AN EFFECTIVE APPROACH FOR SINHALA HAND WRITTEN CHARACTER RECOGNITION (SHWCR)

H.M. Akila Jayasanka[1], H.M.R.S. Sugathadasa[2], G. Wijerathne[3]

[1]ICT Center, Wayamba University of Sri Lanka, Kuliyapitiya
[2]Faculty of Applied Sciences, Rajarata University of Sri Lanka
[3]Faculty of Science, University of Kelaniya

ABSTRACT

Sinhala Optical Character Recognition is a less developed area mainly due the complexity of the process. In this study we propose an effective method for Sinhala hand-written character recognition, which can be used to convert hand-written documents into digital form. The system takes scanned images as an input, then analyses characters step by step and then recognizes the character using machine learning techniques. SHWCR gives solution to the limitations of OCR for Sinhala characters in real environment which is each person has their own unique hand writing. A trained neural network was used to recognizing the characters. The proposed method is tested on all simple 18 characters of Sinhala text. Experimental results using standard fonts show that the accuracy percentage varies for each character. Developed software tool was able to identify all of the tested 18 characters with above 50% accuracy.

***Key words*:** OCR, Neural network, Binarization

1. INTRODUCTION

Documenting is an essential role in almost all offices in both government and industries. Even though English language is used in many organizations in the urban areas, Sinhala is still the main language used in documenting purposes in most part of the country. As a result, a large collection of written documents are piled up in these offices. Therefore there is a huge requirement for simple but robust automated applications such as OCR applications in Sinhala script which is alphabetic in nature and is written from left to right. The alphabet consists of 61 symbols: 18 vowels, 2 semi-vowels and 41 consonants.

There are few solutions suggested for the same problem [1, 2] but still they contain limitations to work on a real environment such as each person has their own unique hand writing, each writing can be in different backgrounds, characters can be touched, noise can be there or characters are not aligned properly etc. [3]. Each Sinhala character has its own physical features. And each person has his own unique hand writings, so recognition should be accurate regardless of the character shape. Simple typesetting in Sinhala wastes considerable amount of time and money which should have been effectively used for other greater purposes of the community. This set the motivation to provide an effective solution for the problems depicted above using an OCR which can recognize wide variety of fonts as well as handwriting with the power of image processing and machine learning techniques

2. METHODOLOGY

The approach of the solution consists of four-phases which are data gathering, pre-processing, feature extraction and classification.

2.1. Data Gathering

Collection of Sinhala hand written characters were used as the data set which is needed to train the neural network as well as to test. For this research 500 different hand written characters were collected from different personals within the age16 to 55, for each 18 vowels in Sinhala alphabet.

2.2. Pre-Processing

Pre-processing stage aims at segmenting the input for segmentation process. The main objectives of this process are noise removing of the input image, smoothing the input image, image binarization, and character segmentation. In order to achieve these objectives the following techniques are used.

2.2.1. Noise Removing

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighbouring entries. The pattern of neighbours is called the "window", which slides, entry by entry, over the entire

signal. For 1D signal, the most obvious window is just the first few preceding and following entries, whereas for 2D (or higher-dimensional) signals such as images, more complex window patterns are possible (such as "box" or "cross" patterns). Note that if the window has an odd number of entries, then the median is simple to define: it is just the middle value after all the entries in the window are sorted numerically. For an even number of entries, there is more than one possible median. This algorithm is used to remove noise on the input image before take it for feature extractions. To make feature extraction more accurate used smoothing on the noise removed inputs.

2.2.2. Smoothing

The smoothing operation eliminates the pixels created improperly due to the hand motion or due to incorrect paper set up during data acquisition. The smoothing algorithm scans the two dimensional array of the binarized word image by row by row, any points has one diagonal neighbor or two neighbors on one horizontal/vertical line or three neighbors on one horizontal/vertical line will be removed

2.2.3. Image Binarization

In Image binarization, the text image which is gray scale image is converted into a binary image with each pixel taking a value of 0 or 1 represent an individual pixel of image. Here consider the background pixels have a value of 1 and the foreground pixels have value of 0. Actually the main target is finding a vector from the image. So image is processed and then binary image is created. Here, the Otsu's threshold algorithm is used to binaries the gray scale image [4].

