IDENTIFICATION OF TELUGU, DEVANAGARI AND ENGLISH SCRIPTS USING DISCRIMINATING FEATURES

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

[1]Department of Computer Science Engineering, PES College of Engineering, Mandya, India
padmapes@gmail.com
[2]Department of Electronics and Communication Engineering, Malnad College of Engineering, Hassan, India
pavmkv@gmail.com

ABSTRACT

In a multi-script multi-lingual environment, a document may contain text lines in more than one script/language forms. It is necessary to identify different script regions of the document in order to feed the document to the OCRs of individual language. With this context, this paper proposes to develop a model to identify and separate text lines of Telugu, Devanagari and English scripts from a printed tri-lingual document. The proposed method uses the distinct features extracted from the top and bottom profiles of the printed text lines. Experimentation conducted involved 1500 text lines for learning and 900 text lines for testing. The performance has turned out to be 99.67%.

KEYWORDS

Multi-script multi-lingual document, Script Identification, Feature extraction.

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. 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.

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, which 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.

In a multi-script multi-lingual country like India (India has 18 regional languages derived from 12 different scripts [1]), a document page like bus reservation forms, question papers, language translation books and money-order forms may contain text lines in more than one script/language forms. One script could be used to write more than one languages. For example, languages such as Hindi, Marathi, Rajastani, Sanskrit and Nepali are written using the Devanagari script; Assamese and Bangla languages are written using the Bangla script. In order to reach a larger cross section of people, it is necessary that a document should be composed of text contents in different languages. However, for a document having text information in different languages, 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, the problem of recognizing the language type of the text content is addressed. However, it is perhaps impossible to design a single recognizer, which can identify a large number of scripts/languages. As a via media, this paper proposes to work on the prioritized requirements of a particular region- Andra Pradesh, a state in India. According to the three-language policy adopted by most of the Indian states, the documents produced in any Indian state are composed of text information in their regional language, the National language - Hindi and the general importance language - English. Accordingly, the documents of Andra Pradesh are generally printed in Telugu, Hindi and English languages. Consequently, majority of the documents produced in many of the private and Government sectors, railways, banks, post-offices of Andra Pradesh are of tri-lingual (a document having text in three languages) type. So, when it comes to automation, assuming that there are three OCRs for Telugu, Hindi (Devanagari) and English languages, a pre-processor is necessary by which the language type of the different texts lines are identified. In this paper, a script identification technique to identify the text lines of Telugu, Hindi and English languages from a tri-lingual document is presented.

The rest of the paper is organized as follows. Section 2 briefs about the previous work carried out on script identification. The database constructed for training and 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 results obtained are presented in section 6. Conclusions are given in section 7.

2. LITERATURE SURVEY

Automatic script identification has been a challenging research problem in a multilingual environment over the last few years. All existing works on automatic language identification are classified into either local approach or global approach. Local approaches extract the features from a list of connected components like line, word and character in the document images and hence they are well suited to the documents where the script type differs at line or word level. In contrast, global approaches employ analysis of regions comprising at least two lines and hence do not require fine segmentation. As a result, global approaches are faster since no segmentation of the document image into lines, words and characters is required. Global approaches are applicable to those documents where the whole document or paragraph or a set of text lines is in one and only one script. The script identification task is simplified and performed faster with the global rather than the local approach. Ample work has been reported in literature on both Indian and non-Indian scripts using local and global approaches

2.1 Local approaches on Indian scripts

Majority of the work on Indian script identification has been carried out by Pal, Choudhuri and their team [1, 3, 5,9]. 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 binary tree classifier seems to be complex since the features are extracted at line, word and even at character level. From the literature it is observed that adequate work has been carried out on bi-lingual and tri-lingual documents of Indian languages specifically with respect to some Indian states [7, 8, 9, 10, 11, 12, 13]. Basavaraj Patil et. al. [7] have proposed a neural network based system for script identification of Kannada, Hindi and English languages. Word level script identification in bilingual documents through discriminating features has been developed by Dhandra et. al. [8]. A method to automatically separate text lines of Roman, Devnagari and Telugu scripts has been proposed by Pal et. al. [9]. Lijun Zhou et. al. [10] have developed a method for Bangla and English script identification based on the analysis of connected component profiles. Padma et. al. [11, 12] have proposed a method based on visual discriminating features to identify Kannada, Hindi and English text lines. Vipin Gupta et.al. [13] have presented a novel approach to automatically identify Kannada, Hindi and English languages using a set of features viz., cavity analysis, end point analysis, corner point analysis, line based analysis and Kannada base character analysis. Survey on the existing techniques of script identification of Indian documents shows that majority of the work is constrained to languages followed by a particular state. However, no work has been reported so far, that works in the direction of sorting a collection of documents printed in different languages. So, this paper addresses the problem of script identification useful for sorting an archive of documents printed in different languages.

