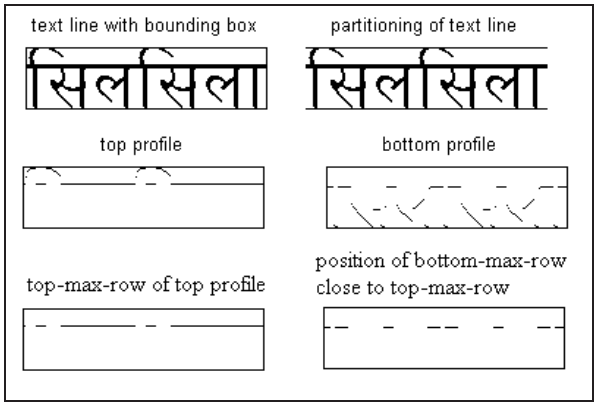

Figure 3 : Output Image of Hindi Text Word.

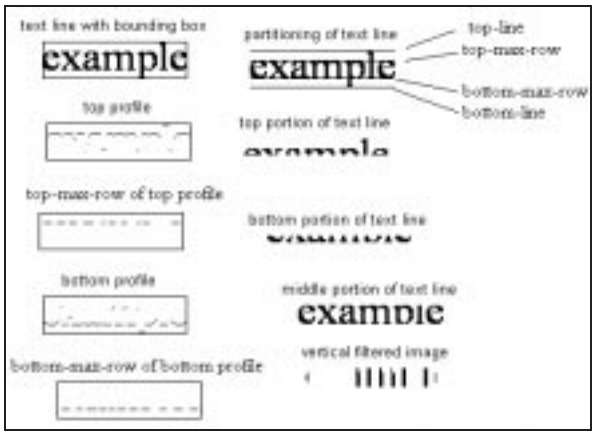

Figure 4 : Output Image of English Text Word.

C. Script Identification Technique

Based on the above principal features, a binary tree classifier is used to identify the script type of the text word. A portion of the binary tree classifier is shown in Figure 5. In the Figure 5, the left and right of the decision box is true and false respectively. The input text word of unknown script type is examined to check for the presence of features in the order as shown in Figure 4. The feature that is present in one and only one language amongst the three scripts is considered at the first level of the binary tree. In the proposed method, the feature - F1 is chosen first, out of the seven features because, the feature F1 is strong enough to discriminate among the three scripts. If the input test word is a Kannada text word with a descendant, then the test word is classified as Kannada script by extracting only one feature – F1. Only in the absence of the feature - F1, the next feature – F2, is chosen among the remaining six features as shown in the Figure 2. In this way, the features are extracted at each level and classified to the respective script type if the corresponding feature is present. Thus the features are extracted in the sequence that minimizes the testing time. At the best case, the input text word is classified at the first level itself by extracting only one feature. At the worst case, all the features are extracted. In the absence of all the features, the text word is considered rejected.

6. RESULTS AND DISCUSSION

The proposed algorithm has been tested on a test data set of 500 document images containing about 400 text words from each script. The test data set is constructed such that the English text words contain characters that possess ascenders (for example b, d, f, h, k, l, t) and descenders (for example g, j, p, q, y). The English text word without any ascenders and descenders (for example cow, man, scanner) are also considered in the test data set. The performance of classification is encouraging for the words having characters with and without ascenders and descenders. Similarly, in the test data set, Kannada and Hindi text words that contain characters with different features are considered. The success rate is sustained even for the text words with only two characters. This is because the features are chosen such that every text word has at least one or the other feature. The proposed method is independent of font type and size. Since the features are considered to be at specific region of the partitioned text word, the variation in the font size does not affect the performance of the algorithm. The percentage of recognition of all the three scripts is given in Table 6. From the experimentations on the test data set, the overall accuracy of the system has turned out to be 99%. From the Table 6, it could be observed that the 100% accuracy is obtained for Hindi text word. This is because of the distinct feature of Hindi script. The performance of the proposed algorithm falls down for English text words printed in italics. This is one limitation. However, for the Kannada text words printed in italics, the performance is sustained. The proposed algorithm has been implemented using Matlab R2007b. The average time taken to identify the script type of the text word is 0.08436 seconds on a Pentium-IV with 1024 MB RAM based machine running at 1.60 GHz.

The proposed algorithm is also tested on another test data set constructed from the scanned document images. The overall accuracy of the system reduces to 98.5% due to noise and skew-error in the scanned document images. However, if the scanned document images undergo suitable preprocessing techniques, the performance can be improved.

