International Journal of Computer Science and Applications,
©Technomathematics Research Foundation
Vol. 7 No. 4, pp. 16 - 33 , 2010

SCRIPT IDENTIFICATION FROM TRILINGUAL DOCUMENTS USING PROFILE BASED FEATURES

M. C. PADMA*

Dept. of Computer Science & Engg.,PES College of Engineering,
Mandya-571401,Karnataka, India,
Email: padmapes@gmail.com

P. A. VIJAYA

Dept. of E & C Engg.,Malnad College of Engineering,
Hassan-573201, Karnataka, India
Email: pavmkv@yahoo.co.in

In a multi script environment, majority of the documents may contain text information printed in more than one script/language. For automatic processing of such documents through Optical Character Recognition (OCR), it is necessary to identify different script regions of the document. In this paper, it is proposed to develop a model to identify the script type of a trilingual document printed in Kannada, Hindi and English scripts. The distinct characteristic features of Kannada, Hindi and English scripts are thoroughly studied from the nature of the top and bottom profiles. The proposed model is trained to learn thoroughly the distinct features of each script. Experimentation conducted involved 1500 text lines for learning and 1500 text lines for testing. The k-nearest neighbor classifier is used to classify the test sample. The results are encouraging and prove the efficacy of the proposed model. The average success rate is found to be 99.5% for data set constructed from scanned document images.

Keywords: Multilingual document processing, Script Identification, Top Profile, Bottom Profile, Feature extraction, K-Nearest Neighbor Classifier.

1. Introduction

Automatic script identification has been a challenging research problem in a multi script environment and has acquired importance through the years. As the world is moving electronically, there is a growing tendency of converting the physical documents into electronic forms for easier access and purposes of privacy and security.

* Assistant Professor and Head, Department of Computer Science and Engineering, PES College of Engineering, Mandya-571401, Karnataka, India.

But still, a large proportion of all kinds of business writing communication exist in physical form for various purposes. Two such purposes are to fax a document, to produce a document in the court. Also, there has been a strong need to identify the script type of the document to construct electronic libraries for the interconnected world. All of these tasks are grouped under the general heading of document image analysis, which has been an increasing emerging 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 and undesirable. 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. So, a preprocessor to the OCR system is necessary to identify the script type of the document, so that specific OCR tool can be selected. In this context, the problem of script identification is addressed here. Some practical application potentials of automatic script/language identification schemes are (i) to sort document images, (ii) to select specific OCRs, (iii) to search online archives of document image for those containing a particular language and (iv) to design a multi-script OCR system.

India is a multi script multi lingual country having 18 regional languages derived from 12 different scripts [Pal and Chaudhuri 1999]. According to the three-language policy adopted by most of the Indian states, the documents produced in an Indian state Karnataka, are composed of texts in the regional language-Kannada, the National language-Hindi and the world wide commonly used language-English. In addition, majority of the documents found in most of the private and Government sectors of Indian states, are tri-lingual type (a document having text in three languages). So, there is a growing demand to automatically process these tri-lingual documents in every state in India, including Karnataka. For automatic processing of such tri-lingual documents through the respective OCRs, a pre-processor is necessary which could identify the script type of the text lines. In this paper, it is proposed to develop a model to identify and separate text lines of Kannada, Hindi and English scripts from a trilingual document. Here, the terms script and language could be interchangeably used as the three languages - Kannada, Hindi and English belong to three different scripts.

The rest of the paper is organized as follows. Related literature work is given in Section 2. The Section 3 describes useful discriminating features and the new model developed to identify the three anticipated languages. The details of experimental results obtained are presented in Section 4. Conclusion is given at Section 5.

2. Previous Work

Existing works on automatic script 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 of at least two lines and hence do not require fine segmentation. Global approaches are applicable to those documents where the whole document or paragraph or a set of text lines is in one script only. 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

