Research Report on the Preprocessing Engine of the Optical Character Recognition System for Sinhala Scripts

Dulip Herath and Nishantha Medagoda
Language Technology Research Laboratory
University of Colombo School of Computing,
Sri Lanka.
dherath@webmail.cmb.ac.lk, nmedagoda@webmail.cmb.ac.lk

Abstract

Preprocessing is an essential step in any image processing task. This paper describes the issues that have to be addressed when a Preprocessing engine is developed for an OCR System for Sinhala characters based on a template matching technique. A detail description of the procedures implemented is also given along with their results.

1. Introduction

1.1. Background

Most of the information that is available in the world is in printed medium. This has hindered storing, exchanging and processing of this information electronically. Converting them from printed to electronic medium is time consuming, laborious and expensive as well. These factors have motivated people to develop automated systems to perform this task. Optical Character Recognition (OCR) is an important technique that is very popular among research and technical communities. As a result of these research and development activities several OCR applications are now available in the market.

Most of the existing OCR systems are for Latin character based languages. Though research has been going on for several decades, commercial grade OCR systems for Indic languages are very rare. Researches have been done for Hindi, Tamil, Telugu, Oriya and several other Indic languages. The most significant difference between Indic scripts and Latin scripts is the cursive property. Different techniques have to be applied in the process of automatically recognizing them. An OCR system for Sinhala scripts is being developed at the Language Technology Research Laboratory at UCSC based on the research work carried out recently.

As described in the literature typical OCR system should contain some essential modules, namely, Data Acquisition, Preprocessing, Classification/Recognition, Postprocessing and Output. The aim of the present report is to describe the Preprocessing module of the OCR developed for Sinhala Scripts.

The recognition algorithm of the present OCR system will be based on template matching. In order to achieve higher recognition rate it essential to have an effective preprocessing engine. Following sections of this report describe the functionalities of the preprocessing module of the present OCR system.

1.2. Sinhala scripts

Sinhala scripts can be of two types namely, base characters and modified characters. This latter type is formed by combing vowel modifiers to base characters. Vowel modifiers stand for vowel sounds.

Base characters: අ, ක, ම, ද

Vowel modifiers: ◌ෙ, ◌ො, ◌ෝ, ◌ි , ◌ු, ◌ා, ◌්

Modified characters: මෙ, මේ, මෝ ,මි ,මු ,මූ ,මා ,මී, ක්‍රො, කු,රු

These scripts can be categorized into four groups according to their structure. Basically three (3) zones can be identified in a typical Sinhala character. The three zones are named Upper, Middle and Lower zones. The heights of the Upper and Lower zones are normally 25% of the line height and the Middle zone is of height 50% of the line height. Figure 1 shows these zones in the letter ප්‍රි (pri).

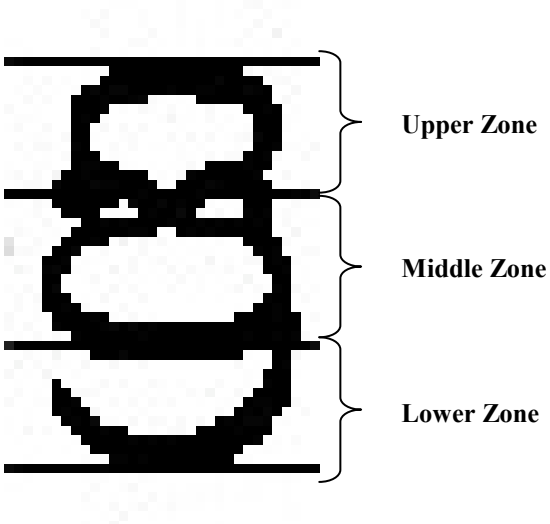


Figure 1: Three zones appear in a typical character

The four character groups are classified as shown in Table 1.

Table 1: Different character groups

Zone ID	Characters
0	ක න ත ම ව ට ය ස ප බ ණ
1	වි මි ටි සි යි ජි
2	වු මු සු කු
3	ප්‍රි ක්‍රි ශ්‍රී

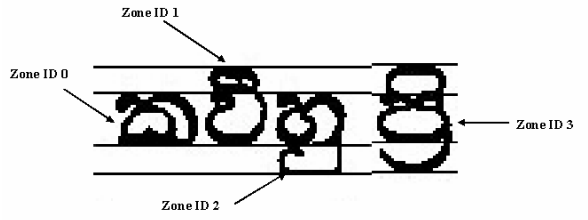


Figure 2: Characters with different Zone IDs

Figure 2 shows how characters can be classified according to the structure of them.

