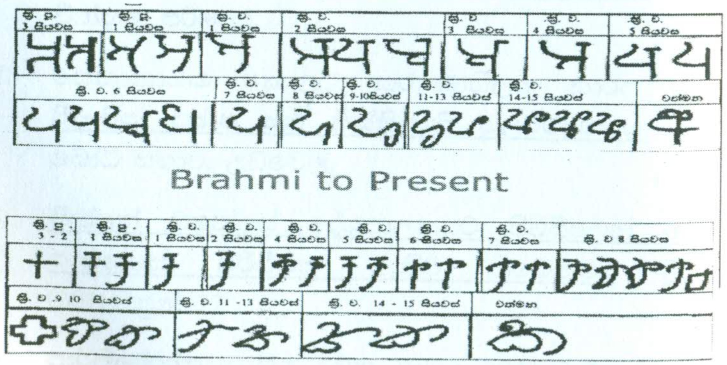
Feature Extraction and Classification of Sinhala Fonts

- B. Venura Lakshman

1) Introduction

1.1 About Sinhala Language

The Sinhala language is spoken by more than 15 million people and is the national language of Sri Lanka which has a high literacy rate of over 90%. It is said that Sinhala Language comes from north Indian *Brahmi* Letters. [1]



Brahmi to Present

1.2 Information Technology in Sinhala Language

The first efforts to introduce Sinhala language computing in Sri Lanka occurred in the 1980's. [2] Due to the lack of bitmapped devices, they required that the fonts be embedded in the displays and printers, and were thus limited to specific hardware. This limited their adoption, and they were eventually discontinued. Thereafter, Sinhala fonts were developed for the Macintosh, which are widely used in publishing. With the advent of Microsoft Windows, several organizations produced Sinhala fonts and word processing packages, which are currently in use. Sinhala was included in Unicode in 1998, but there was no implementation even by 2002. [3], [4]

A standard Sinhala encoding, known as SLASCII, was approved by the Sri Lanka Standards Institute as SLS 1134 in 1996. SLASCII has a structure similar, but not identical, to the ISCII standards for Indian languages. Digitizing printed document is successful in languages like English, but is a challenge for the Sinhala Language. English OCR is widely used in postal services...Etc. Linux group in Sri Lanka is working for Sinhala OCR & other applications. Also private firms are working for Sinhala application. [5]

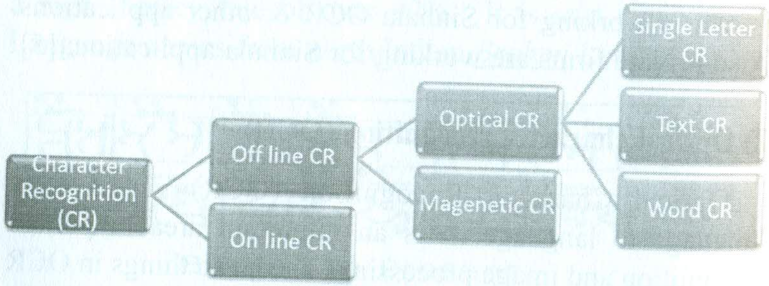
2) Optical character recognition (OCR)

Optical Character Recognition (OCR) varies from language to language .It is an important area in pattern recognition and image processing. The basic things in OCR are extracting the features & classification of relevant characters.

The roots of the character recognition technology go back a long way. In 1870, Carey (Boston) patented an image transmission system using a mosaic of photocells (the retina scanner). In 1890, Nipkow (Poland) invented a device where an image was scanned line by line (sequential scanner). Those systems were not intended to perform recognition but rather transmission of the images. However, the idea of transforming the data into a more suitable form for processing was underlying.

The first trace we can find of a true reader, that is a machine that converts printed characters into code, is in the telegraph applications. In 1912, Goldberg patented a machine to read characters and convert them into the telegraph code. It was then possible to send messages without human intervention. In 1929, Tausheck obtained a patent named "Reading Machine" in Germany. This patent reflects the basic concept of today's OCR. The principle of Tausheck's patent is template matching which is still used in some applications even today. Figure illustrates the detailed structure of this patent.

3) Different Families of Character Recognition (OCR)



3.1 On-line Character Recognition

In on-line character recognition applications, the computer recognizes the symbols as they are drawn. The typical hardware (like touch screen, light pen...etc) is used for data acquisition. Also it is using the digitizing object, which can be electromagnetic, electrostatic, pressure sensitive, and so on; a light pen can also be used.

As the character is drawn, the successive positions of the pen are memorized and are used by the recognition algorithm.

3.2 Offline Character Recognition

Off-line character recognition is performed after the writing or printing is completed. Two families are usually distinguished: magnetic and Optical Character Recognition.

- In Magnetic Character Recognition (MCR) the characters are printed with magnetic ink and are designed to modify in a unique way a magnetic field created by the acquisition device. MCR is mostly used in banking applications, as for example checks reading, because overwriting or overprinting these characters does not affect the accuracy of the recognition.
- Optical Character Recognition, which is the field investigated in this document, deals with recognition of characters acquired by optical means, typically a scanner or a camera. The symbols can be separated from each other or belong to structures like words, paragraphs, figures, etc. They can be printed or handwritten, of any size, shape, or orientation.

3.3 OCR Applications

The number of applications where optical character recognition is involved is very large. They can be roughly separated in three main classes:

- Letter readers: Only separated characters have to be recognized. For example form reading engines read symbols that have been written in separated boxes.

- Word readers: The input document is composed of only few words. For example addresses on envelopes in mail sorting applications.

