ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

Script Identification from Multilingual Text Documents

A.H.Kulkarni[1], P.S.Upparamani[2], R J Kadkol[3], P.V. Tergundi[4]

Computer Science and Engineering Department, KLS Gogte Institute of Technology, Belagavi, Karnataka, India[1,4]

Information Science & Engineering Department, KLS Gogte Institute of Technology, Belagavi, Karnataka, India[2,3]

Abstract: In a multilingual country like India, a document may contain text words in more than one language. For a multilingual environment 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. But on the other hand, this causes practical difficulty in OCR such a document, because the language type of the text should be pre-determined, before employing a particular OCR. It is perhaps impossible to design a single recognizer which can identify a large number of scripts/languages. So, it is necessary to identify the language region of the document before feeding the document to the corresponding Optical Character Recognition (OCR) system. Identification aims to extract information presented in digital documents namely articles, newspapers, magazines and e-books. This has given rise to many language identification systems. The objective is to develop visual clues based procedure to identify different text portions of a document. In this work eight feature namely top max row, bottom max row, top horizontal lines, vertical lines, bottom components, tick components, top holes and bottom holes have been used to identify the script type.

Keywords: OCR, EDGE, PNN, KNN.

I. 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 project work 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, analyzes and stores information from physical documents for later retrieval. India is a multi-script multi-lingual country which has more than 18 regional languages derived from 12 different scripts. In such a country it is common to have multilingual documents as shown in Fig. 1. Example for such pages are bus reservation forms, question papers, language translation books and money-order forms that may contain text lines in more than one script/language forms[13].

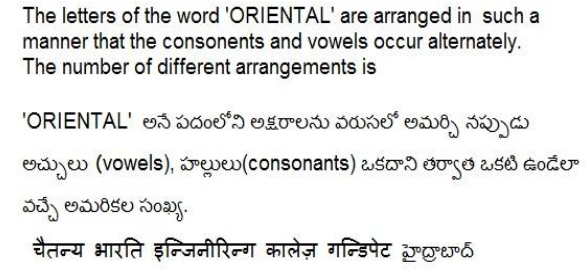


Figure 1. Sample multilingual document

One script could be used to write more than one languages. For example, Devanagari script is used by Hindi, Marathi, Rajasthani, Sanskrit and Nepali languages. 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. 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 script. It is difficult to feed a document as an input to OCR unless the language type of the text in it is pre-determined since a single OCR cannot recognize multiple languages. This can be solved by developing script identification systems. This addresses the need of developing tools that can recognize and analyze varied documents [13].

It can be seen that, most of the Telugu/Kannada characters have tick shaped structures at the top portion of their characters as shown in Fig. 2. Also, it could be observed that majority of Kannada characters have upward curves present at their bottom portion. These distinct properties of

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

Kannada characters are helpful in separating them from Hindi and English languages [8].

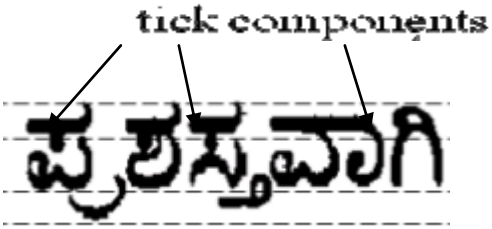


Figure 2 . Sample Kannada word with tick shaped structure

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 in Devanagari as shown in Fig. 3. It joins two or more basic or compound characters to form a word. These head lines are present at the top portion of the characters and they are used as supporting features in identifying Devanagari script. Another strong feature in a Devanagari 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 Kannada 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 Kannada and English languages [5].

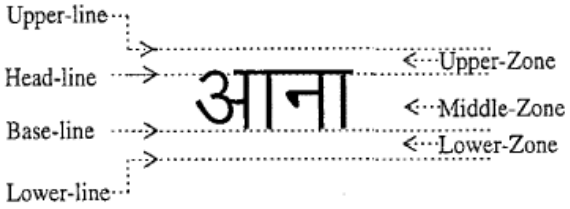


Figure 3. Sample Hindi word with partitions

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 Kannada and Hindi. Thus, this characteristic attribute is used as supporting visual features in the proposed model.

