Segmentation of Handwritten Tamil Character from Palm Script using Histogram Approach

Abstract

Tamil Palm script character segmentation is one of the difficult phases in machine recognition. Tamil is the most popular script in India. Tamil palm script consist of vowels, consonants and various modifiers. Individual letter decides the accuracy of character recognition technique hence proper segmentation is needed. This paper presents an image segmentation of Tamil handwriting from palm leaf manuscripts. The process is includes of three steps: background elimination to separate text by Otsu's algorithm, line segmentation and character segmentation. A simple histogram based approach to segment Tamil palm script character is proposed in this paper. Various challenges in segmentation of Tamil script are also discussed.

I. INTRODUCTION

Tamil character segmentation from palm manuscript is an important task for Recognition System. Segmentation is the method of splitting the image into text lines, words and then into characters which are particularly useful for classification. Segmenting character from palm manuscript is extremely challenging, while the characters structure and content differ significantly. The correctness of the OCR system depends on the segmentation. If the characters are segmented correctly the recognition system gives best results. Regions or objects are divided from an image in segmentation phase. Mainly segmentation, tries to extract essential component of the script, which are certainly characters. This is desirable because the classifier recognizes these characters only [1]. Segmentation phase is also critical in contributing to this inaccuracy due to touching characters, which the classifier
cannot correctly recognize. Even in good quality documents, some nearby font touch each other due to improper scanning resolution [2]. The organization of the paper is as follows feature of Tamil Script are described in section 2, section 3 illustrate about the palm manuscript. Section 4 describes the framework of proposed system which includes preprocessing techniques and segmentation algorithms. Section 5 presents experimental results and Section 6 describes the conclusion and future works. Section 7 provides the references.

II. FEATURES OF TAMIL SCRIPT

Tamil is a traditional language which is broadly spoken in most part of the south India. There are 12 vowels, 18 consonants and one special character present in unique Tamil Script. Each vowel combined by pure consonant to create a value of 216 consonant-vowel (CV) combinations. These add up to a total of 247 Tamil characters. Tamil Language alphabetic system is derived from the prehistoric Brahmi script which serves as a base for most of the Indian languages. The vowels and consonants of Tamil alphabet set are given in the table 1.1:

<table>
<thead>
<tr>
<th>Table 1.1 Modern Tamil Character Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vowels</td>
</tr>
<tr>
<td>Constants</td>
</tr>
<tr>
<td>Grantha</td>
</tr>
<tr>
<td>Aytam</td>
</tr>
</tbody>
</table>

A well developed handwritten character Recognition system is still not available for Tamil language. The main reasons for this are:
1. Tamil Language has a very large character set
2. Letter structure is very complex
3. Due to complex letter structure, writing styles of people vary significantly
4. There is no Tamil character database that exists for testing purposes in the public domain.

III. PALM LEAF DOCUMENT IMAGE

Palm leaf manuscript is one of the writing medium in India especially in Southern India. However palm leaf writing was accomplished since the prehistoric times its precise origin is still unclear. Agrawal stated that, “It is hard to say exactly when the palm-leaf begins to be used for writing. There is no scope of palm-leaf manuscripts in India before the 10th century. However, the palm-leaf was certainly in use before than this from the time when it’s mentioned as a writing material in several mythical works and its picture representation
can be seen in several sculptures and monuments [3]. Palm leaf manuscripts are special from other documents. The major troubles with them investigation are deprived quality (fragility and deterioration over age), Poor contrast, holes, ghosting noise, spots on the media, constricted spaced lines with overlapping and touching components, abnormal varying shapes, and different styles of characters, which depend on the writer and even the era of writing.

In general, Palm leaf images changes to distorted images due to the presence of noise, low or high contrast both in the edge area and image area. Removal of distortion, edge or boundary improvement, self regulation edge detection, variance adjustment is included in preprocessing stage and characters are finally segmented. Various enhancement techniques are necessary for improving the difference stretch in palm leaf images [4] because much kind of noises damages the quality of palm manuscripts. Different techniques for sequential manuscript image enhancement are determined by the purpose of improving individual readability while maintaining the original look and feel of the document. The current printed document, also palm leaf documents images are typically crucial to analyses.

![Diagram](image.png)

**Figure 1: Tamil Character Segmentation from Palm Manuscripts**

**IV. FRAMEWORK FOR PROPOSED SYSTEM**

The Tamil character segmentation from palm manuscript process is shown above in the figure 4.1. We performed our experiments over a database generated by Collecting 15 Tamil palm manuscript samples and scanned by optical scanner. The image files are stored in jpg format.

**4.1 IMAGE PREPROCESSING**

Data are collected every time for recognition and optically scanned image. Text is converting into digital form by using a flatbed scanner having resolution between 100 and 600 dpi and stored. These picture elements such as pixels may have values of 0 or 1 for binary images, 0– 255 for gray-scale images and three channels of 0–255 color i.e. RGB values for color images. This collected raw data should be further analyzed to get useful information. Such processing includes the following:

**4.1.1 RGB TO GRAY CONVERSION**

The optically scanned palm script is stored as a JPEG image. The image may save as other formats such as BMP; TIFF etc. are also used for recognition. All these images are in
RGB format are converted into gray scale as shown below in the figure 2, then the RGB values for each pixel and make as output a distinct value reflecting the intensity of that pixel. The averages of each channel are contributed as \((R+B+C)/3\). The value of a pixel lies under 0 to 1 or under 0 to 255 depending upon its class.