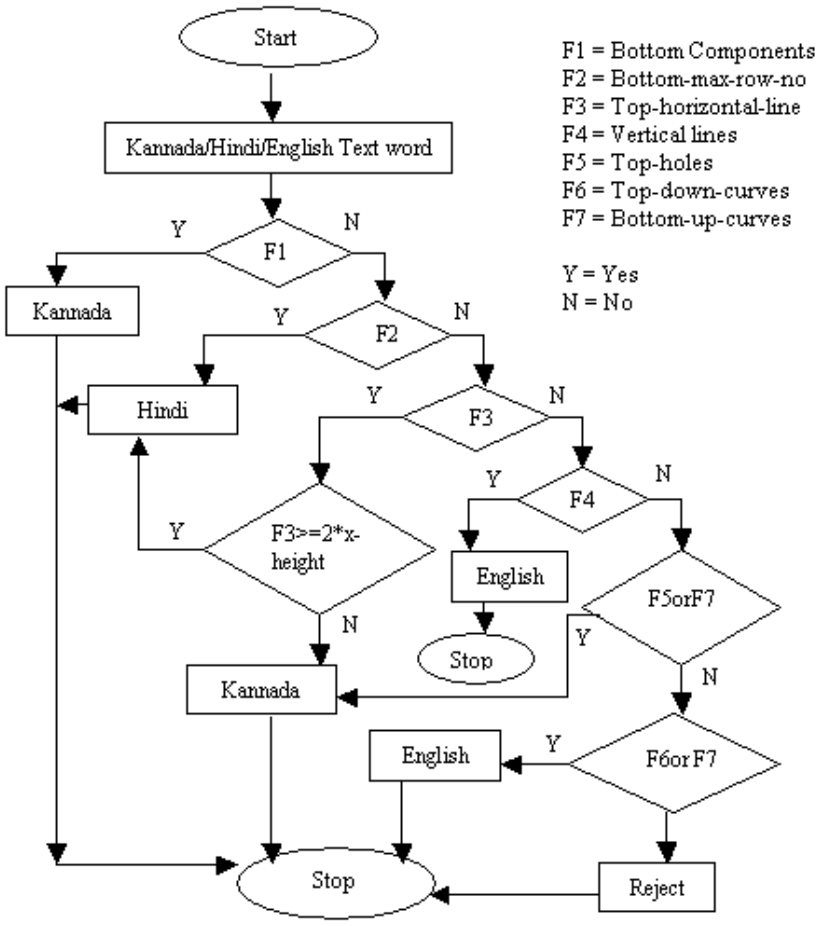

Figure 5 : Flow diagram of the binary tree classifier.

Table 2 : Percentage of Classification of the Three Languages

	Dataset 1 (Good Quality)			Dataset 2 (Poor Quality)		
	Classified	***Misclassified***	***Rejected***	***Classified***	***Misclassified***	***Rejected***
Kannada	98.6%	0.6%	1.8%	97.8%	0.8%	1.4%
Hindi	100%	0%	0%	100%	0%	0%
English	98.4%	0.4%	1.2%	97.6%	0.8%	1.6%

The proposed method is compared with the four methods given in [3, 7, 11, 13] as shown in Table 4. Method discussed in [13] has reported a marginal higher rate of success (99.2%). However, in [13], the features are extracted from individual characters, which lead to more time complexity in recognizing the sample text word. Whereas in the proposed method, the sample text word is not segmented into characters and the features are extracted from the text word it self. So, the time complexity in identifying script type of the text word is reduced. In the proposed method, the overall time taken to identify the sample text word is 0.08436 seconds on a Pentium-IV with 1024 MB RAM based machine running at 1.60 GHz.

Table 3 : Comparison of the proposed method with the previous methods

Previous work	Number of scripts	Database size	Proposed Technique	Performance	Remarks
[7]	3	450 words	Neural network based system	98%	The method is tested only on manually created less data sets.
[3]	12	750 text lines	Local Features: Water reservoir principle, contour tracing, profile, etc.	98%	More complex since the features are extracted from individual characters.
[13]	3	5000 words	Local Features: cavities, corner points, end point connectivity.	99.2%	More computation time since the features are extrcted from individual characters.
[12]	3	1450 words	Local features: horizontal lines, vertical lines, variable sized characters and characters with more than one component	95.66%	Performance reduces when number of characters in a word is less than 3
Proposed method	3	1500 text words	Local features: Bottom descendants, horizontal lines, vertical lines, top holes, upward curves and downward curves. Classifier: binary tree classifier.	99% (data set 1) 98.5% (data set2)	Performance is better compared to previous methods and also because of the use of binary tree classifier, computation time is reduced.

