Design and Development of a Script Recognition Tool for Indian Document Images

Sharmila S.\(^1\), Abirami S.\(^2\), Murukappan S.\(^3\), Bhaskaran R.\(^4\)

\(^1\)Dept. of Information science & Technology, Anna University, India.
sharmiatindia@gmail.com
\(^2\)Dept. of Information science & Technology, Anna University, India.
abirami_mr@yahoo.com
\(^3\)2Dept. of Computer Science, Annamalai University, India.
\(^4\)Dept. of Information science & Technology, Anna University, India.

Abstract— Identification of scripts from multi-script document is one of the important steps in the design of an OCR system for successful analysis and recognition. Most optical character recognition (OCR) systems can recognize at most a few scripts. But for large archives of document images that contain different scripts, there must be some way to automatically categorize these documents before applying the proper OCR on them. Much work has already been reported in this area. In the Indian context, though some results have been reported, the task is still at its infancy. This paper presents a research in the identification of Tamil, English, Hindi, Malayalam, Kannada and Telugu scripts at word level irrespective of their font faces and sizes. The proposed technique performs the nine zones segmented over the characters based on their shape, density and transition features. Then script is determined by using Rule based classifiers containing set of classification rules which are raised from the zones. Results from experiments, simulations, and human vision encounter that the proposed technique identifies scripts with minimal pre-processing and high accuracy. It can also be extended for other scripts. Since this system can act as a plug-in, this can be embedded with OCR prior to the recognition stage.

Keywords— Multi-lingual document, Script Identification, Rule Based Classification, Optical Character Recognition (OCR).

I. INTRODUCTION

With recent emergence and widespread application of computers and multimedia technologies, there is an increasing demand to create a paperless environment. Therefore all printed documents are converted into digital images. In the early 90s, several document analysis systems have appeared that are able to handle single language documents. Identification of script is relatively little attention in that document analysis field because one can normally deduce a document’s script from its country of origin, or by examining the document or it can be performed manually based on people’s experience.

But in a multi-lingual country like India (India has 18 regional languages derived from 12 different scripts), documents like bus reservation forms, passport application forms, examination question papers, bank-receipt, language translation books and money-order forms may contain text words in more than one language forms. For such an environment, multi-lingual OCR system is needed to read the multilingual documents.

Though most existing OCR systems are equipped with multiple OCR engines, manual routing is still required to switch incoming document images to the proper OCR engine for large archives of document images. But with the knowledge of the underlying script and language, incoming document images can be switched automatically and this significantly reduces the human involvement. So, there must be some way to automatically categorize these documents before applying the proper OCR on them.

To make a multi-lingual OCR system successful, it is necessary to separate portions of different language regions of the document before feeding to individual OCR systems.

The amount of multimedia data captured and stored is increasing rapidly with the advances in computer technology. Such data creates multi-lingual documents which are stored in typical large databases [18], [19], [21].

In such environment the large volume of data and variety of scripts makes such manual identification unworkable [18], [19]. In such cases the ability to automatically determine the script, and further, the language of a document, would reduce the time and cost of document handling. So the development of script identification from Multi-lingual document image systems has become an important task.

Script identification aims to determine the underlying script of document text, either in an image or a character-coded format [20]. This is fundamental issues in document analysis. The capability of recognizing multi-lingual document is both novel and useful, with such capability, many potential applications can be supported including multi-lingual access to patent, business and regulatory information, document sorting in support of character recognition, translation and keyword finding in document images [18]. Dealing with multi-lingual document raises many challenges including script identification, language determination, text reading direction and differing character sets. Most of the document images used in India is bilingual. This has motivated us to develop a method for automatic script identification of text words from multilingual documents images.

In the proposed technique, Document images are binarized, gray scale images are converted into black and white images. Sentence, words and characters are segmented from the binarized images. Features of characters such as its shape, density and frequency are extracted through zone segmentation and the extracted ones get transformed into digital values. Language is then determined according to the classification imported by the formation of the rules for digital values.

Since we intend to develop a script recognizer to discriminate between Tamil, English, Hindi, Malayalam,
Kannada and Telugu scripts, an overview about these scripts and its character set has been explained.

The remainder of this paper is organized as follows: Related Script identification work has been reviewed in Section II. Proposed Script identification technique is discussed in Section III Section IV and Section V. Experimental Results are discussed in Section VI. Performance analysis is discussed in Section VII and concluding remarks has been given in last section.