Pal and Chaudhuri [Pal and Chaudhuri 1999, 2003; Pal et al 2003] have reported majority of the work on Indian language identification. Pal and Choudhuri [Pal and Chaudhuri 1999] have proposed an automatic technique of separating the text lines from 12 Indian scripts (English, Hindi, 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 [Pal et al 2003] 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 some Indian states [Basavaraj Patil and Subbareddy 2002, Dhandra et al 2006, Vipin 2006, Lijun et al, 2006]. Basavaraj Patil et. al. [Basavaraj Patil and Subbareddy 2002] 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. [Dhandra et al 2006]. They [Dhandra et al 2006] have exploited the use of discriminating features (aspect ratio, strokes, eccentricity, etc,) as a tool for determining the script at word level in a bi-lingual document containing Kannada, Tamil and Devnagari containing English numerals. A method to automatically separate text lines of Roman, Devanagari and Telugu scripts has been proposed by Pal et. al. [Pal et al 1999]. Lijun Zhou et. al. [Lijun et al, 2006] have developed a method for Bangla and English script identification based on the analysis of connected component profiles. Padma et. al. [Padma 2002] have proposed a method using horizontal and vertical linear edge features as visual clues to identify Kannada, Hindi and English text lines. Vipin Gupta et.al. [Vipin 2006] have presented an 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.

2.2 Global approaches on Indian scripts

Adequate amount of work has been reported in literature using global approaches [Santanu Chaudhury et al. 2000, Hiremath 2008, Ramachandra and Biswas, 1997]. Santanu Choudhuri, et al. [Santanu Chaudhury et al. 2000] 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. [Gopal et al 2006] have presented a script identification technique for 10 Indian scripts using a set of features extracted from log-Gabor filters. Dhanya et al. [Dhanya et al 2002] have used Linear Support Vector Machine (LSVM), K-Nearest Neighbour (K-NN) and Neural Network (NN) classifiers on Gabor-based and zoning features to classify Tamil and English scripts. Hiremath et al. [Hiremath 2008] have proposed a novel approach for script identification of South Indian scripts using wavelet based co-occurrence histogram features. Ramachandra Manthalkar et.al. [Ramachandra and Biswas, 1997] have proposed a method based on rotation-invariant texture features using multi channel Gabor filter for identifying seven Indian languages namely Bengali, Kannada, Malayalam, Oriya, Telugu and Marathi. They [Ramachandra and Biswas, 1997] have used multichannel Gabor filters to acquire rotation invariant texture features. From their experiment, they observed that rotation invariant features provide good results for script identification. Srinivas Rao Kunte et al. [Srinivas Rao Kunte and Sudhakar Samuel 2002] have suggested a neural approach in on-line script recognition for Telugu language employing wavelet features. Nagabhushan et al. [Nagabhushan et al, 2005] have presented an intelligent pin code script identification methodology based on texture analysis using modified invariant moments. Peeta Basa Pati et al. [Peeta et al 2004] have presented a technique using Gabor filters for script identification of Indian bilingual documents.

2.3 Local and global approaches on non-Indian scripts

Sufficient amount of work has also been carried out on non-Indian languages [2, 22-24]. One of the first attempts in automatic script and language identification is due to Spitz and his co-researchers [Spitz 1994]. Spitz has proposed a system, which relies on specific, well defined pixel structures for script identification [Spitz 1994]. Such features include locations and numbers of upward concavities in the script image, optical density of connected components, the frequency and combination of relative character heights. This approach has been shown to be successful in distinguishing between Asian languages (Japanese, Chinese, and Korean) against European languages (English, French, German, and Russian). Wood et al. [Wood et al 1995] have proposed projection profile method to determine Roman, Russian, Arabic, Korean and Chinese characters. Hochberg et al. [Hochberg et al 1997] have presented a method for automatically identifying script from a binary document image using cluster-based text symbol templates. The system develops a set of representative symbols (templates) for each script by clustering textual symbols from a set of training documents and represents each cluster by its centroid. In

[Ding et al 1997], a method that uses a combined analysis of several discriminating statistical features to classify Oriental and European scripts is presented. Peake and Tan [Peake and Tan 1977] have proposed a method for automatic script and language identification from document images using multiple channel (Gabour) filters and gray level co-occurrence matrices for seven languages: Chinese, English, Greek, Korean, Malayalam, Persian and Russian. Tan et al. [Tan 1998] has proposed a rotation invariant texture feature extraction 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, Koreans, Malayalam, Persian and Russian. Andrew Busch et. al [Andrew et al 2005] has presented the use of texture features (gray level co-occurrence matrix and Gabor energy features) for determining the script of a document image.

