Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Monothetic Separation of Telugu, Hindi and English Text Lines from a Multi Script Document

M.C. Padma
Dept. of E. & C. Engg.,
Malnad College of Engineering,
Hassan-573201, Karnataka, India
Email: padmapes@gmail.com

P. A. Vijaya
Dept. of E. & C. Engg.,
Malnad College of Engineering,
Hassan-573201, Karnataka, India
Email: 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 monothetic algorithmic model to identify and separate text lines Telugu, Hindi and English scripts from a printed multilingual document. The proposed method uses the distinct features of the target script and searches for the text lines that possess the anticipated features. Experimentation conducted involved 1500 text lines for learning and 900 text lines for testing. The performance has turned out to be 98.5%.**

Index Terms*— *Multi-script multi-lingual document, Script Identification, Feature extraction, Monothetic Classifier.

I. Introduction

From many years, almost all organizations are converting paper documents to electronic files to facilitate quicker additions, searches and modifications and also to prolong the life of such records. However, the usage of physical documents still finds its demand in majority of business documents and communications and the fax machine remains a vital tool of communication worldwide. 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 that is achieved through the tool Optical Character Recognition (OCR), which is broadly defined as the process of reading the optically scanned text by the machine. At present, all existing OCR's are developed to a specific language. So, for large archives of document images that contain different languages, there must be some way to categorize these documents before applying the proper OCR on them. If a document has multi-lingual segments, then analysis and recognition of the text portion becomes much more complex. So, a pre-processor to the OCR system is necessary that can identify the language type of the document, so that specific OCR tool can be selected.

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 research work, 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. In addition, majority of the documents produced in many of the private and Government sectors, railways, airlines, banks, post-offices of Indian states are of type tri-lingual (a document having text in three languages). So, working on a tri-lingual document of a particular state is important for an Indian scenario. Further, there is a growing demand to automatically process these tri-lingual documents in every state in India, including Andra Pradesh. So, when it comes to automation, assuming that there are three OCRs for Telugu, Hindi and English languages, a pre-processor is necessary by which the language type of the different texts lines are identified. With this context, it is proposed to work on the tri-lingual documents

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

would contain the texts in Telugu, Hindi and English languages. Generally, majority of the documents produced in the border regions of Andra Pradesh may even contain the languages Kannada, Tamil, Malayalam and Urdu, followed by the neighboring states of Andra Pradesh. Hence, the proposed method is made strong enough to identify only the three languages- Telugu, Hindi and English and to reject the text lines printed in other than three languages into a separate class called OTHERS. So, the goal of this paper is to develop a monothetic algorithmic model that searches for the distinct features of the target language in order to select only those text lines that possess the specified features of the target language.

In the context of Indian language document analysis, major literature is due to Pal and Choudhuri [1]. This group worked on automatic separation of words from multi-script documents by extracting the features from projection profile and water reservoir concepts. Tan [2] has developed rotation invariant texture feature extraction method for automatic script identification for six languages: Chinese, Greek, English, Russian, Persian and Malayalam. Pal and Choudhuri [3] have proposed an automatic technique of separating the text lines from 12 Indian scripts (English, Devanagari, Bangla, Gujarati, Tamil, Kashmiri, Malayalam, Oriya, Punjabi, Tamil, Telugu and Urdu) using ten triplets formed by grouping English and Devanagari with any one of the other scripts. 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. Chanda and Pal [5] have proposed an automatic technique for word wise identification of Devnagari, English and Urdu scripts from a single document. Gopal Datt Joshi, et. al. [6] has proposed script Identification from Indian Documents. Word level script identification in bilingual documents through discriminating features has been developed by B V Dhandra et. Al. [7]. Neural network based system for script identification (Kannada, Hindi and English) of Indian documents is proposed by Basavaraj Patil et. Al. [8]. Lijun Zhou et. Al. [9] has developed a method for Bangla and English script identification based on the analysis of connected component profiles. Our earlier methods [10, 11] were based on visual discriminating features for identification of Kannada, Hindi and English text lines. Though some considerable amount of work has been carried out on Indian languages, satisfactory work has not been reported specifically considering these three languages. In addition, existing methods on identification of these three languages needs further improvement to reach high recognition.

This paper is organized as follows. In Section II, monothetic classification is briefed out. The section III describes the proposed technique of language identification. The details of the experiments conducted and the states of results obtained are presented in section IV. Conclusions are given in section V.