1.3 Template matching

As mentioned in the section 1.1 the present OCR system is based on a template matching technique. The templates are initially classified according to the Zone ID. When a character template is matched, the pixels belong only to the relevant zone are taken into consideration. That would improve the efficiency as well as the accuracy of the algorithm. In order to perform this task layout information of the particular image that has to be recognized. For example, height and width of the image, number of lines in the image, average line height of the given image.

The templates are of fixed height and the image that has to be recognized is compressed until its average line height becomes the size that of template height before the template matching. Since template matching techniques are very sensitive to noisy images system would give erroneous results if the source image consists of too much noise. These factors demand an effective preprocessing engine to make the OCR system more accurate and robust mainly through sound segmentation, normalization and noise removal.

2. Preprocessing Engine

2.1 Preprocessing stage

A typical OCR system consists of several modules connected in a sequential manner to simulate well defined stages in character recognition. Figure 3 shows the overall architecture and the place occupied by the preprocessing engine in the present system.

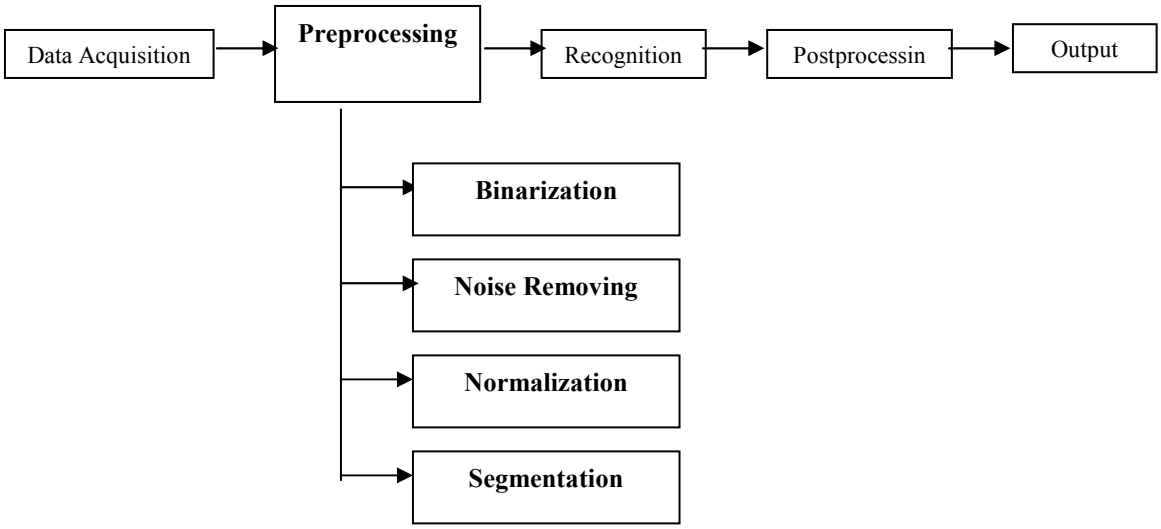


Figure 3: Overall architecture of the OCR

The preprocessing module receives an image captured at the Data Acquisition stage. The procedures implemented in the Preprocessing module extract layout information such as height and width of the image, horizontal and vertical projections, etc;, standard noise removal algorithms remove noise present in the image. Finally, the image is broken down into lines and they are further broken down into characters. The implementation of all the procedures used in the Preprocessing module has been done in ANSI C so that they can be run on any platform. A detail description on each of these sub modules will be given in sections to come.

2.2. Binarization stage

Normally pixel intensity values of an image are in the range of 0 to 255. At this stage pixel intensities are set either to 0 or to 255 by using a threshold determined dynamically. This will improve the recognition rate since that helps the algorithm to differentiate the background and the foreground by inspecting the pixel intensity. The threshold for Binarization is determined dynamically by inspecting the most frequent and least frequent pixel intensities.

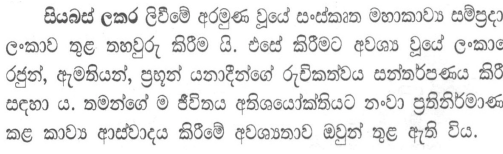

සියබස් ලකර ලිවීමේ අරමුණ වූයේ සංස්කෘත මහාකාව්‍ය සම්ප්‍රදාය ලංකාව තුළ තහවුරු කිරීම යි. එසේ කිරීමට අවශ්‍ය වූයේ ලංකාවේ රජුන්, ඇමතියන්, ප්‍රභූන් යනාදීන්ගේ රුචිකත්වය සන්තර්පණය කිරීම සඳහා ය. තමන්ගේ ම ජීවිතය අතිශයෝක්තියට නංවා ප්‍රතිනිර්මාණය කළ කාව්‍ය ආස්වාදය කිරීමේ අවශ්‍යතාව ඔවුන් තුළ ඇති විය.