It can be seen from the references cited above that ample amount of work has been done in the area of document script/language identification. Even though some considerable amount of work has been carried out on Indian script identification, hardly few attempts focus on the three languages - Kannada, Hindi and English, followed by Karnataka, an Indian state. So, an intensive work needs to be done in this field as the demand is increasing. Also the existing methods have to be improved to reach a stage of satisfactory practical application. It is in this direction the research work proposes a model that automatically identifies the three languages Kannada, Hindi and English from a trilingual document. The proposed method could be considered as a local approach as the script identification is done at line level by segmenting the input image into text lines.

3. Data Collection

Standard dataset of Indian scripts is currently not available. Data set construction with respect to the script identification problem seems to be complex since the factors like the font type and font size of each script needs to be considered. In this paper, it is assumed that the input data set contains text lines of the three scripts - Kannada, Hindi, and English. Also, it is assumed that the script type, font and size of the text words within a text line are same.

For the experimentation of the proposed model, three separate datasets are constructed, out of which one dataset is used to train the proposed system and the other two datasets are constructed to test the system. Thus separate data sets are constructed for training and testing. The document of Kannada and English scripts were created using the Microsoft word software and these text lines were imported to the Micro Soft Paint program. In the Microsoft Paint, a portion of the text lines was saved as black and white BitMaP (BMP) image of size 600X600 pixels. The font type of Times New Roman, Arial, Bookman Old Style and Tahoma were used for English language. The font type of Kannada Extended, Vijaya and Sirigannada are used for Kannada script. The font sizes of 12 to 48 were used for both Kannada and English text lines. The input image of Hindi script was constructed by clipping only text portion of the document downloaded from

the Internet. The training dataset comprised of 500 text lines from each of the three scripts.

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 size of the test image considered was 600x600 pixels comprising of five to eight text lines with different font sizes and font types. The printed documents like application forms, language-translation books, manuals 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 were comprised of 300 text lines and the data set constructed from the scanned document images were comprised of 200 text lines from each of the three scripts.

4. Preprocessing

Preprocessing is a method of enhancing the image for better feature extraction. The choice of preprocessing method to be adopted on a document image depends on the type of application for which the image is used. There are many techniques that are generally available to accomplish preprocessing on images; however, several experiments on script identification suggest that preprocessing methods have got to be customized to suit the requirements of script identification. Any script 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 and thresholded. All these methods, help in obtaining appropriate features for script 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. Skew detection and correction was achieved using the technique proposed by Shivakumar [Shivakumar et. al. 2006]. 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 image of size *m* rows and *n* columns is 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 a process called thinning as the texts may be printed in varying thickness. In this paper, thinning operation is achieved by using the morphological operations. A sample English and Kannada text line that has undergone thinning operation to a single pixel width is shown in Figures 1 and 2 respectively. Thus, the normalized image of the bounded text line is prepared ready for further processing such as feature extraction.

Figure 1. Thinned English Text Line

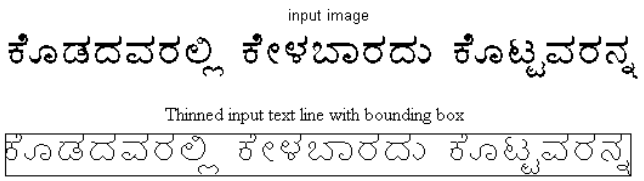

Figure 2. Thinned Kannada Text Line

5. The Proposed Model

The new model is inspired by a simple observation that every script/language defines a finite set of text patterns, each having a distinct visual appearance, which serves as useful visual clues to recognize the language. The character shape descriptors take into consideration any feature that appears to be distinct for the language and hence every language could be identified based on its discriminating features.

The proposed approach has adopted the concept of the top and bottom profiles of the input text lines proposed by Lijun Zhou et. al. [Lijun et al, 2006]. In [Lijun et al, 2006], the two languages - Bangla and English are identified using only one feature, which is obtained by computing the ratio of the sum of the differences of the black pixels of the top and bottom profiles. The method proposed in [Lijun et al, 2006] is not applicable for the trilingual documents as only one feature is used. With this backdrop, in this paper, a new model has been proposed that uses the concept of top and bottom profiles of a connected component proposed by Lijun Zhou et. al. [Lijun et al, 2006]. However, the new proposed method uses four features extracted from the top and bottom profiles of an input text line to identify the three anticipated languages - Kannada, Hindi and English.
The terms top_profile and bottom_profile of a text line are defined below:

Top_profile and Bottom_profile: 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 top_profile and bottom_profile of a text line are obtained through the algorithms 1 and 2 respectively.