2.2 Global approaches on Indian scripts

Adequate amount of work has been reported in literature using global approaches [4, 6]. Santanu Choudhuri, et al. [4] have 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. Almost all existing works on global approaches extract the features from the entire document image. Alternatively, it seems to be simpler and faster if only few text lines are considered for identification. With this idea, a global approach is proposed which involves segmentation only up to line level. Instead of considering the entire document image, as in the methods reported in literature, this method considers only few text lines for identification. The complete description of the new model is presented in the later section of this paper.

2.3 Local and global approaches on non-Indian scripts

Sufficient amount of work has also been carried out on non-Indian languages [2, 17, 19]. 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. Spitz has [20] proposed a method to discriminate between the Chinese based scripts and the Latin based scripts. Andrew Bhush [18] has presented a texture-based approach. Wood et al. [20] have proposed projection profile method to determine Roman, Russian, Arabic, Korean and Chinese characters. Hochberg et al. [22] have presented a method for automatically identifying script from a binary document image using cluster-based text symbol templates. Peake and Tan [23] 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.

Though much work is found in literature on Indian script identification, sufficient work on the triplets - Telugu, Devenagari and English scripts has not been carried out. Also, no work is reported so far that can give better recognition rate of identifying and separating text lines of Telugu, Devanagari and English scripts from a trilingual document. As a result, in this paper an automatic technique that can identify and separate text lines of Telugu, Devanagari and English scripts from a trilingual document is proposed.

3. DATA COLLECTION

Standard database of documents of Indian languages is currently not available. Data base construction with respect to the language identification problem seems to be complex since the factors like the font type and font size of each language needs to be considered. In this paper, it is assumed that the input data set contains documents having the text lines of Telugu, Devanagri and English scripts. Also, it is assumed that the language type, font and size of the text words within a text line are same.

For the experimentation of the proposed model, three sets of database are constructed, out of which one database was used to train the proposed system and the other two databases were constructed to test the system. The size of the document images considered were 600x600 pixels having about six to ten text lines depending upon the font size of the text. The document of English language was created using the Microsoft word software and these text lines were imported to the Micro Soft Paint program. In the Micro Soft Paint, a portion of the text lines was saved as black and white BitMaP (BMP) image having 600X600 pixels. The font type of Times New Roman, Arial, Bookman Old Style and Tahoma were used for English language. The font sizes of 12 to 26 were used for English text lines. The input images of Telugu and Hindi language were constructed by clipping only text portion of the document downloaded from the Internet. The training database is constructed such that 500 text lines were considered from each of the three languages.

To test the proposed model, two different data sets were constructed out of which one dataset was constructed manually similar to the dataset constructed for training and the other data set was constructed from the scanned document images. The printed documents like textbooks and magazines were scanned through an optical scanner to obtain the document image. The HP Scan Jet 5200c series scanner was used to obtain the digitized images. The scanning was performed in normal 100% view size at 300 dpi resolution. Manually constructed dataset is considered as good quality dataset and the data set constructed from the scanned document images are considered as poor quality data set. The test datasets were constructed such that 300 text lines from each of the three languages - Telugu, Hindi and English, were present from each of the good quality and poor quality datasets.

4. PREPROCESSING

Any language identification method requires conditioned image input of the document, which implies that the document should be noise free and skew free. Apart from these, some recognition techniques require that the document image should be segmented, thresholded and thinned. All these methods, help in obtaining appropriate features for text language identification processes.

In this paper, the preprocessing techniques such as noise removal and skew correction are not necessary for the datasets that are manually constructed by downloading the documents from

the Internet. However, for the datasets that is constructed from the scanned document images, preprocessing steps such as removal of non-text regions, skew-correction, noise removal and binarization is necessary. In this paper, text portion of the document image was separated from the non-text region manually, though page segmentation algorithm such as [24] could be readily be employed to perform this automatically. Skew detection and correction was achieved using the technique proposed by Shivakumar [17].