2.2.4. Character Segmentation

This is one of the most important sections in character recognizing. After image is binarized segmenting image into few contours is done by simple algorithms. These obtained segments can process more if needed or used for extract features.

2.3. Feature Extraction

Once the contour of the image is obtained we apply freeman chain-code.

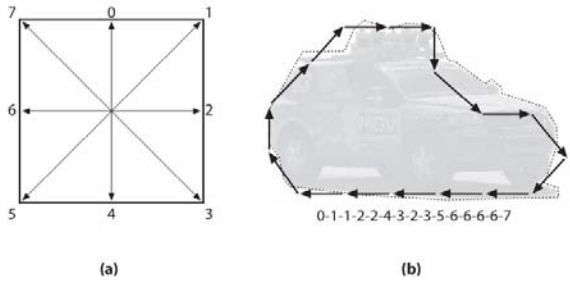

Figure 1: Eight chain-code directions

Figure 2: Eight chain-code of letter "ඍ".

Freeman chain code generated for the letter "ඍ"

0	7	7	0	0	0	0	0	0	0	0	0
1	0	0	1	0	1	0	1	7	7	0	0
7	1	1	2	2	2	2	6	5	6	6	5
4	4	3	3	1	1	1	2	4	3	2	1
2	2	1	0	0	0	0	0	7	0	4	3
3	3	3	3	2	2	2	2	2	1	2	2
2	2	1	2	2	4	4	6	6	6	5	6
6	5	6	6	6	6	6	6	6	7	5	4
4	4	4	4	4	4	0	0	0	0	0	7
7	5	6	5	6	5	6	6	5	5	5	5
4	5	4	4	3	3	2	4	4	3	1	0
4	3	4	3	3	3	2	2	1	1	0	0
1	0	0	0	4	4	4	4	4	5	4	4
5	5	6	6	7	7	7	0	7	5	4	4
4	5	4	6	6	0						

The coordinates of the boundary pixels are obtained first, based on these coordinates the chain code of the character image is found.

The normalized chain code is obtained by transforming it to a two dimensional matrix. The first row of this matrix contains the value of the chain code, and the second row contains the frequency of occurrence of that value (10). For example, if the chain code of a given character is: 88883331111225833312 then it can be converted into the following form of a 2×9 matrix:

8 3 1 2 5 8 3 1 2

4 3 4 2 1 1 3 1 1

We remove all values whose frequencies are 1. For instance, in the above example, the chain code will be reduced to:

8 3 1 2 3

4 3 4 2 3

8 3 1 2

4 6 4 2

The process of removing the less-frequent digits can be continued. For instance in our test, the frequencies less than or equal to five were deleted. Again in the resulted chain code the frequency of each remained digit is summed. Then to transform the chain code matrix to a normalized chain code with length of 10, the relative frequency of each digit is computed using:$F_i^n = \frac{F_i}{\sum F_i} \times 10$ where F_i^n is the normalized frequency and F_i is the each digit in the chain code respectively. In the above example we will obtain:

8	3	1	2
2.22	3.33	2.22	1.11

Then the normalized frequency would be rounded to the nearest decimal which in turn would be concatenated to generate the 10 length chain code:

8833311222

2.4. Classification

The principal function of a pattern recognition system is to yield decisions concerning the class membership of the patterns with which it is confronted [5]. In the context of an OCR system, the recognizer is confronted with a sequence feature patterns from which it must determine the character classes. There are two steps in building a classifier, which are called training and recognition.

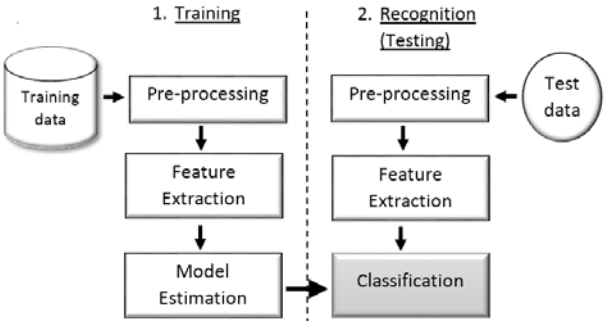

Figure 3: The pattern classification process

2.4.1. Training

1. Pre-processing – Processes the data to obtain the clear contour of the image
2. 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)
3. Model Estimation – from the finite set of feature vectors, need to estimate a model (Usually statistical) for each class of the training data

2.4.2. Testing

1. Pre-processing
2. Feature extraction
3. Classification – Compare feature vectors to the various models and find the closest match. One can use a distance measure

3. RESULTS AND DISCUSSION

MATLAB used to implement the proposed system and neural network, and for normalizing chain code is used and trained neural network is used for recognizing characters.