Algorithm 1: Top_profile ()
Input: Preprocessed input text line - Matrix a.
Output: Top_profile - Matrix b.
1. Initialize matrix b=ones (size (a)) // The elements of the matrix b are initialized to 1's.
2. Do for j =1 to n columns
{ Do for i= 1 to m rows
{ If (a (i, j) == black)
{ b (i, j) = a (i, j) exit }
else continue
}
}
3. Return Matrix b.

Algorithm 2: Bottom_profile ()
Input: Preprocessed input text line - Matrix a.
Output: Bottom_profile - Matrix c.
1. Initialize matrix c=ones (size (a)) // The elements of the matrix c are initialized to 1's.
2. Do for j =1 to n columns
{ Do for i = m down to 1 rows

```
    {   If  (a (i, j) == black)
        { c (i, j) = a(i , j)  exit }
        else  continue
    }
  }
```

3. Return Matrix c.

5.1 Useful Discriminating Features of Kannada, Hindi and English text lines

It has been observed that the three scripts - Kannada, Hindi and English considered in this paper possess their own distinct features. These distinct features could be used as supporting features in the process of script identification system.

It could be observed that most of the Kannada characters have horizontal line like structures present at top portion of the characters. The pixels of these horizontal lines happen to be the pixels of the top profile. Also, it could be observed that majority of Kannada characters have upward curves present at their bottom portion. Hence for a Kannada text line, the density of the top profile is comparatively more than the density of the bottom profile of a given text line.

Many characters of Hindi language have a horizontal line at the upper part called headline or sirorekha [Pal and Chaudhuri 1999]. It could be seen that, when two or more 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. The pixels of these horizontal lines happen to be the pixels of the top profile. Also, in a Hindi text line most of the pixels of the headline happen to be the pixels of bottom profile.

It is observed that the most of the English characters are symmetric and regular in the pixel distribution. 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 uniform distribution of the pixels in top and bottom profiles of an English text line is not found in the other two anticipated languages - Kannada and Hindi. Thus, this characteristic attribute is used as a supporting feature to separate an English text line.

5.2 Feature Extraction from Top and Bottom Profiles

Choosing suitable features useful for discriminating the different text lines of a trilingual document is an important step. By thoroughly studying the nature of the top and bottom profiles of the three scripts, a set of distinct features that yield discriminating values are extracted. The features used in the proposed technique are chosen with the following considerations: (i) Easy to detect the features since the method does not require any character or word segmentation; (ii) Feasible for identification since the range of feature values obtained are found to be distinct for these three languages; (iii) Accuracy when all the four features are combined and (iv) Speed of computation.

The technical phrases that are used in this paper are defined below:

Top_max_row and Bottom_max_row: The attribute top_max_row (bottom_max_row) represents the row of the top_profile (bottom_profile) with maximum density i.e., the row with 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).

The features are extracted from the top and bottom profiles of the three scripts as explained below. Figures 3 to 5 show the output images of the text lines to extract the specific features.

Feature 1: Profile_value

It could be observed that the occurrence of the distinct characteristic features of the scripts is generally more concentrated near the top_max_row and bottom_max_row. The density (number of black pixels per unit area) of the pixels present at top_max_row and bottom_max_row is different for different scripts. The density of the two attributes top_max_row and bottom_max_row are combined to form one feature and hence, the ratio of the density of top_max_row and the bottom_max_row could be used as a feature named profile_value and it is computed as given in Equation (1).

$$Profile\_value = \frac{density\_top\_max\_row}{density\_bottom\_max\_row} \quad (1)$$

where density_top_max_row represents density at top_max_row and density_bottom_max_row represents density at bottom_max_row.

Feature 2: Bottom_max_row_no

Through experimentation, it is observed that the location of the attribute bottom_max_row is different for different scripts. Hence the value of the attribute bottom_max_row is used as the feature named 'bottom_max_row_no'. The value of bottom_max_row for Hindi script is found to be distinct among the three scripts and it is shown in Figure 3.