A global thresholding approach was 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 text area is segmented from the document image by removing the upper, lower, left and right blank regions. It should be noted that the text block might contain lines with different font sizes and variable spaces between lines. It is not necessary to homogenize these parameters, as the input to the proposed model is the individual text lines. 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. The segmented text lines might have varying inter-word spacing. So, it is necessary to normalize the inter-word spacing to a maximum of 5 pixels. Normalization of the inter-word spacing is achieved by projecting the pixels of each text line vertically; counting the number of white pixels from left to right and reducing the number of white pixels greater than 5 pixels to 5. Due to varying size of fonts, it is necessary to normalize the input text lines to fixed size. Through experimental observation, it was determined to fix the height of the text line as 40 rows that facilitate to extract the features efficiently. So, the input text line of size m rows and n columns resized to fixed size of 40 rows and (40 x n/m) columns keeping the aspect ratio. Then, a bounding box is fixed for the segmented and resized text line by finding the leftmost, rightmost, topmost and bottommost black pixel of each text line. Also, it is necessary to preprocess the text line by thinning as the texts may be printed in varying thickness. In this paper, thinning is performed by using the morphological operations. A sample text line that has undergone thinning operation to a single pixel width is shown in Figure 1. Thus, the normalized image of the bounded text line is prepared ready for further processing such as feature extraction.

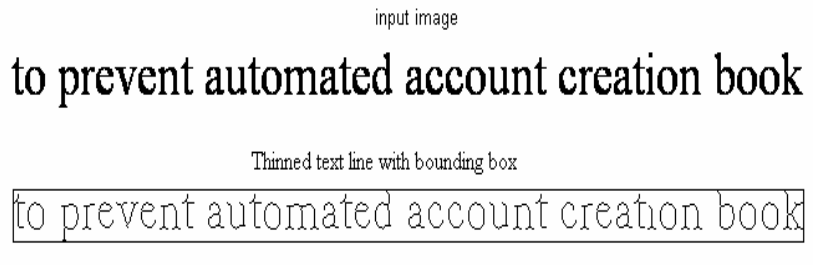

Figure 1. Input image of English text line and its image after thinning operation.

5. PROPOSED MODEL

Every script defines a finite set of text patterns called alphabets. Alphabets of one script are grouped together giving meaningful text information in the form of a word, a text line or a paragraph. Thus, when the alphabets of the same script are combined together to yield meaningful text information, the text portion of the individual script exhibits a distinct visual appearance. The distinct visual appearance of every script is due to the presence of the segments

like – horizontal lines, vertical lines, upward curves, downward curves, descendants and so on. The presence of such segments in a particular script is used as visual clues for a human to identify the type of even the unfamiliar script. It was motivated to adopt the idea of human visual perception capability into the proposed model to use the distinct features exhibited by each script. So, the target of this paper is to identify the script type of the texts without reading the contents of the document.

By thoroughly observing the structural outline of the characters of the three scripts - Telugu, Devanagari and English, it is observed that the distinct features are present at some specific portion of the characters. So, in this paper, the discriminating features are extracted from the top-profile and the bottom-profile of each text line. The top-profile (bottom-profile) of a text line represents a set of black pixels obtained by scanning each column of the text line 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 number having the maximum number of black pixels (black pixels having the value 0's correspond to object and white pixels having value 1's correspond to background) in the top-profile (bottom-profile) is called the attribute top-max-row (bottom-max-row).

5.1 Properties of Telugu, Devanagari and English Languages

Telugu is the official language of one of the South Indian state - Andra Pradesh. Telugu script is derived from Telugu script itself. It can be seen that, most of the Telugu characters have tick-shaped structures at the top portion of their characters. Also, it could be observed that majority of Telugu characters have upward curves present at their bottom portion. These distinct properties of Telugu characters are helpful in separating them from Hindi and English languages.

It could be noted that many characters of Devanagari script have a horizontal line at the upper part called headline which is named as sirorekha [1] in Devanagari. 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 for each text word. These long horizontal lines are present at the top portion of the characters and they are used as supporting features in identifying Devanagari script. Another strong feature that could be noticed in a Devanagari text line is that most of the pixels of the headline happen to be the pixels of bottom profile (bottom profile is defined in the later section). This results in both top and bottom profiles of a Hindi text line to lie at the top portion of the characters. However this distinct feature is absent in both Telugu and English text lines where the density top and bottom profiles occur at different positions. Using these features Hindi text line could be strongly separated from Telugu and English languages.