II. MONOTHETIC CLASSIFICATION

One well-known division between kinds of classification systems and ways of classifying is between Aristotelian classification and prototype theory. An Aristotelian classification works according to a set of binary characteristics, which the object being classified is either present or not present. At each level of classification, enough binary features are adduced to place any member of a given population into one, and only one class. A technical classification system operating by binary characteristics is called monothetic if a single set of necessary and sufficient conditions is adduced and polythetic if a number of shared characteristics are used. Aristotelian models—monothetic or polythetic—have traditionally informed formal classification theory in a broad range of sciences, including biological systematic, geology, and physics.

A monothetic classifier searches for exact match and checks whether an item is present or not present. Monothetic classifier makes use of association analysis, where the classes defined by the objects possess attributes both necessary and sufficient to belong to a class. Monothetic models use a single descriptor as a basis for the partitioning. At each partition one descriptor is chosen. Polythetic models use several descriptors, which in most cases are combined in to an association matrix prior to clustering.

From the literature survey in the area of document script/language identification, it is observed that most of the existing methods adopt the polythetic classification. However, in some applications where it is necessary to search collection of document image for those containing a particular language, monothetic classification is well suited. Also, if the multilingual document has three or more number of language types, then extracting the set of features from each text line and then arriving at the decision of classifying into the specific language type becomes too complex. Rather, it seems to be better to search for the features of a target language in order to extract those text lines from a multilingual document. In this context, this paper proposes to develop a monothetic algorithmic model that aims at extracting the text line of the target language autonomously. Among the existing works reported so far, no work has been attempted to solve the problem of script identification using the monothetic approach.

III. THE PROPOSED MODEL

The proposed model is developed by thoroughly understanding the characteristic features of the top profile and bottom profile of the printed text lines in the three languages Telugu, Hindi and English. 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. Specific features that exhibit distinct values for these three languages are used in the proposed model.

The technical phrases used in this paper are defined below:

Top-max-row (Bottom-max-row): The attribute top-max-row (bottom-max-row) represents the row of the top (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).

Upward-curves (Downward-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.

A. Properties of Telugu, Hindi and English Languages

Telugu is the official language of one of the South Indian state - Andra Pradesh. Telugu language is derived from Telugu script itself. It can be seen that, most of the Telugu characters have tick-shaped structures and horizontal line-like structures top 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 Hindi language have a horizontal line at the upper part. This line is called sirorekha in Devanagari [1]. However, we shall call it as 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 for each text word. These long horizontal lines are present at the top portion of the characters. This kind of line however is absent in the lower part. Another strong feature that could be noticed in a Hindi text line is that most of the pixels of the headline happen to be the pixels of bottom profile. 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 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 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. Also upward-curve and downward-curve like structures are present at the bottom and top portion of majority of English characters. The presence of the distinct characteristic structures of each language is used as supporting visual features in the proposed model.

B. Feature Extraction

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

Horizontal-above-top-max-row:
(i) Obtain the top-max-row-no from the top-profile.
(ii) Obtain the portion of the top-profile from first row to top-max-row-no and call it top-portion. From the top-portion, search for the components that have number of pixels is greater than 5 and all five pixels occur in one row. Such components are called the feature top-horizontal-line.

Tick-Feature:
(i) Obtain the top-portion from the top-profile.
(ii) Obtain the tick-shaped component from the top-portion and use it as a tick-feature.

Bottom-components:
(i) Obtain the bottom-portion of the input text line by copying the portion of the input text line from bottom-max-row-no to last row of the text line.
(ii) From the bottom-portion, find a component whose number of pixels is greater than 8 pixels and name that component as a feature bottom-component.

Top-pipe-size (Bottom-pipe-size): The attribute top-pipe (bottom-pipe) is obtained by deleting the connected components whose number of pixels is less than *threshold1* and by deleting the black pixels from the scan line whose sum of the scan line pixels is less than *threshold2*. *Threshold1* and *threshold2* are fixed through experimentation. *Threshold1* is fixed to 5, 5 and 10 pixels and *threshold2* is fixed to 15, 10 and 10 pixels for Telugu, Hindi and English text lines respectively. The number of rows comprising the top-pipe (bottom-pipe) is used as the feature top-pipe-size (bottom-pipe-size).