Feature 3: Coeff_profile

It is observed from the top and bottom profiles that the nature of the distribution of the black pixels is different for different scripts. This observation inspired us to compute the feature based on the spread of black pixels in the top and bottom profiles by using the formulae-coefficient of variation. The position value of the spatial occurrence of the black pixels of the top (bottom) profile is stored in a one-dimensional vector called top_vector (bottom_vector). The coefficient of variation of the top profile and bottom profile are computed by using the Equations (2) and (3) respectively and they are named as the attributes 'coeff_top' and 'coeff_bot' respectively.

$$coeff\_top = \frac{\sigma_{top\_vector}}{\mu_{top\_vector}} * 100 \quad (2)$$

$$coeff\_bot = \frac{\sigma_{bottom\_vector}}{\mu_{bottom\_vector}} * 100 \quad (3)$$

where σ and μ represents the standard deviation and mean of the top_vector and bottom_vector respectively.

Then the two attributes 'coeff_top' and 'coeff_bot' are combined to get one feature named 'coeff_profile' and it is obtained by using the Equation (4).

$$coeff\_profile = \frac{coeff\_top}{coeff\_bot} \quad (4)$$

Feature 4: Top_component_density

From the top profile, the top_max_row is selected. The top_max_row represents a number of connected components. The density i.e., the number of pixels comprising the connected components varies from language to language. So, the density of the connected components at the top_max_row is used as a feature 'top_component_density'.

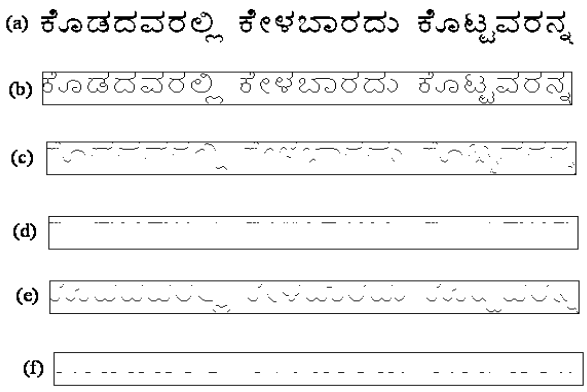

Figure 3. Output image of Kannada text line: (a) Input Text Line (b) Preprocessed Text Line (c) Top Profile (d) Top_max_row of Top Profile (e) Bottom Profile and (f) Bottom_max_row of Bottom Profile

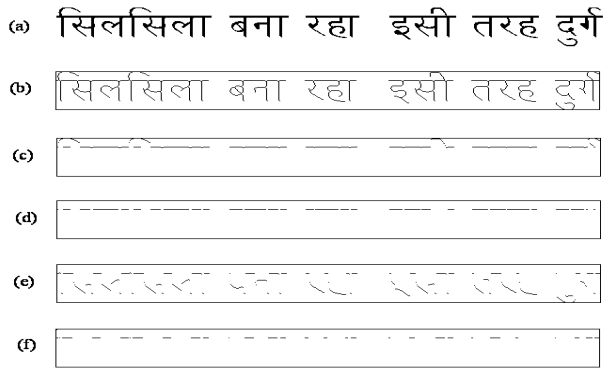

Figure 4. Output image of Hindi text line: (a) Input Text Line (b) Preprocessed Text Line (c) Top Profile (d) Top_max_row of Top Profile (e) Bottom Profile and (f) Bottom_max_row of Bottom Profile.

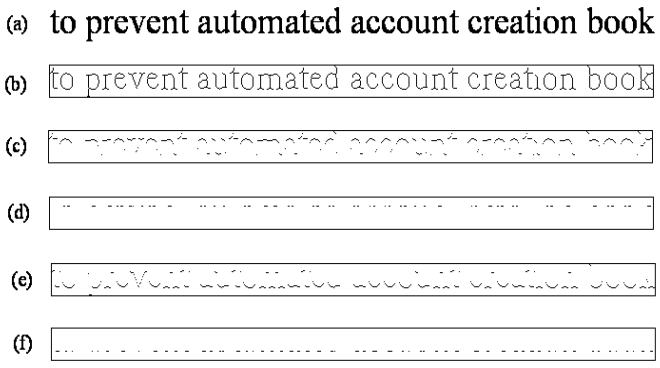

Figure 5. Output image of English text line: (a) Input Text Line (b) Preprocessed Text Line (c) Top Profile (d) Top_max_row of Top Profile (e) Bottom Profile and (f) Bottom_max_row of Bottom Profile

5.3 The Learning Algorithm

The proposed model is learnt with a training data set of 500 text lines from each of the three languages - Kannada, Hindi and English. The mean value (Mean= sum of feature values of 100 text lines / 100) of all the four features is computed for every 100 training data set of each language and stored in a knowledge base. Table 1 shows the mean value of the four features obtained through training dataset of 500 text lines. The Algorithm 3 is used in the learning phase of the proposed model.