Figure 4: Before binarization

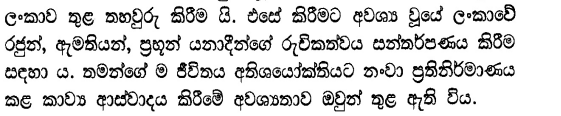

සියබස් ලකර ලිවීමේ අරමුණ වූයේ සංස්කෘත මහාකාව්‍ය සම්ප්‍රදාය ලංකාව තුළ තහවුරු කිරීම යි. එසේ කිරීමට අවශ්‍ය වූයේ ලංකාවේ රජුන්, ඇමතියන්, ප්‍රභූන් යනාදීන්ගේ රුචිකත්වය සන්තර්පණය කිරීම සඳහා ය. තමන්ගේ ම ජීවිතය අතිශයෝක්තියට නංවා ප්‍රතිනිර්මාණය කළ කාව්‍ය ආස්වාදය කිරීමේ අවශ්‍යතාව ඔවුන් තුළ ඇති විය.

Figure 5: After binarization

2.3. Noise removal stage

Noise that exists in images is one of the major obstacles in image processing tasks. The quality of an image degrades when noise is present in it. Noise can occur at image capturing, transmission and compression stages. Various standard algorithms are available for removing noise exist images.

The Noise removal is done using a Gaussian Filter which is one of the popular effective noise removal techniques. The optimum mask size and standard deviation are obtained by trying out several Gaussian masks of different sizes and also with different standard deviation values.

2.4 Layout information extraction stage

Having removed the noise existed in the image, several procedures extract layout information such as Horizontal Projection, Vertical Projection, Number of Lines, Number of Characters, etc.

2.4.1. Horizontal projection. In this procedure horizontal lines are scanned from left to right. As each line is scanned number of black pixels that were available in the line is recorded. It is possible to

identify the number of lines in a given documents as well as the line boundaries.

2.4.2. Vertical projection. In this procedure vertical lines are scanned from top to bottom. As each line is scanned number of black pixels that were available in the line is recorded. It is possible to identify the number of characters in a given line as well as the character boundaries.

2.4.3 Number of lines. Based on the information provided by the Horizontal Projection this procedure calculates the number of lines that exist in the image. When there is some noise that was added at the image capturing phase some lines are not properly segmented as shown in Figure 6.

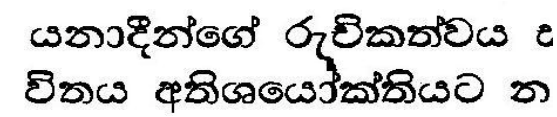


Figure 6: Incorrectly segmented lines

In cases like this two lines are identified as one line. This will lead to errors in calculating the number of lines and average line height of an image.

2.4.4. Number of characters. Based on the information provided by the Vertical Projection this procedure calculates the number of characters exist in a give line. When there is some noise that was added at the image capturing phase some characters are not properly segmented as shown in Figure 7.

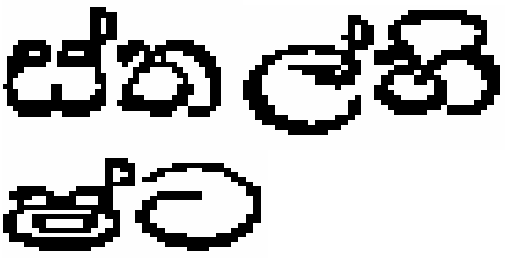


Figure 7: Incorrectly segmented characters

2.4.5. Line information. Based on the information provided by the Horizontal Projection and Line Information this procedure extracts the pixels where each line starts and ends.

2.4.6. Character information. Based on the information provided by the Horizontal Projection and Character Information this procedure extracts the pixels where each character starts and ends in a given line.

2.2.7. Average line height. Based on the information provided by Number of Lines and Line Information procedures, this procedure calculates the average line height.

2.4. Normalization stage

This is an important step in the preprocessing phase particularly in template matching. In the present OCR system all the character templates are of fixed height. The height of the template is determined by calculating the average line height of the images that are used to create templates. When an image is given for recognition it is compressed until its average line height is equal to the height of the templates. Having compressed the image, the templates are matched against each character in the image.

2.5. Segmentation stage

In this stage the image that is expected to recognize is broken down into lines and each line is further broken down into characters. These segmented characters are matched against character templates. Following procedures are used for segmentation.