It is observed that the pixel distribution in most of the English characters is found to be symmetric and regular. This uniform distribution of the pixels of English characters results in the density of the top profile to be almost same as the density of the bottom profile. However, such uniformity found in pixel distribution of the top and bottom profiles of an English text line is not found in the other two anticipated languages Telugu and Hindi. Thus, this characteristic attribute is used as a supporting feature to separate an English text line. Thus, the distinct characteristic structures of each language are used as supporting visual features in the proposed model.

5.2 Feature Extraction

The distinct features used in the proposed model are extracted as explained below:

Feature 1: Bottom-max-row-no: The feature bottom-max-row-no represents the row number of the bottom-profile at which the maximum number of black pixels lies (black pixels having the value 0's correspond to object and white pixels having value 1's correspond to background).

Feature 2: Top-horizontal-line: (i) Obtain the top-max-row from the top-profile. (ii) Find the components whose number of black pixels is greater than threshold1 (threshold1 = half of the height of the bounding box) and store the number of such components in the attribute horizontal-lines. (iii) Compute the feature top-horizontal-line using the equation (1) below:

$$\text{Top-horizontal-line} = (\text{hlines} * 100) / \text{tc} \quad (1)$$

where hlines represent number of horizontal lines and tc represents total number of components of the top-max-row.

Feature 3: Tick-component: The observation of the characters of Telugu script motivated to use the tick shaped components as a feature. A component is said to have the shape of the tick-like structure if the pixel values of the components are in the sequence (i, j), (i+1, j+1), (i+2, j+2), … , (i+m1, j+n1), (i+m1-1, j+n1+1), (i+m1-2, j+n1+2), (i+m1-3, j+n1+3), ... , (i+m1-m2, j+n1+n2), where m2=i+m1 and n2>n1. The shape of the tick-like structure is shown in Figure 2. The component having the shape of the tick-like structure (√) is named as the feature 'tick-component'. Such tick-components extracted from the top portion of Telugu script are shown in Figure 3.

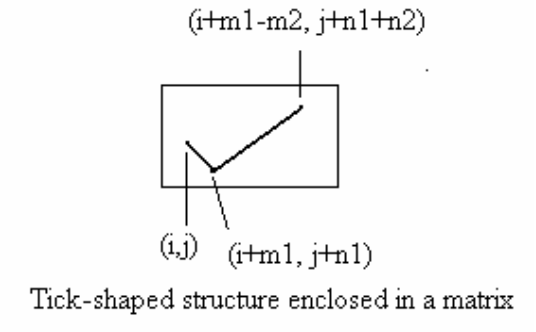

Figure 2. A tick shaped component.

Feature 4: Bottom-component: If more than 50% of the connected component is present below the attribute bottom-max-row, then that connected component is considered as the descendent. The presence of descendants or vathaksharas found at the bottom portion of the Telugu 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

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 Telugu script are shown in Figure 2.

Feature 5: Top-pipe-size: The attribute top-pipe (bottom-pipe) is obtained by deleting the connected components whose number of pixels is less than threshold2. The value of threshold2 is computed through experimentation and it is fixed to 10 pixels. The number of rows comprising the top-pipe is used as the feature top-pipe-size.

Feature 6: Bottom-pipe-size: The feature bottom-pipe-size is computed as that of top-pipe-size.

Feature 7: Top-pipe-density: The feature top-pipe-density is computed using the equation (2).

$$\text{top-pipe-density}=(nbp*100)/(m*n) \quad (2)$$

where nbp correspond to number of black pixels present in the top-pipe and (m,n) is the size of the image - top-pipe.

Feature 8: Bottom-pipe-density: The feature bottom-pipe-density is computed as that of top-pipe-density.