II. RELATED WORK

In this section, There are several approaches that can be used for determining the script/ language of printed documents and they can be typically classified into four categories: (a) connected components analysis (local features based analysis), (b) characters, words and text lines analysis, (c) text block analysis (global features based analysis), (d) hybrid information of connected components, text lines etc. In most of the text block and word level script separation work reported in (Dhanya et al 2002, Peake and Tan 1997, Pati et al 2004, Dhandra et al 2007 and Tan 1998) directional energy features of Gabor filters has been used. The discrimination of the scripts at text line and word level based on shape, conventional, strokes and water reservoir features can be found in (Padma and Nagabhushan 2003, Pal et al 2003). The connected components, clusters and projection profile features are used in (Dhanya et al 2002, Hochberg et al 1995, Dhandra et al 2007, Spitz 1997 and Wood et al 1995) for scripts separation. The hybrid information of connected components, text lines method is used in (Lu and Tan 2008).

Spitz 1994 Script can be identified with the help of a method which is used for distinguishing between Asian and European languages by examining the upward concavities of connected components.

Wood et al 1995 Identifies scripts with particular reference to the effect on the horizontal and vertical projections of the document image. This approach based on detecting the peaks in the horizontal projection profile (The black-count horizontal projection profile is a one-dimensional integer-valued function f(y), where the value off is the number of black pixels in row y).

Hochberg et al 1995 proposed a token based approach which first develops a set of representative symbols (templates) for each script by clustering textual symbols from a set of training documents and representing each cluster by its centroid. To identify a new document's script, the system compares a subset of symbols from the document to each script's templates, screening out rare or unreliable templates, and choosing the script whose templates provide the best match.

Spitz 1997, the script is classified as being either Han- or Latin based. Language determination in Han images (Chinese, Japanese, and Korean) is based on optical density distribution. Language determination in Latin images is based on the spatial relationships of features related to the upward concavities in character structures.

Anoop et al 2004, proposes a method to classify words and lines in an online handwritten document into one of the six major scripts.

Pal and Chaudhuri 1999 is a Combination of different features like: shape based features, statistical features and some features obtained from a concept of water overflow from the reservoir. The concept may be explained by the analogy of water overflow from a reservoir. If we pour water from the top, water will be stored in the concavity of the character, which is imagined as a reservoir. If we pour sufficient water, the water will flow out of the reservoir.

Dhanya et al 2002, proposed two different approaches. In the first method, words are divided into three distinct spatial zones. The spatial spread of a word in upper and lower zones, together with the character density, is used to identify the script. The second technique analyses the directional energy distribution of a word using Gabor filters with suitable frequencies and orientations.

Dhandra et al 2006, developed includes a feature extractor and a classifier. The feature extractor consists of two stages. In the first stage, the morphological erosion and opening by reconstruction is carried out on a document image in horizontal, vertical, right and left diagonal directions using the line structuring element. The length of the structuring element is fixed, based on the average height of all the connected components of an image. In the next stage, average pixel distribution is found in these resulting images. A nearest neighbour analysis is used to classify the new documents.

Dhandra et al 2006 is a continuation of Dhandra et al 2006 demonstrates the feasibility of morphological reconstruction approach for script identification at word level. Dhandra et al 2007 is used to identify scripts in bilingual document page may contain text words in regional language and numerals in English. it examine the word level in three bilingual documents, based on the observation that every text has the distinct visual appearance.

Lu and Tan 2008, Document images are vectorized by using vertical component cuts and character extremum points, which are both tolerant to the variation in text fonts and styles, noise, and various types of document degradation. For each script or language under study, a script or language template is first constructed using k-means clustering algorithm through a training process. Scripts and languages of document images are then determined according to the distances between converted document vectors and the pre-constructed script and language templates.

The work presented in this paper differs from the previous in several distinct ways and it has so many advantages compared to already existing works. First this work focuses on the study of two scripts Tamil and English. It works across document images with different constraints. It uses spatial approach, which can also work for low quality images. Design is not complex as in texture-based approach. Later, the proposed technique can be easily extended to handle document images of newly introduced scripts.