Algorithm 3: Learning ()

Input: Pre-processed text lines of Kannada, Hindi and English scripts (m=3 and n=500).
Output: Knowledge base stored with the feature values of the three scripts.

```
1. Do for k = 1 to m language types
2. {  Do for j=1 to 5
3.     {   Do for i = 1 to 100 text lines
4.          {  Call top_profile() and bottom profile() function.
               Compute the four feature values-profile-value, bottom_max_row_no,
                  coeff-profile and top-component-density.
            }
5.     Find the mean of all the four features of 100 text lines and
          store them in a knowledge base.
       }  /*  Repeat for each 100 text lines */
   }      /*  Repeat for each language type */
```

Table 1. Mean value of the features obtained through training data set.

Script Type	Feature 1 (Profile_value)	Feature 2 (Bottom_max_row_no)	Feature 3 (Coeff_profile)	Feature 4 (Top_component_Density)
Kannada	1.7859	27	2.6231	55%
Hindi	2.4964	13	0.9245	75%
English	0.9794	31	2.1476	10%

5.4 Classification

In the proposed model, *K*-nearest neighbor classifier is used to classify the test samples. The four features are extracted from the test image *X* and these feature values are compared with feature values stored in the knowledge base. The Euclidean distance formula given in equation (5) is used to measure the distance between the test sample and the *k* neighbors. The Euclidean distance formula is given as

$$D(M) = \sqrt{\sum_{j=1}^{N} [f_j(x) - f_j(M)]^2} \tag{5}$$

where *N* is the number of features in the feature vector *f*, $f_j(x)$ represents the *j*th feature of the test sample *X* and $f_j(M)$ represents the *j*th feature of *M*th class in the knowledge base. Then, the test sample *X* is classified using the *k*-nearest neighbor (*K*-NN) classifier. In the *K* -NN classifier, a test sample is classified by a majority vote of its *k* neighbors, where *k*

is a positive integer, typically small. If K =1, then the sample is just assigned the class of its nearest neighbor. It is better to choose K to be an odd number to avoid tied votes. So, in this method, the K -nearest neighbors are determined and the test image is classified as the script type of the majority of these K-nearest neighbors. The experiment is conducted for varying number of neighbors like K = 3, 5 and 7. The performance of classification was best when the value of K = 3.

The Algorithm 4 is used to test the proposed model in classifying the input text lines.

Algorithm 4: Testing ()

Input: Document image containing text lines of Kannada, Hindi and English scripts.
Output: Script Type of Each Text lines.

1. Repeat for each test document image
2. Segment the document image into text lines.
3. Repeat for each text line
   { Preprocess the test text line.
   Call top_profile() and bottom_profile() function.
   Compute the four feature values: profile-value, bottom_max_row_no,
   coeff-profile and top-component-density.
   Compare these feature values with the range of feature values stored in the knowledge base using *k*-nearest neighbor classifier and classify the test text line to the type of the class of its nearest neighbor.
   }
4. Accuracy=(Number of text lines identified / Total number of text lines) * 100

6. Results and Discussion

The system is trained to thoroughly understand the nature of the top and bottom profiles using a training data set of 500 text lines from each of the three scripts. The proposed system is tested thoroughly using a manually created data set of 900 text lines obtained from 170 document images. Totally, 300 text lines from each of the three scripts are considered for testing. Each test image contained approximately five to eight text lines printed in different font type and font sizes. However, the font size and font type is assumed to be same within the text line. The font type of Times New Roman, Arial, Tahoma, Bookman Old Style are considered for English script. The font type of Kannada Extended, Vijaya and Sirigannada are used for Kannada script. Varying font size of 12 to 48 is considered for English and Kannada scripts. Test document images containing text lines in mixture of Kannada, Hindi and English languages were considered. Few test images containing text lines in only one script and mixture of two scripts were also considered. 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 success rate of classification has found to be 100% for manually created test data set.