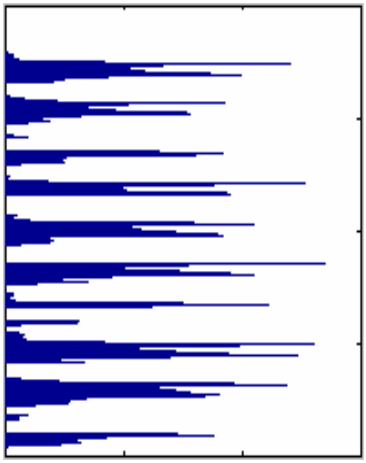
2.5.1. Line segmentation. The image is segmented into the lines based on the information provided by the procedures Number of Lines and Line Information. When noise is present this procedure cannot segment lines properly. (See Section 2.4.3)

2.5.2. Character segmentation. Having segmented the image into lines each line is segmented into characters by this procedure based on the information provided by the procedures Number of Characters and Character Information. When noise is present this procedure cannot segment characters properly. (See Section 2.4.4)

3. Results

The results of some of the procedures described in the section 2 are given in this section.

3.1. Horizontal projection



"එකමුතුව යනු විවිධත්වයේ එකමුතුව යි. විවිධත්වය වනාහි ඒකාබද්ධතාවේ විශ්ලේෂණයෙන් පහළ වෙයි. විවිධත්වයෙහි වටිනාකම එකමුතුවෙහි වටිනාකමට සමාන වෙයි. කුමක් හෙයින් ද යත්, එකමුතුව මතු වී ක්‍රියාත්මක වන්නේ විවිධ කොටස් පාලනය කිරීමෙන් හෙයිනි. කලා කෘතියේ එකමුතුව මනුෂ්‍ය ශරීරයෙහි එකමුතුවට සමාන කිරීම ඉතා මත් ම යෝග්‍ය වෙයි. කලා කෘතියක් මෙන් ම මනුෂ්‍ය ශරීරය ද ස්වයං-ශක්ති සම්පන්න ඒකීය සමස්තයකි. එහි ඒකාබද්ධ ජීවය රඳා පවත්නේ විවිධ ඉන්ද්‍රියන්ගේ ක්‍රියාකාරිත්වය මත ය ඒ ඉන්ද්‍රියෝ සමස්තයෙන් වෙන් කළ කල්හි ප්‍රනෂ්ට වෙත්"[1]