7. CONCLUSION

In this paper, a new method to identify and separate text words of the Kannada, Hindi and English scripts is presented. Experimental results show performance of the proposed model. The performance of the proposed algorithm is encouraging when the proposed algorithm is tested using manually created data set. However, the performance slightly comes down when the algorithm is tested on scanned document images due to noise and skew-error.

REFERENCES

1. U.Pal, B.B.Choudhuri, “Script Line Separation From Indian Multi-Script Documents”, 5th Int. Conference on Document Analysis and Recognition(IEEE Comput. Soc. Press), 406-409, (1999).
2. T.N.Tan, “Rotation Invariant Texture Features and their use in Automatic Script Identification”, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 7, pp. 751-756, (1998).
3 U. Pal, S. Sinha and B. B. Chaudhuri, “Multi-Script Line identification from Indian Documents”, In Proceedings of the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003) 0-7695-1960-1/03 © 2003 IEEE, vol.2, pp.880-884, (2003).
4. Santanu Choudhury, Gaurav Harit, Shekar Madnani, R.B. Shet, “Identification of Scripts of Indian Languages by Combining Trainable Classifiers”, ICVGIP, Dec.20-22, Bangalore, India, (2000).
5. S. Chaudhury, R. Sheth, “Trainable script identification strategies for Indian languages”, In Proc. 5th Int. Conf. on Document Analysis and Recognition (IEEE Comput. Soc. Press), pp. 657–660, 1999.
6. Gopal Datt Joshi, Saurabh Garg and Jayanthi Sivaswamy, “Script Identification from Indian

Documents", LNCS 3872, pp. 255-267, DAS (2006).

7. S.Basavaraj Patil and N V Subbareddy, "Neural network based system for script identification in Indian documents", Sadhana Vol. 27, Part 1, pp. 83–97. © Printed in India, (2002).
8. B.V. Dhandra, Mallikarjun Hangarge, Ravindra Hegadi and V.S. Malemath, "Word Level Script Identification in Bilingual Documents through Discriminating Features", IEEE - ICSCN 2007, MIT Campus, Anna University, Chennai, India. pp.630-635. (2007).
9. Lijun Zhou, Yue Lu and Chew Lim Tan, "Bangla/ English Script Identification Based on Analysis of Connected Component Profiles", in proc. 7th DAS, pp. 243-254, (2006).
10. G. S. Peake and T. N. Tan, "Script and Language Identification from Document Images", Proc. Workshop Document Image Analysis, vol. 1, pp. 10-17, 1997.
11. M. C. Padma and P.Nagabhushan, "Identification and separation of text words of Karnataka, Hindi and English languages through discriminating features", in proc. of Second National Conference on Document Analysis and Recognition, Karnataka, India, pp. 252-260, (2003).
12. Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, "Digital Image Processing using MATLAB", Pearson Education, (2004).
13. Vipin Gupta, G.N. Rathna, K.R. Ramakrishnan, "A Novel Approach to Automatic Identification of Kannada, English and Hindi Words from a Trilingual Document", Int. conf. on Signal and Image Processing, Hubli, pp. 561-566, (2006).
14. S. L. Wood, X. Yao, K. Krishnamurthy and L. Dang, "Language identification for printed text independent of segmentation", Proc. Int. Conf. on Image Processing, pp. 428–431, 0-8186-7310-9/95, 1995 IEEE.
15. J. Hochberg, L. Kerns, P. Kelly and T. Thomas, "Automatic script identification from images using cluster based templates", IEEE Trans. Pattern Anal. Machine Intell. Vol. 19, No. 2, pp. 176–181, 1997.
16. Shivakumar, Nagabhushan, Hemanthkumar, Manjunath, 2006, "Skew Estimation by Improved Boundary Growing for Text Documents in South Indian Languages", VIVEK- International Journal of Artificial Intelligence, Vol. 16, No. 2, pp 15-21.
17. Andrew Busch, Wageeh W. Boles and Sridha Sridharan, "Texture for Script Identification", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 11, pp. 1720-1732, Nov. 2005.
18. A. L. Spitz, "Determination of script and language content of document images", IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 19, No.3, pp. 235–245, 1997.
19. Ramachandra Manthalkar and P.K. Biswas, "An Automatic Script Identification Scheme for Indian Languages", IEEE Tran. on Pattern Analysis And Machine Intelligence, vol.19, no.2, pp.160-164, Feb.1997.
20. Hiremath P S and S Shivashankar, "Wavelet Based Co-occurrence Histogram Features for Texture Classification with an Application to Script Identification in a Document Image", Pattern Recognition Letters 29, 2008, pp 1182-1189.

■