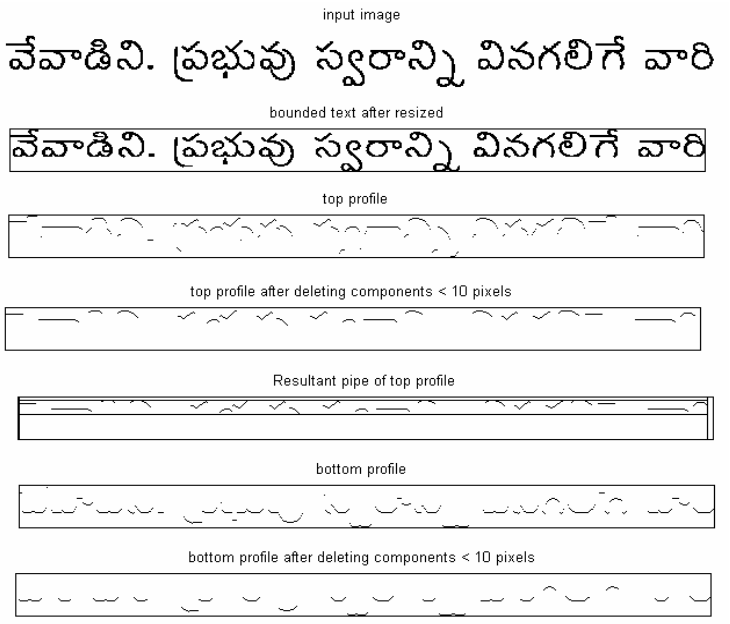

Figure 3. Sample output image of Telugu text line.

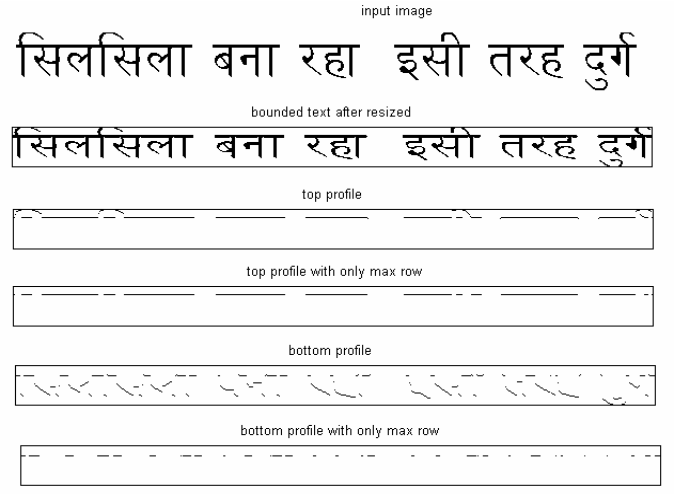

Figure 4. Sample output image of Hindi text line.

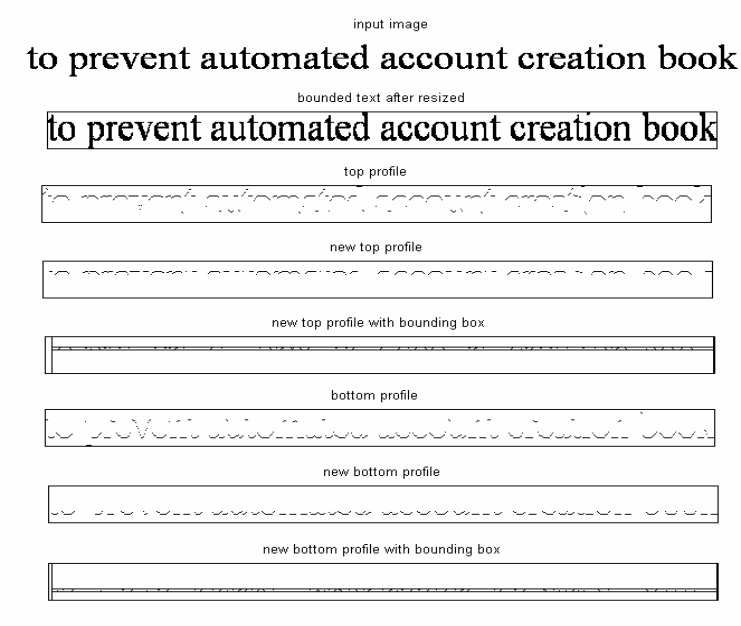

Figure 5. Sample output image of English text line.

5.3 Feature Selection

Feature selection is a process of minimizing the number of features and maximizing the discriminating property of the feature set. Feature selection is a process that aims to identify an optimal subset of relevant features from a large number of features collected in the data set, such that the overall accuracy of classification is increased.