Although differences between different scripts are distinct in semantic level, it's hard for computers to comprehend them directly. Taking this into account, research on special distribution and visual attribute is necessary for document image analysis. And abstraction of structure feature becomes the key technology of script identification. Although the skew of a camera-based image is often more severe and unpredictable than that of a scanned image. Therefore, it is difficult for a component-based approach to train an appropriate representative character set from images of all possible skew angles.

II. EXISTING 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.

In one of the earlier, it is assumed that a given document should contain the text lines in one of the three languages Telugu, Hindi and English. In one of the previous papers, the results of detailed investigations were presented related to the study of the applicability of horizontal and vertical projections and segmentation methods to identify the language of a document considering specifically the three languages Telugu, Hindi and English. It is reasonably natural that the documents produced at the border regions of AP may also be printed in the regional languages of the neighboring states like Telugu, Tamil, Malayalam and Urdu. The system was unable to identify the text words for such documents having text words in Telugu, Tamil, Malayalam, Urdu languages.

III. PROPOSED SCRIPT IDENTIFICATION METHODS

From the literature it can be understood that most of the existing identification systems are either line, sentence or block level in this paper, we discuss two word-level script identification methods namely PNN approach and KNN based approach [13]. These methods use various visual features to identify the script type from multilingual documents.

Our proposed model consists of 4 phases. They are Training, knowledge base, Feature Extraction and Classification. A natural scene text image consisting of different scripts is given as input. Training involves applying discrete wavelet transform for feature extraction and linear discrete analysis for classification of different scripts. In the training phase, the texture features are extracted from the training samples selected randomly belonging to each script using the feature extraction

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

algorithm. These features are stored in the feature library. The database consists of distinct features of the scripts such as horizontal lines (head line), vertical lines and circle like structures in English and combination of vertical and horizontal, half rounded symbols and special symbols at the bottom.

The features which are stored in database are also used for future references. In the classification phase, based on the features extracted we classify the language into Hindi, English, and Kannada. The texture features are extracted from the test sample using the feature extraction algorithm and then compared with the corresponding feature values that are stored in the feature library.

All the features are extracted from the top and bottom profiles.

These profiles are defined as follows

Top_profile and Bottom_profile: Represent 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 following algorithms

Top_profile ()
Input: Preprocessed input text line - Matrix a.
Output: Top_profile - Matrix b.
1. Initialize matrix b of size m x n with 1's
2. for j =1 to n columns
3. for i = 1 to m rows
if a (i, j) == 0 (black)
b (i, j) = a (i, j)
break;
else
continue
4. Return Matrix b.

Bottom_profile ()
Input: Preprocessed input text line - Matrix a.
Output: Bottom_profile - Matrix c.
1. Initialize matrix c of size m x n with 1's.
2. for j =1 to n columns
for i = m down to 1 rows
if (a (i, j) ==0 (black)
c (i, j) = a(i , j)
break;
else
continue
3. Return Matrix c.

Top-max-row: Represents the row number of the topprofile 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).

Bottom-max-row: 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).

Top-horizontal-line: To compute *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:

*Top-horizontal-line = (hlines * 100) / tc ……(1)*

Training: The extracted features are then used to train the neural network. For this Probabilistic neural network has been used. We use an inbuilt function provided by MATLAB that is "newpnn".

Newpnn function is used to design a new Probabilistic neural network.

Syntax:

Var = newpnn(P,T,spread)
P- is a matrix of Q input vectors.
T- S X Q matrix of target class vectors.
Spread- spread of radial basis functions.

The output of newpnn is then saved in a file which is then used in the testing phase.

Testing: In the testing phase for the test Script the user is suppose to follow the same procedure till feature extraction to extract the features. We make use of a inbuilt function "sim" which compares the extracted features of the Script under test conditions to all the Scripts in the database, if the features are matched it displays the appropriate result to the user, else it displays it as "Unknown Script".