- Text readers: The application attempts to read and rebuild entire documents. It first divides the page into its components (paragraphs, header, footer, figures...). Each written component is divided into words and finally into single characters which are sent to the OCR engine. The input documents can be printed (articles, books), handwritten, or both (facsimile). The recognized characters can be used for example to reconstruct the input document for further processing with a text processing application.

4.0 OCR in Sinhala Language

Colombo University Sinhala OCR

Only one commercially available OCR application could find that can recognize Sinhala fonts. The results will give as a text (with .txt file extension) file in to a separate folder. This Software was developed under PAN Localization Project, Language Technology Research Laboratory, University of Colombo School of Computing. This OCR system requires totally noise removed and skew-corrected images in jpg format as its input. In addition to that, it requires matching type of font to select (E.g. Abaya, Lakbima…etc)

The main disadvantage is this software is, it requires well processed image and requires a good image processing knowledge (For instance Adobe Photoshop). Also it supports for only selected font types. There are more than 100 true font types that are available for Sinhala, but this OCR supports up to 8 font types.

5.0 Steps in OCR

There are two steps in building an OCR application

a. Training

b. Testing.

These steps can be broken down further into sub-steps. They are shown here

5.1. Training

a. *Pre-processing* – Processes the data so it is in a suitable form for the classifier. (Gray & binary)

b. *Feature extraction* – Reduce the amount of data by extracting relevant information—usually results in a vector of scalar values. (We also need to NORMALIZE the features for distance measurements).

Features can be roughly classified by their nature in four groups:

1. Bitmap itself: This group is apart from the others since the features are the bitmaps themselves.

2. Distributions: The features are statistical measures of distributions of points on the bitmap, the contour curve, the profiles, or the HV-projections. Widely described methods are: zoning, n-tuples, characteristic loci, and moments.

3. Series expansion coefficients: Images or curves can be expressed as infinite series (Fourier, Walsh, Cosinus...) and approximated by retaining only the first K terms of the sums. The coefficients or more often, combinations of the coefficients of these terms are used as features.

4. Structural features: The patterns are analyzed in terms of their structure: number of strokes, their relative orientations and positions, number of loops, etc. This information is used as features and classified using a special classification method.

c. *Model Estimation* – from the finite set of feature vectors, need to estimate a model (usually statistical) for each class of the training data.

5.2. Testing

a. *Pre-processing*-(same as above)

b. *Feature extraction* – (same as above)

c. *Classification For* one figure it can be understood that there are two classifiers that perform the recognition in parallel. A module then compares the results.

6.0 Algorithms for different stages of the OCR System[7][9]

6.1 Algorithm for OCR Train

1. Load image file.
2. Perform the Segmentation Operation
3. Get correct identification of each digit image from user
4. Find Features for each Character.
5. Estimate models.
6. Create Neural Network and Train the network using labeled input.

6.2 Algorithm for OCR Engine

1. Load image file.
2. Load Trained Data
3. Perform OCR Operation (for the loaded image)
4. Calculate accuracy, recognized rate according to the option.
5. Save the document as text document or send it to MSWord for further processing & terminate the program.

6.3 Algorithm for OCR operation (Image)

1. Convert the loaded image into binary.
2. Perform the *linebreak()* operation on the image to get the number of lines and their index.
3. For each line perform the following operations:-

a. Split each character from the line.

b. call splitChars() for the characters which can't be split using regionprops

c. for each character perform the following operation

i. resize the image.

ii. Call FindFeatures() to find the features and classify the digit using features.

iii. Call Trained_Neural_Network() to find the best Character using neural network.

d. Combine the result of both the classifiers to obtain the correct classification.

6.4 Algorithm for OCR operation (Image) summary

1. Convert the loaded image into binary.
2. Perform the *linebreak()* operation on the image to get the number of lines and their index.
3. For each line perform the following operations:-

   a. Split each character from the line.

   b. call splitChars() for the characters which can't be split using region props

   c. for each character perform the following operation

      i. resize the image

      ii. Call FindFeatures() to find the features and classify the character using features.

      iii. Call Trained_Neural_Network() to find the best character using neural network.

   d. Combine the result of both the classifiers to obtain the correct classification.

7.0 Further development

Further researches & development in this field can be done in Sinhala web translator, develop SDK & apply where Sinhala OCR is required.

List of References

[1] Evaluation of Sinhala Character at **http://www.ceylon-online.com**, 2008

[2] Gihan V. Dias, Challenging of enabling IT in the Sinhala Language, ICTA Sri Lanka, 2005

[3] Muthu Nedumaran, Sinhala Unicode Developer Work shop, 2004

[4] Unicode website at **http://www.unicode.org**, 2008

[5] Samaranayake, V. K., Nandasara, S. T., Dissanayake, J. B., Weerasinghe, A.R., Wijayawardhana, An introduction to Unicode for Sinhala Character. *University of Colombo School of Computing, Sinhala Department, University of Colombo, 2003*

[6] Per-Ola Kristiansson, **(per-ola@swesign.com)** Defeating a simple CAPTCHA using Optical Character Recognition Göteborg University - Computer Science,Göteborg, Sweden 2007.

[7] Brian D. Hahn & Daniel T. Valentine, Essential math lab for Engineers, ISBN 13: 9-78-0-75-068417-0, 2007

[8] Java programming language official web site at **http://java.sun.com**, 2008

[9] Math lab official web site at **http://www.math-works.com**, 2008

[10] Wikipedia free encyclopedia at **http://www.wikipedia.org**, 2008