In this paper, the values for the above-explained eight features are experimentally computed using a training data set of 500 text lines from each of Telugu, Devanagri and English scripts. The mean value of one feature of a specific script is computed by summing the feature values of that script divided by the total number of text lines. Thus, the mean value of each of the eight features is computed for all the three scripts and is given in Table 1. It is observed from the Table 1 that, among the eight feature values, some feature values are more discriminating for specific scripts whereas some features values are overlapped. For example, the feature 'bottom-max-row' of Devanagari script has distinct value of 13 when compared to all other scripts. Similarly, the feature tick-component is present in Telugu script only and absent in the other two scripts. Such features, which are present in only one script and yield more discriminating values among the other scripts, are selected as the optimal features and they are shown in the shaded form in Table 2. Thus, three sets of optimal features are selected from the eight features {F1, F2, F3, F4, F5, F6, F7 and F8} for each script and they are given below:

Dev-set = {F1, F2} for Devanagari script

Tel-set = {F3, F4, F5, F7} for Telugu script

Eng-set = {F5, F6, F7, F8} for English script

where F1 through F8 are the features as given in the Table1. The three sets of optimal features selected for each of the three scripts Telugu, Devanagari and English are used as supporting features in the learning process of the proposed model. Using the training data set of 500 text lines from each of the three scripts, a range of feature values was computed for each script and they are given in Table 2. The range of the distinct feature values of all the seven scripts is stored in a separate supportive knowledge base for later use during classification. The effectiveness of the identification of each script is better achieved when the values of all the optimal features are combined together.

Table 1. Mean value of the features for the three scripts.

	Features	Telugu	Hindi	English
1.	Bottom-max-row-no	27	13	31
2.	Top-horizontal-line	0%	68%	0%
3.	Tick-top	2	0	0
4.	Bottom-component	2	0	0
5.	Top-pipe-size	18	1	3
6.	Bottom-pipe-size	20	23	3
7.	Top-pipe-density	3.24	63.75	20.41
8.	Bottom-pipe-density	2.3334	2.54	19.7358

Table 2. Knowledge base of the range of feature values for the three scripts.

	Features	Telugu	Hindi	English
1.	Bottom-max-row-no	---	13	---
2.	Top-horizontal-line	---	55%-85%	---
3.	Tick-top	2-3	---	---
4.	Bottom-component	2-3	---	---
5.	Top-pipe-size	16-20	---	2-4
6.	Bottom-pipe-size	---	---	2-4
7.	Top-pipe-density	2.47-4.72	---	18.52-22.84
8.	Bottom-pipe-density	---	---	17.87-22.26

5.4 The Learning Algorithm

Using the optimal features of each script, the proposed model is learnt with a training data set of 500 text lines from each of the three scripts - Telugu, Devanagari and English. Learning algorithm used in the proposed model is given below.

Algorithm Learning ()
Input: Pre-processed text lines of Telugu, Devanagari and English scripts
Output: Range of feature values.

1. Do for i = 1 to 3 script types
2. { Do for k = 1 to 500 text lines of i th script
3. { Obtain top profile and bottom profile.
4. Compute the values of the optimal features of the i th script. }
5. Find minimum, maximum and mean of the optimal features for n text lines and store them in a knowledge base. }

5.5 Testing Algorithm

In order to classify the given test text line, the input text line is tested for the presence of the features of the script that needs least number of features amongst the three scripts. Accordingly, the test text line is first tested to check whether it is a Devanagari script because only two features are sufficient for identification as shown in the shaded form in Table1. If the test text line is not identified as Devanagari script, then the text line is tested for Telugu script by extracting the features of Telugu script. Further, if the test text line is not identified as either Devanagari or Telugu, then it is tested for English script by extracting the features of English script. If the test text lie is not identified as any of the three scripts, then it is rejected. In this paper, a rule-based classifier is used to classify the test text line into any of the three scripts.