Algorithm for the proposed approach

Input: Sample script image.
Output: Recognition of script
Step 1: scan script images of Hindi, English and Kannada. Store the images in database.
Step 2: Read the sample images form stored database; preprocess the image for smoothening and noise removal. Segment the input image into the binary image using edge detection.
Step 3: Extract the interested features, Store the extracted features for training.
Step 4: Build the probabilistic neural network (PNN) or KNN for training & classification of images.
Step 5: Once the image is classified, the system shows the type script.
Step 6: End.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

IV. IMPLEMENTATION

Implementation details: This section deals with an implementation of the work carried. In this part image database used for training of the network is introduced, important steps carried out are discussed i.e., preprocessing & segmentation, then extraction of features from the segmented image and storing of those features in order to train the neural network for classification of the Script image.

Modules: Modules are sub parts of the system each designed for a specific purpose. Details of all the modules used in the system are explained below:

Data Acquisition: Collecting the images of Scripts of different languages (Hindi, English, and Kannada) for training as well as testing purpose. We have trained our system for 3 different Scripts which includes Hindi, English, and Kannada.

In order to train the system we took images of each Script from different angles. Once all the images were captured, few of them were selected for training and testing purpose. The images are as shown below.

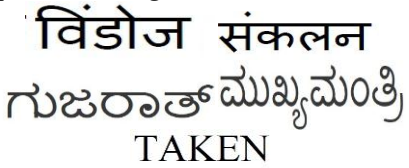


Image Preprocessing: Image pre-processing is the name for operations on images at the lowest level of abstraction whose aim is an improvement of the image data that suppress undesired distortions or enhances some image features important for further processing and analysis task. It does not increase image information content.

Image Preprocessing uses techniques such as Segmentation.

Segmentation: Image segmentation is process is used to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. As the premise of feature extraction and pattern recognition, image segmentation is one of the fundamental approaches of digital image processing. Image Segmentation is the process that is used to distinguish object of interest from background.

Our system first converts the original image into gray scale, and then uses edge as a factor to extract the region of interest from the given image. Edges are pixels where the intensity image function changes abruptly. Edge detectors are collection of local image pre-processing methods used to locate changes in the brightness function. Edges are the sign of lack of continuity, and ending. EDGE Find edges in intensity image. EDGE takes image as its input, and returns a binary image of the same size, with 1's where the function finds edges in image and 0's elsewhere.

V. RESULTS AND DISCUSSIONS

Proposed methods implemented in MATLAB. Since the standard dataset of Indian scripts is currently not available for the experimentation separate datasets comprising Kannada, English and Hindi are created from the internet news paper and text books. Few separate data sets are used for training and others for testing. The KNN and PNN method is tested with different number of Kannada, English and Hindi word. The results are formulated as follows.

Kannada

KNN accuracy

TP/(TP+TN) = 8/10 = 80%

PNN accuracy

TP/(TP+TN) = 10/10 =100 %

IMAGE NO	Actual Type	KNN Based	PNN Based
1	Kannada	Kannada	Kannada
2	Kannada	Kannada	Kannada
3	Kannada	Kannada	Kannada
4	Kannada	Kannada	Kannada
5	Kannada	Kannada	Kannada
6	Kannada	Kannada	Kannada
7	Kannada	Hindi	Kannada
8	Kannada	Kannada	Kannada
9	Kannada	Kannada	Kannada
10	Kannada	Hindi	Kannada

IMAGE NO	Actual Type	KNN Based	PNN Based
1	Hindi	Hindi	Hindi
2	Hindi	Hindi	Hindi
3	Hindi	Hindi	Hindi
4	Hindi	Hindi	Hindi
5	Hindi	Hindi	Hindi
6	Hindi	Hindi	Hindi
7	Hindi	Hindi	Hindi
8	Hindi	Hindi	Hindi
9	Hindi	Hindi	Hindi
10	Hindi	Hindi	Hindi
11	Hindi	Hindi	Hindi
12	Hindi	Hindi	Hindi
13	Hindi	Hindi	Hindi
14	Hindi	English	Hindi
15	Hindi	Kannada	Hindi

Hindi

KNN Accuracy

TP/(TP+TN) = 13/15 =0.86%

PNN Accuracy

/(TP+TN) = 15/15 = 1%

DOI 10.17148/IJARCCE.2015.4605

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 6, June 2015

IMAGE NO	Actual Type	KNN Based	PNN Based
1	English	English	English
2	English	English	English
3	English	English	English
4	English	English	English
5	English	English	English
6	English	English	English
7	English	English	English
8	English	English	English
9	English	English	English
10	English	English	English
11	English	English	English
12	English	English	English
13	English	English	English
14	English	Hindi	English
15	English	Hindi	English

