Sinhala Braille Translator

[1]N.M.T De Silva and [2]S.R Liyanage,
[1]Scientist, Information Technology, National Building Research Organization, Colombo, Sri Lanka
[2]Senior Lecturer, Faculty of Computing and Technology, University of Kelaniya, Colombo, Sri Lanka

***Abstract*—** Braille is one of the most valuable and indispensable method used for written communication by the blind. Typically, editing and reprinting of Braille that are embossed on paper are done manually. It is a time consuming and labor intensive task. Motivation to understand Braille codes by sighted people has encouraged the development of Optical Braille Recognition (OBR) for different languages across the World. However, OBR for Sinhala Braille has not been attempted previously. A communication gap exists among the blind and the Braille illiterate sighted people in the Sri Lankan society as written communication is restricted. This paper addresses this communication gap between the blind and sighted people using the Sinhala language. The system is capable of extracting Braille characters from a Braille document followed by decoding them into Sinhala characters. The decoded Sinhala characters are further normalized to Sinhala text that is legible for a sighted person. The system is capable of recognizing an image of embossed Sinhala Braille and then converts it to Sinhala text. It takes the scanned braille documents using normal flat-bed scanner and those images undergoes pre-processing steps. Noise removal is done using gray scaling and the noise filtered image is converted to a threshold image. Then by applying edge detection techniques braille documents are subjected to segmentation. For each and every character, feature extraction is carried out identifying main dots. Then the extracted features are used to train the classifier to predict the Sinhala character. Artificial Neural Networks (ANN), Support Vector Machines (SVM) and K-Nearest Neighbour (K-NN) classifiers were tested on the sample Braille documents. Finally decoded characters are normalized to Sinhala text which is in human understandable form. The prototype Sinhala Braille Translator was tested on documents collected from Library, University of Kelaniya. The system was able to recognize Single sided Sinhala Braille with an average accuracy of 91.2%.

***Keywords*:** *Braille, Classification, Feature extraction, Image processing, Sinhala fonts*

I. INTRODUCTION

Braille is a writing system that enables visually impaired people to read and write using a series of raised dots with their fingertips. It is the first digital form of writing and it was named after its creator, Frenchman Louis Braille. Braille characters are small rectangular blocks called cells that contain tiny palpable bumps called raised dots. The number and arrangement of these dots distinguish one character from another. The mappings vary from language to language, therefore various braille alphabets exists.

Many efforts have been made to translate Braille characters to different languages worldwide. An attempt was made by J. Mennens, L. Tichelen, G. François and J. J. Engelen [1] to develop a system that runs on commercially available scanners for both single- and double sided-character recognition. In 2011 Al-Saleh, Amany, Ali El-Zaart, and Abdul Malik Al-Salman [2] made dot detection of Braille Images using a mixture of beta distributions. In this attempt commercially available flatbed scanner is used to scan the document with both side printed. Several researchers have made efforts to recognize English Braille documents. The work of R. T. Ritchings, A. Antonacopoulos and D. Drakopoulos [3] is one of the first approaches to use commercially available flatbed scanners for English Braille document recognition. C.M. Ng, Vincent Ng and Y. Lau also attempted a system that translates Braille characters to English characters [4]. Another robust Braille recognition system has been developed by Antonacopoulos and Bridson [5]. Lisa Wong, Waleed Abdulla and Stephan Hussmann [6] has proposed an OBR system that is capable of recognizing a single sided Braille page that preserves the format of the original document in the produced text file. They have also reported high classification accuracies with efficient recognition algorithms [6]. Braille recognition systems for Arabic language have also been developed. AbdulMalik Al-Salman, Yosef Alohali, Mohammed AlKanhal and Abdullah AlRajih [7] have attempted to recognize Arabic Braille embossed on both single-sided and double-sided documents using a regular flat-bed scanner. In 2010, Saad D. Al-Shamma and Sami Fathi [8] have developed a system that can recognize Arabic Braille characters and transcript them into text and voice. Halder et al. has attempted to recognize Bangla characters from Braille notations [9]. In 1995, Hentzschel and Blenkhorn [10] presented a system for optical Braille recognition based on twin shadows approach and also by using dot detection module incorporated in the OBR system is proposed by Y.Oyama, T. Tajima, and H. Koga [11] in 1997 and it is designed to detect both recto and verso Braille dots.

Although many braille character translators have been developed for many languages around the world, an automated system to recognize Sinhala Braille characters has not been developed. In this study we propose a Sinhala Braille Translator for translating written Braille documents to Sinhala text. The proposed system uses both formatted and unformatted Braille documents for character recognition after preprocessing the documents. A novel character segmentation algorithm was attempted combined with three different classifiers. The performance of the developed system is evaluated on a data set of Sinhala Braille documents collected from the Library of the University of Kelaniya.

The rest of the paper organized as follows: The section II describes the methodology used. Section III presents results obtained and section IV depicts the discussion and finally section V concludes and remarks about some of the aspects analyzed in this paper.

II. METHODOLOGY

Each Braille character or "cell" is made of 6 dots arranged in a rectangle comprising 2 columns of 3 dots each. A dot may be raised at any of the 6 positions. These dots are numbered downward 1, 2, 3 on the left, and 4, 5, and 6 on the right as shown in Fig. 1 for easy identification.