![Gray Scale Image](image2)

**Figure 2: Gray Scale Image**

![Binary Image](image3)

**Figure 3: Binary image**

### 4.1.2 Thresholding Using Otsu’s Algorithm

Thresholding is a main technique in image segmentation applications. Otsu technique is kind of global thresholding in which it depend only gray value of the image. The basic suggestion of thresholding is to decide on a best gray-level threshold value for separating objects of interest in an image from the background based on their gray-level distribution. The gray-level histogram of an image is generally considered as well-organized tools for improvement of thresholding algorithms. By turning all pixels below some threshold to zero and all pixels about that threshold to one, thresholding creates binary image which is shown in the above figure4.3. If \(g(x, y)\) is a threshold account of \(f(x, y)\) at various global thresholds \(T\), it can be defined as [4]

\[
g(x, y) = 1 \text{ if } f(x, y) \geq T
\]

\[= 0 \text{ otherwise}
\]

### 4.1.2 Skew Detection and Correction

Handwritten Palm manuscript may initially be skewed or skewness may introduce in palm script scanning process. This effect is accidental in many real cases, and it should be eliminated because it effectively decreases the accuracy of the consecutive processes such as segmentation. Skewness is corrected by using projection profile Analysis[5]. A twofold image into one-dimensional array (projection profile) conversion is known as projection. Each line in projection profile has a value that produce a number of black pixels in the corresponding row of the image and lines on document are represented as horizontal histogram profile. For those images contain zero skewed angle, the horizontal projection profile has channel which is equal with the space between the lines. And also the
maximum peak height which is equal to text lines height present in document images. Therefore, this method calculates the difference in projection profile at a number of dissimilar angles is equal to angle that have the most difference.

4.2 SEGMENTATION

After the image is preprocessed using methods discussed in section 4.1, histogram projections in y direction are obtained in order to perform Line segmentation and then x histogram projections for character segmentation. The process of segmentation mainly follows the following pattern:
1) Identify the text lines in the page.
2) Finally identify individual character in each line.

4.2.1 LINE SEGMENTATION

Using global horizontal projection profile each line in a given image are identified [4]. It is used to calculate sum of all white pixels on each row and construct subsequent histogram. The process of line segmentation is as follows [6]:

Figure 4: Histogram of the image

Step 1: Horizontally scan the preprocessed image and find the number of white pixels in each row.
Step 2: Plot the histogram in y direction for the white pixel count for the image.
Step 3: Scan the histogram projection to find first white pixel count with zero and remember that y coordinate as y1.
Step 4: To find white pixel counts to be non zero since the characters would have started continuous histogram projection scanning is needed.
Step 5: Finally we get the first white pixel count as zero and remember that y coordinate as y2.
Step 6: Scan the image from y1 to y2 rows for the segmented line.
Step 7: Clear y1 and y2.
Step 8: Repeat the above steps till the end of the histogram.

4.2.2 CHARACTER SEGMENTATION

To segment each character present in the database must follow the below process:
Step 1: Initially Tamil palm manuscript is scanned by using optical scanner.
Step 2: Convert Palm script scanned image to gray image by using Matlab function.
Step 3: Gray image converted to binary image by using Otsu’s thresholding algorithm.
Step 4: To normalize image against thickness of the character, thinning operation is done by using Matlab bwmorph function.
Step 5: Calculate the white pixel in each column.
Step 6: Find the region which contain single white pixel.
Step 7: Substitute all such columns by 1.
Step 8: Invert the image to make such columns as 0 and text characters will have original pixels.
Step 9: By using standard Matlab function plot the Bounding Box for characters.
Step 10: Extract and save the segmented character in separate file as shown in the figure 5.

Figure 5: Segmented Characters

5. EXPERIMENTAL RESULTS

In this research, segmentation scheme on 75 different script line used and it totally contains 2098 characters are found in script line. From the experiment, total number of lines recognized of 96% accuracy and boundary box algorithm bound the characters of 97% result and characters found correctly were 87% accuracy. Different script segmentation detail is shown in the following table 5.1. During character segmentation, the error occur because of Tamil Script include consonants are modified in various ways from left, right, up or down to form a meaningful letter. Independent upright lines in the words are well-known as separate symbol by the algorithm used. For perfect segmentation, all the modifiers must be segmented so that their detection can be properly done.

Table 5.1: Overall segmentation Recognized results

<table>
<thead>
<tr>
<th>Line segmentation</th>
<th>Lines in the script</th>
<th>Recognized line</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>75</td>
<td>72</td>
<td>96%</td>
<td></td>
</tr>
<tr>
<td>Character Segmentation</td>
<td>Characters in the script</td>
<td>Recognized character</td>
<td>Accuracy</td>
</tr>
<tr>
<td>2098</td>
<td>1817</td>
<td>87%</td>
<td></td>
</tr>
</tbody>
</table>

VI. CONCLUSION AND FUTURE WORK

In this segmentation process the global horizontal projection algorithm is tested with several Tamil palm manuscript images and produce results as 87%. This method provides robust results, but it could not exactly segment the overlapped lines. A heuristic techniques needs to be consideration of in case of overlapping lines and characters to recover the loss text. In future, the progress will aim at improve the recognition accuracy of the proposed
system by using more data sets and provide solution to the overlapping line and character problem present in the ancient Tamil manuscripts.

VII. REFERENCES


TO CITE THIS PAPER