The proposed system is also tested with scanned document images obtained from the text portions of application forms, manuals, language-translation books and such other trilingual documents. Data set of 200 text lines from each of the three scripts was considered from scanned images. Scanned images having text lines in various font type and font size are considered. Experiments conducted indicate that scanned images are recognized efficiently with an average recognition rate of 99.5%. The average success rate of manually created dataset and printed scanned data set has found to be 99.75%. This indicates the effectiveness of the proposed features. Table 2 gives the details of the recognition rate of Kannada, Hindi and English scripts on the manually created dataset, scanned data set and also handwritten data set.

We have found that 100% accuracy is obtained for English text lines with only uppercase letters. From the experimentation, we have noticed that better recognition rate is achieved for text lines of bigger font size than smaller font size for the scanned images. It is observed through the experimental study that the misclassification occurs only between Kannada and English text lines. The presence of digits and other symbols such as punctuations does not affect the recognition rate as they are rarely seen in a meaningful text line. Some of the text lines that got rejected while classifying are due to the presence of one or two text words.

Just for the sake of curiosity, we have also tested our algorithm on neatly handwritten document images even though the proposed system is developed mainly for printed documents. Script recognition of handwritten documents is more complex than the recognition of printed documents. The complexity is due to the variations found in writing the characters by different authors. Handwritten data set of 300 text lines obtained from 80 document images are used to test the proposed system. Satisfactory results are obtained with an average recognition rate of 93.67%. However, this recognition rate is dependent on the data set used to test the system.

Table 2. Recognition Rate of Kannada, Hindi and English Scripts

Type of Data Set		Kannada	Hindi	English
Printed Manually Created Data Set	**Correct Classification**	**100%**	**100%**	**100%**
	Misclassificatio	0%	0%	0%
	Rejection	0%	0%	0%
Printed Scanned Data Set	**Correct Classification**	**99.3%**	**100%**	**99.2%**
	Misclassificatio	0.4%	0%	0.5%
	Rejection	0.3%	0%	0.3%
Handwritten Scanned Data Set	**Correct Classification**	**93%**	**97%**	**91%**
	Misclassificatio	5%	2%	6%
	Rejection	2%	1%	3%

Results of the proposed method and the four methods given in the reference papers [Basavaraj et. al 2002, Pal et. al 2003, Dhany et. al 2002, Padma et. al 2003] are shown in Table 3. From the Table 3, it is seen that the proposed method yields comparatively a better recognition rate of success (99.75%). However, due to lack of standard (benchmark) Indian scripts data set, it is not possible to directly compare the performance of the proposed scheme with previously reported script classification schemes.

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

Previous work	Number of scripts	Database size	Proposed Technique	Performance	Remarks
[Basavaraj et al 2002]	3	450 words	Neural network based system	98%	The method is tested only on manually created data sets
[Pal et al 2003]	12	750 text lines	Local Features: Water reservoir principle, contour tracing, profile, etc.	98%	More complex since the features are extracted from individual characters.
[Dhany et al 2002]	3	5000 words	Local Features: cavities, corner points, end points, connectivity.	99.2%	Appreciable work but limited to tri-lingual documents.
[Padma et al 2003]	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	7	2400 text lines	Local Features extracted from top and bottom profiles and k-nearest classifier.	99.75%	Simple method since the features are extracted from only top and bottom profiles and gives better accuracy.

7. Conclusion

In this paper, a new method to identify the script type of the trilingual document containing Kannada, Hindi and English text lines is presented. The proposed model is developed based on the distinct features extracted from the top and bottom profiles of the individual text lines. The method looks simple, as it does not require any character or word segmentation. Experimental results demonstrate that relatively simple technique can reach recognition rate of 99.5% for data set constructed from scanned document images. Future work is to develop script identification model at word level for the text lines with the words printed in different scripts. The proposed algorithm suggested in this paper could be modified to apply on the trilingual documents of other Indian states.

References

Andrew Busch; Wageeh W. Boles and Sridha Sridharan, (2005), Texture for Script Identification, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 11, pp. 1720-1732.