English

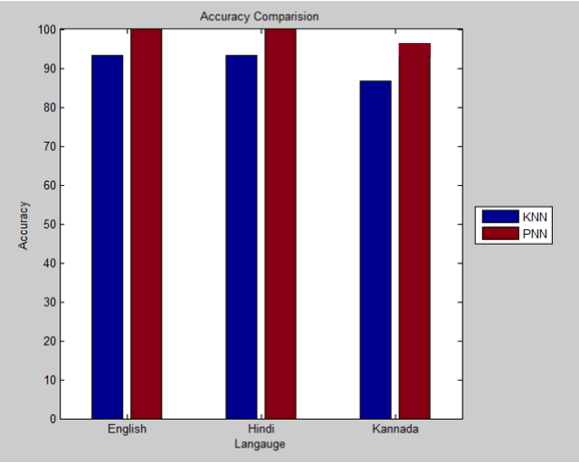
KNN Accuracy

TP/(TP+TN) = 13/15 =86%

PNN Accuracy

TP/(TP+TN) = 15/15 = 100%

Accuracy Analysis



VI. CONCLUSION AND FUTURE ENHANCEMENTS

The proposed KNN approach could successfully identify the three types of script words (Kannada, English and Hindi) with an average accuracy of more than 90%. It is based on the features from F1 to F8 extracted from the given text words.

The PNN based classifier could successfully identify the three script words (Kannada, English and Hindi) with an average accuracy of more than 95%. It is based on the average values of each feature and it is trained with more than 500 words of each script.

Hence the PNN based classifier is better than the KNN based classifier.

Future Work

The scheme can be extended to multiple scales to handle scripts printed at a different resolution. The proposed scheme can be used for other language scripts as well with minimal modification.

REFERENCES

[1] T.N.Tan, "Rotation Invariant Texture Features and their use in Automatic Script Identification", IEEE Trans. Pattern Analysis and Machine Intelligence, 20(7), 751- 756, (1998).

[2] 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.

[3] M.C.Padma, P.Nagabhushan, "Horizontal and Vertical linear edge features as useful clues in the discrimination of multilingual (Kannada, Hindi and English) machine printed documents", Proc. National Workshop on Computer Vision, Graphics and Image Processing (WVGIP), Madhurai, 204-209, (2002).

[4] G.S. Peake, T.N.Tan, "Script and Language Identification from Document Images", Proc. Eighth British Mach. Vision Conference., 2, 230-233, (1997).

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

[6] S.Basvaraj Patil, N.V.Subba Reddy, "Character script class identification system using probabilistic neural network for multiscript multi lingual document processing", Proc. National Conference on Document Analysis and Recognition, Mandya, Karnataka, 1-8, (2001).

[7] U.Pal B.B.Choudhuri, "Automatic Separation of Words in Multi Lingual multi Script Indian Documents", Proc. 4th *International Conference on Document Analysis and Recognition*, 576-579, (1997).

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

[9] U.Pal, S.Sinha, B.B.Choudhuri, "Word-wise script identification from a document containing English, Devanagari and Telugu text", *Proc. 2nd National Conference on Document Analysis and Recognition*, Karnataka, India, 213-220, (2003).

[10] P.Nagabhushan, S.A.Angadi, B.S.Anami, "A Fuzzy Statistical Approachto Kannada Vowel Recognition based on Invariant Moments", *proc. 2nd National Conference, NCDAR*, Mandya, 275-285, (2003).

[11] M.C.Padma, P.Nagabhushan, "Identification and separation of text words of Kannada, Hindi and English languages through discriminating features", Proc. 2nd National Conference on Document Analysis and Recognition, Mandya, Karnataka, 252-260, (2003).

[12] U.Pal, S.Sinha, B.B.Choudhuri, "Multi-Script Line Identification from Indian Documents", *Proc. 7th International Conference on*

[13] M Swamy Das , C R K Reddy , D Sandhya Rani , A Govardhan , "Script identification from Multilingual Telugu, Hindi and English Text Documents" , International Journal of Wisdom Based Computing.