6. RESULTS AND DISCUSSION

The system has been trained to thoroughly understand the behaviour of the top and bottom profiles using a training data set of 500 text lines from each of the three scripts. The proposed algorithm has been tested on a test data set of 1400 document images containing about 300

documents from each script. 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 99.67%. From the Table 6, it could be observed that the 100% accuracy is obtained for Hindi script. This is because top and bottom profile based features show distinct behaviour in Hindi script. From the experimental observations, it is noticed that the recognition rate is 100% for Hindi script even for the text lines having only one word, whereas the recognition rate falls down for the text lines with one or two text words. The proposed algorithm is implemented using Matlab R2007b. The average time taken to identify the script type of the document is 0.08436 seconds on a Pentium-IV with 1024 MB RAM based machine running at 1.60 GHz. The proposed method is compared with [3, 7, 12, 13] as shown in Table 4.

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. The overall performance of recognition verses training data set size is shown in Figure 4. The algorithm is also tested with document images having text lines in different font type and font sizes and found that the recognition rate almost sustained.

Table 3. Percentage of Recognition on the good quality (manually created) data set.

Input/Output	Telugu	Hindi	English	Rejected
Telugu	99.5%	---	0.3%	0.2%
Hindi	---	100%	---	---
English	0%	---	99.5%	0.5%

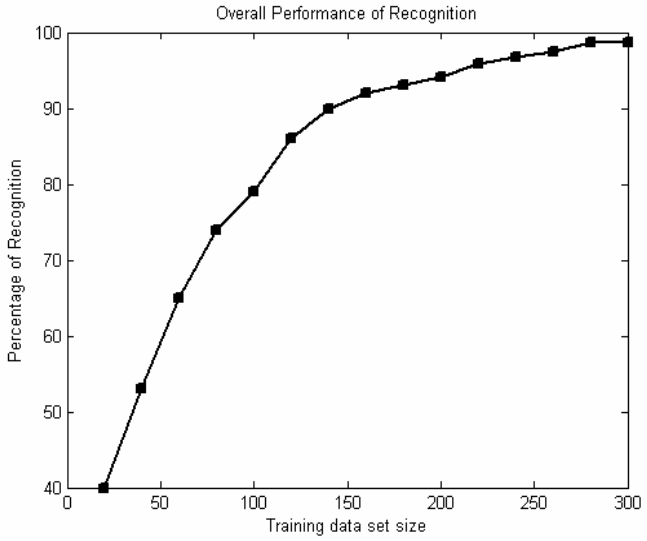

Figure 6. Overall Performace of Recognition verses trainng data set size.

Table 4. 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 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%	Features are extracted from individual characters and hence takes more time to extract the features..
[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	1400 text lines	Top and bottom profile based feature.	99%	Features are extracted from the entire text line and the optimal features are used.

7. CONCLUSION

In this paper, a method to identify and separate text lines of Telugu, Hindi and English scripts from a trilingual document is presented. The approach is based on the analysis of the top and bottom profiles of individual text lines and hence does not require any character or word segmentation. A document may contain text words of different languages within a text line. The proposed method is bound to fail on such documents. This is a limitation. One possible solution is to identify the script type at word level by segmenting the text line into words. Our future work is to consider identification of the script type at word level.

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] U. Pal and B. B. Chaudhuri, "Automatic separation of Roman, Devnagari and Telugu script lines", Advances in Pattern Recognition and Digital techniques, pp. 447-451, 1999.

[10] 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).

[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] M. C. Padma and P.A.Vijaya,: Language Identification of Kannada, Hindi and English Text Words Through Visual Discriminating Features, International Journal of Computational Intelligence Systems (IJCIS), Volume 1, Issue 2, pp. 116-126, (2008).

[13] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins,: Digital Image Processing using MATLAB, Pearson Education, (2004).

[14] 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).

[15] Brunzell H. and Eriksson J., "Feature Reduction for Classification of Multidimensional Data", Pattern Recognition, 33, pp. 1741-1748, 2000.

[16] Sutcliffe, J. P., "On the logical necessity and priority of a monothetic conception of class, and on the consequent inadequacy of polythetic accounts of category and categorization", http://www.db.dk/bh/lifeboat_ko/CONCEPTS/monothetic.html

[17] 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.

[18] 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.

[19] Murali, Vasudev, Hemanthkumar, Nagabhushan, 2006, "Language Independent Skew Detection and Correction of Printed Text Document Images: A Non-rotational Approach", VIVEK-International Journal of Artificial Intelligence, Vol. 16, No. 2, pp 08-15.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] A. K. Jain and Y. Zhong, "Page Segmentation using Texture Analysis", Pattern Recognition 29, pp743-770, 1996.