Basavaraj Patil S. and Subbareddy N.V., (2002), Neural network based system for script identification in Indian documents, Sadhana Vol. 27, Part 1, 83–97.

Dhandra B.V., Nagabhushan P., Mallikarjun Hangarge, Ravindra Hegadi, Malemath V.S., (2006), Script Identification Based on Morphological Reconstruction in Document Images, The 18th International Conference on Pattern Recognition (ICPR'06), Vol.No. 11-3, 950-953.

Dhanya D., Ramakrishnan A.G. and Pati P.B., (2002), Script identification in printed bilingual documents, Sadhana, vol. 27, 73-82.

Ding J., Lam L. and Suen C. Y., (1997), Classification of oriental and European Scripts by using Characteristic features, Proc. 4th ICDAR , 1023-1027.

Gopal Datt Joshi, Saurabh Garg, and Jayanthi Sivaswamy, (2006), Script Identification from Indian Documents, H. Bunke and A.L. Spitz (Eds.): DAS 2006, LNCS 3872, 255–267.

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.

Hochberg J., Kerns L., Kelly P. and Thomas T., (1997), Automatic script identification from images using cluster based templates, IEEE Trans. Pattern Anal. Machine Intell. Vol. 19, No. 2, 176–181.

Lijun Zhou, Yue Lu and Chew Lim Tan, (2006), Bangla/English Script Identification based on Analysis of Connected component Profiles, Proc. 7th IAPR workshop on Document Analysis System, New land, 234-254.

Nagabhushan P., Angadi S.A. and Anami B.S., (2005), An Intelligent Pin code Script Identification Methodology Based on Texture Analysis using Modified Invariant Moments, Proc. of ICCR, 615-623.

Padma M.C. and Nagabhushan P., (2002), Horizontal and Vertical Linear Edge Features as Useful Clues in the Discrimination of Multilingual (Kannada, Hindi and English) Machine Printed Documents, proc. of National Workshop on Computer Vision, Graphics and Image Processing (WVGIP), 204-209.

Pal U. and Chaudhuri B.B., (2003), Script line identification from Multi script documents, IETE journal Vol. 49, No 1, 3-11.

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

Pal U., Chaudhuri B.B., (1999), Script line separation from Indian multi-script document, Proc. 5th Int. Conf. on Document Analysis and Recognition (IEEE Comput. Soc. Press), 406–409.

Pal U., Sinha S. and Chaudhuri B.B., (2003), Multi-Script Line identification from Indian Documents, Proceedings of the Seventh International Conference on Document Analysis and Recognition, 880 – 884, 0-7695-1960-1/03 © 2003 IEEE.

Peake G.S. and Tan T.N., (1977), Script and Language Identification from Document Images, Proc. Workshop Document Image Analysis, vol. 1, 10-17.

Peeta Basa Pati, S. Sabari Raju, Nishikanta Pati and A. G. Ramakrishnan, "Gabor filters for Document analysis in Indian Bilingual Documents", 0-7803-8243-9/04/ IEEE, ICISIP, pp. 123-126, 2004.

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

Ramachandra Manthalkar and Biswas P.K., (1997), An Automatic Script Identification Scheme for Indian Languages, IEEE Tran. on Pattern Analysis And Machine Intelligence, vol.19, no.2, 160-164.

Santanu Chaudhury, Gaurav Harit, Shekar Madnani, Shet R.B., (2000), Identification of scripts of Indian languages by Combining trainable classifiers", Proc. of ICVGIP, India.

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.

Spitz A. L., (1994), Script and language determination from document images, Proc. of the 3rd Annual Symposium on Document Analysis and Information Retrieval, Las Vegas, Nevada, 229-235.

Srinivas Rao Kunte R. and Sudhakar Samuel R.D., (2002), A Neural Approach in On-line Script Recognition for Telugu Language Employing Wavelet Features, National Workshop on Computer Vision, Graphics and Image Processing (WVGIP), 188-191.

Tan T. N., (1998): Rotation invariant texture features and their use in automatic script identification, IEEE Trans. Pattern Anal. Machine Intell. PAMI, Vol.20, No. 7, 751–756.

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).

Wood S. L.; Yao X.; Krishnamurthy K. and Dang L., (1995): Language identification for printed text independent of segmentation, Proc. Int. Conf. on Image Processing, 428–431, IEEE 0-8186-7310-9/95.