මේ සම්බන්ධයෙන් සුසාන් කේ ලැහර් මහාචාර්යවරියගේ අදහස

Figure 8: Horizontal projection of an image

3.2. Vertical projection

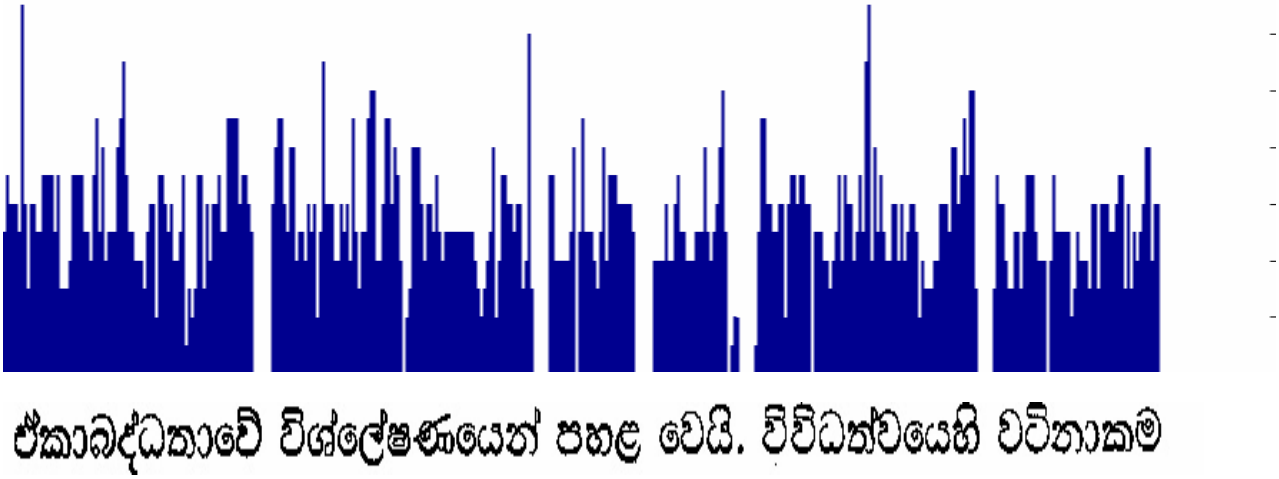


Figure 9: Vertical projection of an image

3.3. Line segmentation

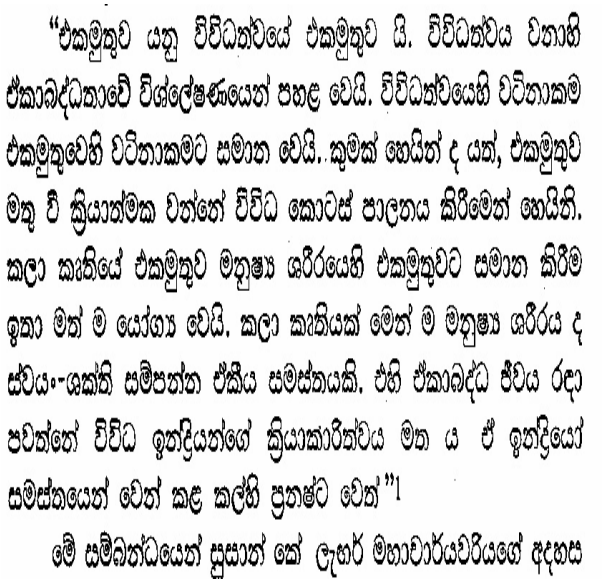


Figure 10 (a): Before line segmentation

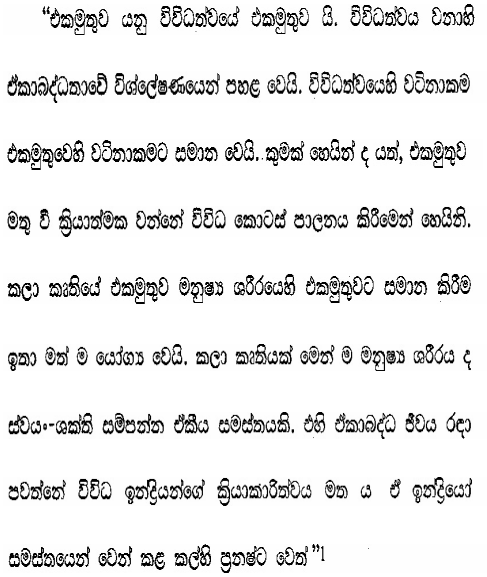


Figure 10 (b): After line segmentation

3.4. Character segmentation

Figure 11 (a): Before character segmentation

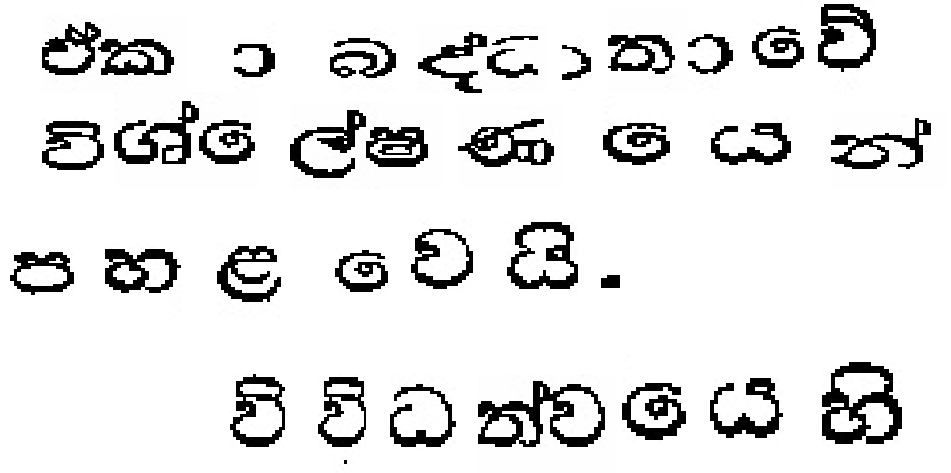


Figure 11 (b): After character segmentation

4. Discussion

The main objective of OCR is to recognize scripts and punctuation marks appear in an image by applying image processing and learning algorithms or deterministic methods. Preprocessing is a mandatory requirement for any of these techniques. At that stage the quality of the image is improved, relevant segments of the image are selected and data present in the image is standardized.

In the present Preprocessing engine consists of almost all of these functionalities and they are realized through standard algorithms. For example, noise removal is using Gaussian Filtering, line and character segmentation is done based on the information obtained from horizontal and vertical projections.

The issues that have to be addressed to improve the outputs of the Preprocessing engine were mentioned in the relevant sections, for example line and character segmentation.

It is expected to release a toolkit that can be used for image preprocessing based on the research and development activities done to build the present Preprocessing engine.