III. SCRIPT IDENTIFICATION ARCHITECTURE

Classification of script of word images in multilingual documents (Tamil, English, Hindi, Malayalam, Kannada and Telugu) has been reported in this paper. The proposed system identifies scripts by using the first character of each word in the multilingual document image. A rule...
based Classifier is used to classify the script at character level. Initially, characters of the word images are segmented. A rule-based classifier is then used to classify the language of a character from its vectors. Figure 1 depicts the architecture of script identification.

A. Tamil Script
Tamil is a South Indian language spoken widely in TamilNadu, India. Tamil has the longest unbroken literary tradition amongst the Dravidian languages. Tamil is inherited from Brahmi script. The Tamil script has 12 vowels, 18 consonants, 1 special character and 216 consonant based vowels. Hence a set of 247 symbols exists in the Tamil script.

B. Roman Script
Roman script has 26 each of upper and lower case characters. In addition, there are some special symbols and numerals. While the capital letters of the Roman script occupy the middle and the upper zones, most of the lower case characters have a spatial spread that covers only the middle zone or the middle and the lower zones. The structure of the Roman alphabet contains more vertical and slant strokes and less horizontal transition in the middle zone compared to Tamil.

C. Hindi Script
In Hindi (Devanagari) language, many characters have a horizontal line at the upper part. This line is called sirorekha in Devanagari. However, we shall call it as head-line. It could be seen that, when two or more characters sit side by side to form a word, the character head-line segments mostly join one another in a word resulting in only one component within each text word and generates one continuous head-line for each text word. Since the characters are connected through their head-line portions, a Hindi word appears as a single component and hence it cannot be segmented further into blocks, which could be used to recognize this script.

D. Malayalam Script
The Dravidian Language Malayalam, which is one of the 22 official languages of India, is being used by around 36 million people, predominantly in Kerala, the southern state of India. Malayalam first appeared in writing in the vazhappalli inscription which dates from about 830 AD. In the early thirteenth century the Malayalam script began to develop from a script known as vattezhuthu or round writing, a descendant of the Brahmi script. Malayalam language script consists of 53 letters including 16 vowels and 37 consonants. The earlier style of writing is now substituted with a new style from 1981. This new script reduces the different letters for typeset from 900 to fewer than 90.

E. Kannada Script
Kannada, the State Language of Karnataka is described as ‘sirigannada’ and is one of the oldest Dravidian languages and is spoken in its various dialects by roughly 40 million people. Kannada has now received the Classical Language status in India. The language has forty-nine phonemic letters, divided into three groups: Swaragalu means vowels with thirteen letters; Yogavaahakagalu which are not vowel and consonant, with two letters and Vyanjanagalu pertaining to consonants with thirty four letters. Figure 6 shows in characters Kannada.
Telugu, a south central language is the official and administrative language of Andhra Pradesh and has been recently included in the classical language list of India. A highly Sanskrit language, it is being spoken by around 80 million people. It is the third most spoken language in India after Hindi and Bengali. The British writer C. P. Brown, has in fact described Telugu as The Italian of the East as all native words in this language end with a vowel sound. The complete character set of Telugu script is shown in figure 7 which consists of 16 vowels and 36 consonants.
IV. PROPOSED ALGORITHM

In the proposed algorithm, input is assumed to be a grayscale image obtained by scanning the multilingual newspapers. The input document is assumed to contain text and thus free from graphics, figures, maps, and tables. Steps involved in the proposed algorithm have been explained below:

1. Pre-process the input document image
2. Perform binarisation using Otsu’s global thresholding method
3. Carry out the line wise and word wise segmentation using horizontal and vertical projection profiles.
4. Extract first connected component.
5. Perform document vectorization technique
6. Classify the new word image based on the Rule based classifier.

Then the scanned images pass through the two phases such as Training Phase and Testing Phase.

V. TRAINING PHASE OF SCRIPT IDENTIFICATION

Training Phase acquires multi-lingual document image corpus (mixture of Tamil, English and Hindi words) constituting various font faces and sizes by scanning the hard copy of magazines in order to adapt the variations. This phase includes pre-processing of document images, word and character image segmentation, Tetra Bit generation, Selection of Classification, Rule based classifier Design, Rule Extraction and Sub classification technique.

A. Pre-processing

Any document images often suffer from noise of different sizes together with those small-connected components. All documents are scanned using HP Scanner at 300 DPI, which usually yields a low noise and good quality document image. So the document images can be binarized directly.