Below graph depicts the results obtain from implementing proposed method. 18 characters were used individually for classification. As future work of this project this algorithm needs to be improved to overcome the limitation of OCR such as noisy backgrounds and touched characters.

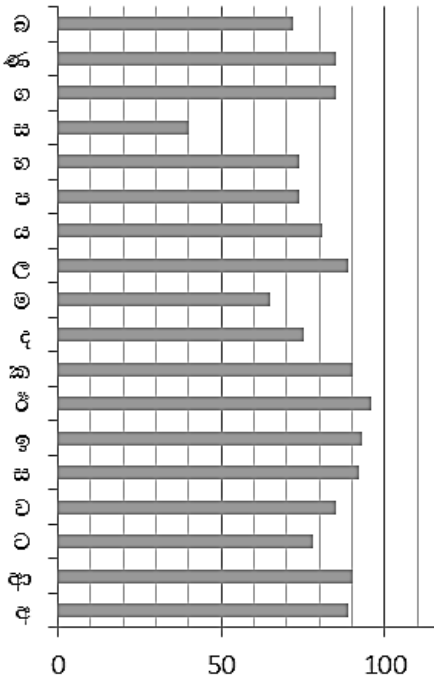

Chart 1: Percentage of recognizing tested 18 Sinhala characters

4. CONCLUSION

This research paper used new freeman chain code-based approach for identification of Sinhala characters. After the pre-processing and after obtaining contour of the characters, features are extracted using Freeman chain code. Classification process is used to recognize a new character with the help of neural network. The proposed method is tested on all simple 18 characters of Sinhala text. The proposed method is tested on all simple 18 characters of Sinhala text. Experimental results using standard fonts show that the accuracy percentage varies for each character. Developed software tool was able to identify all of the tested 18 characters with above 50% accuracy.

5. REFERENCES

[1] J. Bigun, and H. L. Premaratne, "*A segmentation-free approach to recognise printed Sinhala"*.

[2] B. Jayasekara, and L. Udawatta, "*Non-cursive Sinhala handwritten script*".

[3] M. Karunanayaka, C. A. Marasinghe, and N. D. Kodikara, "*Thresholding noise reduction and skew correction of Sinhala*", *APR Conference on Machine VIsion Applications*, 2005.

[4] M. D. Adnan, and S. Shoeb, "*Research report on Bangla optical chracter*" *Center for Research on Bangla Language*.

[5] J. Tou, and R. C. Gonzalez, "*Pattern recognition principles*" Addison-Wesley Publishing Company, Inc., Reading.

[6] "*Wikipedia Sinhala language*", [Online]. Available:http://en.wikipedia.org/wiki/Sinhala_language.

[7] R. Singh, and M. Kaur, "*OCR for Telugu script using back-propagation based classifier*", International Journal of Information Technology and Knowledge Management, 2010.

[8] G. Bradski, and A. Kaebler, "*Learning computer vision with the openCV library*".

[9] D. A. Satharasinghe, Deputy Director, Information Unit, [Online]. Available: http://www.statistics.gov.lk/CLS/index.htm.

[10] K. R. Singh, and M. Kaur, "*OCR for Telugu Script Using Back-Propagation Based Classifier*".

[11] S. R. Kodituwakku, and P. S. Nilanthi, "*Investigating a fuzzy approach for hand written Sinhala character recognitio*n", 2010.

[12] R. K. Rajapakse, and A. R. Weerasinghe, "*A neural network based character recognition system*", 1996.

[13] V. A. Kumar, and M. Ramakrishnan, "*Legacy of footprints recognition- A review*", vol. 35, 2011.

[14] J. Rosenberg, J. Weinberger, and C. Huitema, March 2003. [Online]. Available: http://tools.ietf.org/html/rfc3489.