Top-pipe-density: The feature top-pipe-density is computed using the equation (1).

$$\text{top-pipe-density}=(nbp*100)/(m*n) \qquad (1)$$

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

Bottom-pipe-density:
Compute the feature bottom-pipe-density as that of top-pipe-density.

Coeff-profile:
The attributes top-vector and bottom-vector represents the position of only the connected components of top and bottom profiles respectively. However, the size of these two vectors top-vector and bottom-vector varies from one text line to other depending on the occurrence of inter-word and inter-character gap of individual text line. As it is not advisable to compute any values from variable sized vectors, picking the first 250 elements equalizes the size of the two vectors top-vector and bottom-vector. Then the coefficient of variation of the top profile (coeff-top) and bottom profile (coeff-bot) is computed by using the equations (2) and (3) respectively.

$$\text{Coeff-top}=\sigma(\text{top-vector})/\mu(\text{top-vector})*100 \tag{2}$$

where coeff-top represents coefficient of variation of the top profile, σ and μ represents the standard deviation and mean of the top-vector.

$$\text{Coeff-bot}=\sigma(\text{bottom-vector})/\mu(\text{bottom-vector})*100 \tag{3}$$

where coeff-bot represents coefficient of variation of the bottom profile, σ and μ represents the standard deviation and mean of the bottom-vector.
Then the feature coeff-profile is obtained using the equation (4) and used as a feature to discriminate the three anticipated languages.

$$\text{Coeff-profile}=\text{coeff-top}/\text{coeff-bot} \tag{4}$$

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.

Top-horizontal-line:
(i) Obtain the top-max-row from the top-profile. (ii) Find the components whose number of pixels is greater than 10 and store the number of such components in the attribute horizontal-lines. (iii) Compute the feature top-horizontal-line using the equation (5) below:

$$\text{Top-horizontal-line} = (\text{hlines} * 100) / \text{tc} \tag{5}$$

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

Bottom-horizontal-line:
(i) Obtain the bottom-max-row from the bottom-profile. (ii) Find the components whose number of pixels is in the range 4 to 6 and store the number of such components in the attribute horizontal-lines. (iii) Compute the feature bottom-horizontal-line using the equation (6) below:

$$\text{bottom-horizontal-line} = (\text{hlines} * 100) / \text{tc} \tag{6}$$

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

Top-downward-curves (Bottom-upward-curves): The percentage of the downward-curves (upward-curves) is calculated from top-pipe and bottom-pipe respectively.

The proposed system is learned using a training data set of 500 text lines from each of the three languages and the range of these features values are stored in a separate knowledge base as is given in Table I, Table II and Table III for Telugu, Hindi and English text lines respectively. The mean value of all the features of the three languages shows the distinct values projecting the discriminating property of all the three languages.

TABLE I. KNOWLEDGE BASE OF TELUGU LANGUAGE

Features	Range of feature values	Mean of feature value
Tick-feature	1-5	3
Horizontal-above-top-max-row	1-4	2
Top-pipe-size	12-20	16
Bot-pipe-size	12-16	15
Top-pipe-density	3.75-5.68	4.7269
Bot-pipe-density	1.66-3.28	2.1426
Bottom-Upward-curves	30-50	42
Bottom-components	0-3	2

TABLE II. KNOWLEDGE BASE OF HINDI LANGUAGE

Features	Range of feature values	Mean of feature value
Coeff-profile	0.38-1.25	0.9245
Bottom-max-row-no	11-14	13
Top-horizontal-line (threshold>10 pixels)	40%-95%	68%
Bottom-horizontal-line (6>threshold>4 pixels)	50%-80%	74%
Top-pipe-size	1	1
Bottom-pipe-size	18-24	22

TABLE III. KNOWLEDGE BASE OF ENGLISH LANGUAGE

Features	Range of feature values	Mean of feature value
Top-downward-curves	65%-85%	75%
Bottom-Upward-curves	68%-92%	82%
Top-pipe-size	0-4	3
Bottom-pipe-size	0-4	3
Top-pipe-density	6.78-23.52	15.6587
Bottom-pipe-density	7.65-18.13	16.9752

C. Proposed Algorithm

The input document images are obtained by downloading the images from the Internet and hence do not require preprocessing such as noise removal and skew correction. However, the following preprocessing steps are required for the new model.