But most of the document images are affected by salt and pepper noise as illustrated in the following Figure 8.

Noise spikes are normally significantly brighter or darker than their neighbouring pixels. Centre weighted median filter is used to keep the shape of the character stroke much better. The algorithm for pre-processing is as follows,

1. Input the document images which is affected by salt and pepper noise.
2. Use centre weighted median filter to remove noise
3. Select the mask (3*3)
4. Sort all pixels in particular order. Give weights for each pixel and give maximum weight for centre pixel.
5. Find the median value of all that pixels
6. Assign the median value of the group to the centre pixel.

Binarization uses Global Thresholding method which is used to convert gray scale images into binary images. The most common method is to select a proper threshold for the image and then convert all the intensity values above the threshold intensity to one intensity value representing either “black” or “white” value. All intensity values below a threshold are converted to one intensity level and intensities higher than this threshold are converted to the other chosen intensity.

B. Segmentation

After pre-processing, the noise free image is passed to the segmentation phase, where the image is decomposed into words. To segment the document image into several text lines, horizontal projection profile is used which is computed by a row-wise sum of black pixels. The position between two consecutive horizontal projections where the histogram height is least denotes one boundary line. Using these boundary lines, document image is segmented into several text lines. Similarly, to segment each text line into several text words, we use the valleys of the vertical projection of each text line obtained by computing the column-wise sum of black pixels. The position between two consecutive vertical projections where the histogram height is least denotes one boundary line. Using these boundary lines, every text line is segmented into several text words. Algorithm for Segmentation is as follows,

Input : Multilingual Document Image
Output: Extracted Characters

1. The binarized image is checked for inter line spaces.
2. Each line in the paragraph is identified through Horizontal Projection Profile
3. Once the line bounding has been over, with the help of vertical projection, word images have been identified
4. Characters are identified from the words using Inter character distance in words.
C. Formation of Tetra Zones

Once the characters are extracted from words, spaces on both the sides of the characters are trimmed to extract the exact black pixel density. Next, total character has been divided into nine zones and black tone & white tones of every zone have been grabbed. Applying two horizontal and two vertical bisections over the character has made tetra zones. The character image of *3*3 pixels is divided into M*N grids[37] as shown in figure 9.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Fig. 9 Zonal Division.

Two Horizontal projections are applied over a character. Two Vertical projections are applied over a character. where he is defined as the height of the character and wc represents the width of the character.

Horizontal projections over the characters comprising of ascender, middle and descender zone has been reported in earlier works. Vertical projection is required to analyze further, the spatial spread of every character and we resulted in a 3*3 grid. This is illustrated for a sample Tamil and English character in figure 10.

![Sample Image cut in Nine Zones.](Image)

D. Vector generation

Vectors for every zone have been generated based on its black pixel or white pixel density in that particular zone [37]. Generation of vectors for every zone has been represented in the following algorithm.

1. Repeat steps 2 to 4 for every zone of a character c.
2. Along the height h and width w zone n, traverse the zone n along x direction, (x=0…w) for every y. (y=0…h).
3. For every occurrence of black pixel along the traversal, increment the black pixel counter bpc by 1. This is represented in equation
4. For every occurrence of white pixel along the traversal, increment the white pixel counter wpc by 1.
5. Set the value of zone n to 1, if the ratio of wpc over the total number of black and white pixels in zone n exceeds the threshold. (1- represents the domination of white pixels over black pixels)
6. Set the value of zone n to 0, if the ratio of bpc over the total number of black and white pixels in zone n exceeds the threshold. (0- represents the domination of black pixels over white pixels).

E. Classifier Design

The task of discrimination can be carried out successfully by a supervised learner, which tries to predict the value of the function for any valid input object after seeing a number of training examples (i.e. pairs of input and target output). The proposed technique uses rule based classifier to discriminate different scripts.

First the total number of zones having less dominance of black pixels is identified from the generated document vectors. If number of zones having less dominance of black pixels=0, then horizontal transition is applied directly since the rule based classifier cannot differentiate some of Tamil and English words having same vectors. Else patterns are generated from vectors. Then classification decisions are made by using the following five algorithms

1. Rule Based Classifier
2. Horizontal Transition
3. Density Analysis in Upper Zone
4. Left and Right Profile Analysis
5. Tick Component Analysis

1) Rule Based Classifier:

Since decision trees are large and difficult to interpret, Rule-based classifiers are used which uses only IF-THEN rules. This classifier is designed based on the result of pattern analysis.