Step 1: Preprocessing: (i) The input document image is segmented into several text lines using the valleys of the horizontal projection computed by a row-wise sum of black pixels. (ii) A bounding box is fixed by finding the leftmost, rightmost, topmost and bottommost black pixel of each text line.

Step-2 Learning algorithm: The range of distinct feature values of Telugu, Hindi and English text lines is obtained using a training data set of 500 text lines from each language. Separate knowledge bases are constructed to store the range of feature values of Telugu, Hindi and English text lines.

Step-3 Recognition algorithm:
(i) The test text line is preprocessed as explained before. (ii) The top and bottom profiles of the test text line are obtained. (iii) The distinct feature values of the target language are computed from the top and bottom profiles. (iv) The feature values of the new text line are compared with the feature values stored in the knowledge base of the respective language. If the feature values of the test text line lie within the range of the target language, then the test text line is separated as target (Telugu/Hindi/English) language else, the text line is grouped into a separate class OTHERS.

By executing the proposed algorithm in parallel to identify each of the three languages - Telugu, Hindi and English as the target language, the text lines of the three languages are identified monothetically/autonomously.

If a multi script document having languages - Kannada, Hindi, Malayalam, Tamil, Telugu, Urdu and English is fed into the proposed algorithm, then the algorithm is capable of rejecting the text lines of other than the three languages Telugu, Hindi and English language and group them into a separate class called OTHERS.

IV. Results and discussion

Two sets of database were constructed for the proposed system, one for training and the other for testing. To train the system 500 text lines from each of the three languages were used. The size of the sample image considered was 512x512 pixels. The algorithm is tested with a test data set of 500 document images, which could contain around 900 text lines. The test document images consist of mixture of Kannada, Hindi, Malayalam, Tamil, Telugu, Urdu and English text lines. Sample output images of Telugu, Hindi and English text lines are shown in Figure 1, 2 and 3 respectively. Details of results obtained are tabulated in Table II. From the experimental analysis, it is observed that high accuracy rate is achieved when the font type and font size of the test image is same as that of images used in training data set. 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.

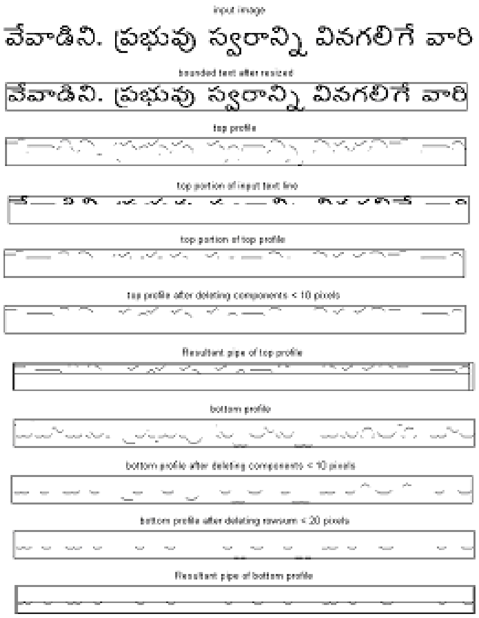
input image
వేవాడివి. ప్రభువు స్వరాన్ని వినగలిగే వారి
bounded text after resized
top profile
top portion of input text line
top portion of top profile
top profile after deleting components < 10 pixels
Resultant pipe of top profile
bottom profile
bottom profile after deleting components < 10 pixels
bottom profile after deleting rowsum < 20 pixels
Resultant pipe of bottom profile

Figure 1. Sample output image of Telugu text line.

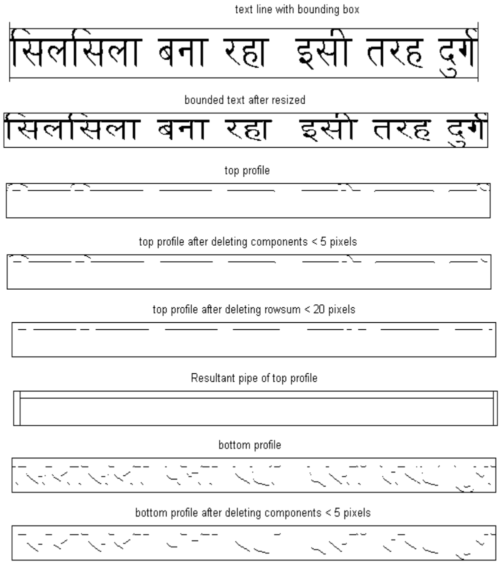
text line with bounding box
सिलसिला बना रहा इसी तरह दुर्ग
bounded text after resized
top profile
top profile after deleting components < 5 pixels
top profile after deleting rowsum < 20 pixels
Resultant pipe of top profile
bottom profile
bottom profile after deleting components < 5 pixels

Figure 2. Sample output image of Hindi text line.

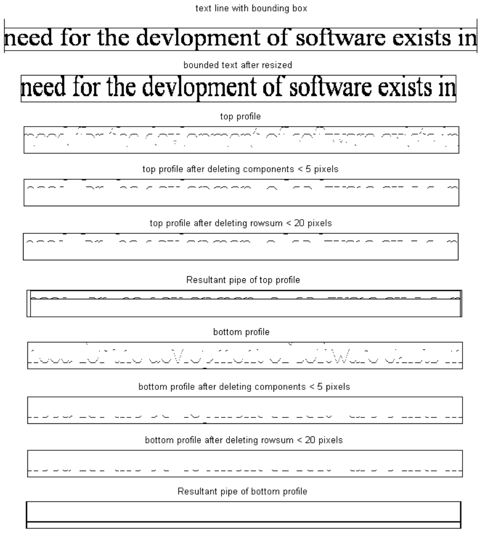
text line with bounding box
need for the devlopment of software exists in
bounded text after resized
need for the devlopment of software exists in
top profile
top profile after deleting components < 5 pixels
top profile after deleting rowsum < 20 pixels
Resultant pipe of top profile
bottom profile
bottom profile after deleting components < 5 pixels
bottom profile after deleting rowsum < 20 pixels
Resultant pipe of bottom profile

Figure 3. Sample output image of English text line.

TABLE IV. PERCENTAGE OF EXPERIMENTAL RESULTS. (* KANNADA, TAMIL, MALAYALAM AND URDU)

Input/Output	**Telugu**	**Hindi**	**English**	**OTHERS**
Telugu	98.2%	---	---	1.8%
Hindi	---	100%	---	0%
English	---	---	97.8%	2.2
OTHERS*	0.6%	0%	0.8	98.6%

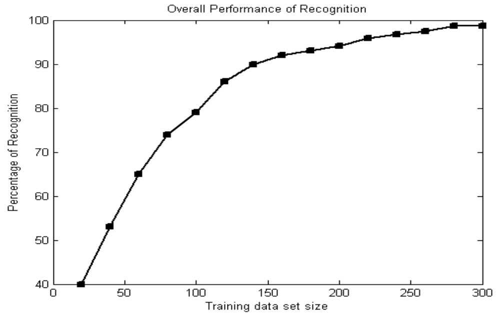

Figure 4. Overall Performace of Recognition verses trainng data set size.

V. CONCLUSION

In this paper, a monothetic algorithmic model to extract Telugu, Hindi and English text lines from printed multi script documents 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. The proposed method shows the applicability of monothetic approach to separate text lines of the target language from a multi script document. The performance of the proposed algorithm is encouraging when the proposed algorithm is tested using manually created test-data-base. However, the performance slightly comes down when the algorithm is tested on scanned document images due to many reasons like noise, skew-error, varying font type and character size and also text lines of different sizes. Our future work in this area is to develop algorithms to identify other Indian languages and also to identify the script type from a handwritten document.

REFERENCES

[1] U. Pal, S. Sinha and B. B. Chaudhuri "Multi-Script Line identification from Indian Documents", 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).

[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, July 1998.

[3] 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), 1999, 406-409.

[4] Santanu Choudhury, Gaurav Harit, Shekar Madnani, R.B. Shet, "Identification of Scripts of Indian Languages by Combining Trainable Classifiers", ICVGIP 2000, Dec.20-22, Bangalore, India.

[5] S.Chanda, U.Pal, English, Devanagari and Urdu Text Identification, Proc. International Conference on Document Analysis and Recognition, 2005, 538-545.

[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, February 2002, pp. 83–97. © Printed in India

[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. Feb. 22-24, 2007. pp.630-635.

[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] 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, 2003, pp. 252-260.

[11] 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.

[12] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins, "Digital Image Processing using MATLAB", Pearson Education, 2004.