Based on the training set of characters, following rules have been framed to classify the vectors generated by zones as English, Tamil or Hindi.

1. Set Lang attribute to null. Lang =“”.
2. Receive the vectors for all the nine zones.(z1…z9)
   a) if  
      
      \[(z2∧z5)∨(z5∧z8)∨(z2∧z8)=1\]
      then Lang = “E”
   b) else if (z4 equals ‘1’) Lang = “E”
   c) else if ((z2∧z3) equals ‘1’) Lang = “E”
   d) else if (z8 equals ‘1’)  then Lang =”T”
   e) else if (z4 equals ‘1’) then Lang = “E”
   f) else if (z3 equals ‘1’)  then Lang=”T”
   g) else if ((z4 and z6) equals ‘1’) Lang = “E”
   h) else if ((z1 and z2) equals ‘1’) Lang = “T”
   i) else if (z4 and z6 and z9) equals ‘1’ then Lang = “T”
   j) else if ((z3 and z5 and z9) equals ‘1’) Lang = “E”
   k) else if ( (z1 and z2) and z4 and z5 and z7 and z8) equals ‘1’) Lang=”N”
   l) else Horizontal Transition is applied.
3. Traverse the following rules (a..e) with the nine vectors to classify the language of a character:
4. Attribute Lang represents the classified language. 
   (E – English, T- Tamil).

2) Horizontal Transition Analysis:

This classifier is used to differentiate most of the Tamil and English characters except few such as உ , ப , etc...because of distribution in all zones. Since vectors are same for these types of characters, sub classifier is required [37].
Sub classifier which is used to classify some of Tamil, English scripts which are distributed in all nine zones or having same density structures is called as Horizontal Transition. Horizontal Transition rate detects the horizontal disposition rate by accounting every black to white and white to black dispositions over the middle zone. It was also observed that from the training samples that the transition rate of middle zones is used to classify both Tamil, English scripts. For example the transition rate of English characters is less than the transition rate of Tamil characters. The Algorithm for Horizontal Transition Analysis,

Consider the middle zone of the character
1. Initially the no of horizontal transition h=0
2. Traverse each and every pixel in that row and count number of every black to white and white to black dispositions and increment h value for each transition
3. Repeat step (2) for all rows in that zone.

Make decision based on the h value. If h>2, Then the script is Tamil else it is English. Rule based classifier and Horizontal transition methods properly identify Tamil and English scripts. But we need some method to differentiate some of Tamil and English scripts from Hindi scripts. Hence another sub classifier called Head line analysis is used to classify the Hindi scripts.

3) Head Line Analysis:

This is based on the Headline and vertical line structure of Hindi words. Headline would be available for most of the Hindi words occupying the top three zones and this act as a discriminating factor for Hindi and Tamil script. In order to discriminate the English characters who have headlines from Hindi, a vertical line associated with the headline in the centroid area or right zone have been analyzed to differentiate Hindi script. The Algorithm for Head Line Analysis,

1. Consider the upper three zones of the character
2. i) Initially the no of pixels in a row r=0 and t is the total width of that character
ii) Traverse each and every pixel in that row and count number of every black pixels and increment r value
iii) Repeat step (ii) for all rows in that zone and find the r which has maximum number of black pixels in that zone.
3. Script can be identified by comparing the pixel position of their concave shape. If the character has concave shape then it is Malayalam else it is Tamil.

We need one more technique to identify Telugu script from Kannada script. In this case Telugu script has a tick feature whereas that feature is missing in Kannada script and this is used in this paper for discrimination.

5) Tick-component:

The observation of the characters of Telugu script motivated us to use the tick shaped components as a feature. A component is said to have the shape of the tick like structure if the pixel values of the components are in the sequence (i, j), (i+1, j+1), (i+2, j+2), ... , (i+m1, j+n1), (i+m1-1, j+n1+1), (i+m1-2, j+n1+2), (i+m1-3, j+n1+3), ... , (i+m1-m2, j+n1+n2), where m2<i+m1 and n2>n1. The shape of the tick-like structure is shown in Figure 11. The component having the shape of the tick-like structure (\( \forall \)) is named as the feature ‘tick component’. Such tick-components extracted from the top portion of Telugu script are shown in above Figure 12. The Algorithm for Tick-Component Analysis.

1. Start
2. Consider the top 3 zones of the character initially bpc = 0
3. i) Traverse each and every pixel in zones check the black pixel peak positions. From the peak position increase j value and decrease i value and check for the pixel value. If it is black then continue. Increase bpc by 1
   ii) Repeat step (ii) till reach the white pixel.
   iii) Now decrease the value of i, j and check for the pixel value. If it is black continue and decrease the i, j value up to bpc/3.
4. If all where black pixel and tick feature is exist then the script is Telugu
5. else then the script is Kannada

F. Testing Phase of Script Recognition

In this paper, scripts are reported for word images using their initial character. They are classified sharply based on the density of black pixels in every zone. Accuracy as well as Time factors are better in this approach.

Because of the differences in character densities i.e. spatial spread per unit area/ per zone, features of the characters of both the language gets varied.

Like previous methods, this does not concentrate the spatial spread as ascender, middle and descender zones. This reveals the identity of different characters through the accumulation of black pixel in every zone as either dense or sparse.

When a testing sample is provided to the system, words and characters are segmented, tetra zones are formed and in turn features of every zone gets extracted. Features are transformed into nine bits as explained above. Nine bits goes through the above rule based classifier to classify the language.

For instance, English word represented in figure 13 is properly classified as English. Initially, the first character gets segmented and the bits generated for that character is 000101101. Classifier clearly discriminates this as Tamil.

**Tamil**

Fig. 13 Sample English Word.

For instance, Tamil word represented in figure 14 is properly classified as Tamil. Initially, the first character gets segmented and the bits generated for that character is 000000000. Here the classification decision is made as “HT”. Horizontal transition procedure clearly discriminates this as Tamil.

VI. EXPERIMENTAL RESULTS

This system has been implemented using Java Advanced Imaging. Inputs were obtained from various magazines, newspapers and other such documents containing variable font sizes. The training and test patterns have been taken, consisting of variable English and Tamil words.

The following figures shows sample outputs of script identification.
Fig. 17 Identified Tamil Script

Fig. 18 Identified English Script

Fig. 19 Identified Telugu Script

Fig. 20 Identified Malayalam Script

Fig. 21 Identified Kannada Script
VII. PERFORMANCE ANALYSIS

Quantitative performance of the classification scheme is computed over a sample document image collection of more than 250 document images. Considerable number of words is used for testing in document images. During the classification of words, priority is given to Multi-lingual document (Tamil, English, Malayalam, Kannada, Telugu and Hindi) words mixed in the same document image. The proposed system performs well for document images even from quality of 200-300 dpi when comparing to other systems. This is not constrained to a single image type and could read images of types BMP, JPEG and TIFF. Further, our system well normalized to act upon different font sizes.

Table I shows Precision Measure of Scripts.

<table>
<thead>
<tr>
<th>Script</th>
<th>No. of words</th>
<th>T</th>
<th>E</th>
<th>H</th>
<th>M</th>
<th>TE</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tamil</td>
<td>125</td>
<td>118</td>
<td>1</td>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>English</td>
<td>50</td>
<td>1</td>
<td>49</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Hindi</td>
<td>23</td>
<td>2</td>
<td>0</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Malayalam</td>
<td>20</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>18</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Telugu</td>
<td>15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>13</td>
<td>1</td>
</tr>
<tr>
<td>Kannada</td>
<td>21</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>16</td>
</tr>
</tbody>
</table>

VIII. CONCLUSION AND FUTURE WORKS

This paper reports an identification method that detects script and languages at word level by using the first character of a word. Also this technique is normalized to a set of font faces and sizes. Document images are binarized, words and characters are segmented from the images. Features of characters such as its shape, density and frequency are extracted through zone segmentation and the extracted ones get transformed into digital values. Language is then determined according to the classification imported by the formation of the rules for digital values. Results from experiments, simulations, and human vision encounter that the proposed technique is promising and easy to extend for other languages. Currently this work focuses only for Tamil, English, Hindi, Malayalam, Telugu and Kannada.

In future, this process will be extended to all Indian Scripts. As the system can act as a plug-in, this can be embedded with OCR prior to the recognition stage.

REFERENCES