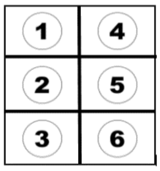

Figure 1: Segmented Braille Cell

The methodology used to identify these raised dots in a braille document is summarized in figure 2 and are described in the subsequent subsections.

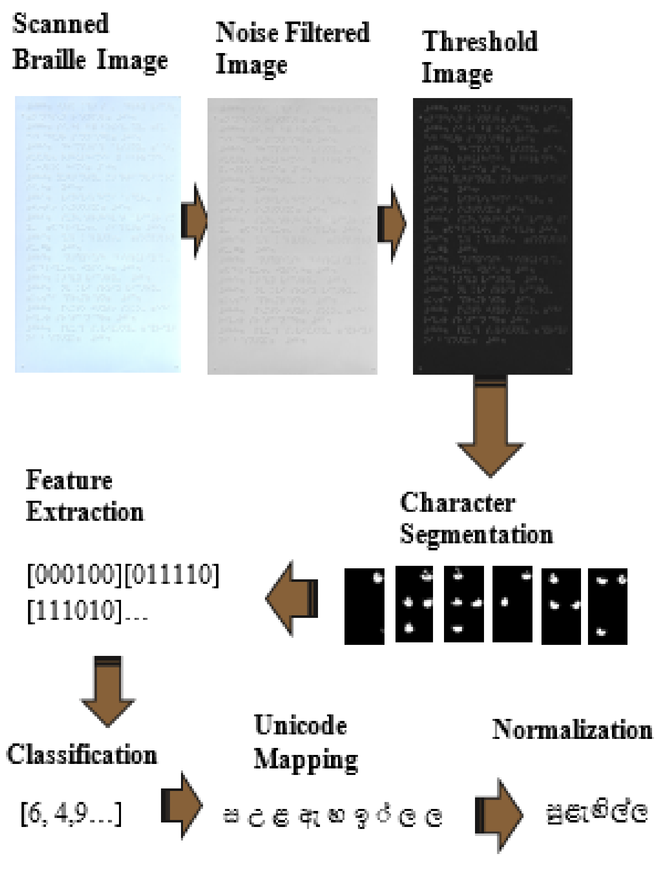

Figure 2: Methodology

A. Data set preparation

Unformatted single sided braille documents were considered in this study. Braille documents available in the Library of University of Kelaniya were used for testing the algorithms. After the Braille documents were collected, scanning was carried out using a flat-bed scanner was used with horizontal and vertical resolutions of 200 dpi for digitization process.

B. Noise Removal

Presence of noise in documents reduces the image quality consequently deteriorating the effectiveness and accuracy of subsequent steps such as edge detection, image segmentation, feature extraction and pattern recognition [12].

Mean filtering [13], median filtering [14] and gray scaling [15] were attempted for noise removal. Gray scale filtering was selected for noise removal because of its relatively low computational cost, low memory consumption, simple implementation, simple analytical process and preservation of more image details than the other two methods.

C. Edge Detection

Edge detection is an essential preprocessing technique in image processing. It is necessary to first extract the relevant areas from the braille documents to get the information contained in them. Two commonly used algorithms for edge detection were attempted for this task; Canny Edge Detector [16] and Sobel Edge Detector [17]. Canny algorithm produces single edges while the Sobel produces wider edges. Consequently, Canny does not damage the features of the characters much while the Sobel tends to distort the features of the characters considerably. Both these algorithms for edge detection consume a lot of processing time and power resulting in a poor performance for an online system. Because of the drawbacks of the existing algorithms an edge detection algorithm was not used, but a novel algorithm was implemented to perform the segmentation without the use of edge detection algorithms.

D. Character Segmentation

In the segmentation section the main expectation is to segment isolated characters for the classification process in an efficient and accurate manner. Accuracy of the segmentation algorithm should be considerably high in order to reach acceptable classification rates.

In order to detect edges, the white pixels in the black background were first identified. To segment character by character, basic logic was to find the coordinates of the left upper corner of the first character. To accomplish this, scanning was conducted horizontally from left to right increasing the horizontal coordinates, which is much easier than scanning vertically from bottom to top direction, because the width of the document is less than its height.

Thresholding was carried out to discretize the grayscale pixels to black and white pixels since segmentation is not possible on an image with pixels of different intensity levels.

Once the first top white pixel in the black background was recognized, segmentation started from that particular coordinate shifting by the size of the braille cell, which has pre-defined size, until the end of the document is reached. The segmented characters were numbered in order and stored for feature extraction. This novel segmentation algorithm resulted in noise minimization without the use of costly edge detection algorithms for segmentation.

E. Feature Extraction

Feature extraction is a representational mechanism of the Braille image. The main function of this process is to extract Braille dots from the binary image. The image capturing, pre-processing and segmentation stages aim to make the image suitable for different feature extraction algorithms.

The selection of image features and corresponding extraction methods are probably the most important step in achieving high performance for an OCR system. At the same time, the selected image features and the extraction method also determine the nature and the output of the image-pre-processing and segmentation steps. Some image features and extraction algorithms work on colour images, while others work only on grey level or binary images. Feature extraction phase can be considered as the most important part of braille character translation, because the accuracy of the translated character heavily depends on the extracted features of the cells.

A Braille character is described as a region in the image and this region is divided into six equal compartments (two across, three down), as shown in Fig.1. In order to extract the features dots must be identified in the Braille cell. Initially, the Braille cell is partitioned to 6 parts based on its height and width. Next, all the connected components in each and every region are calculated. Then the most significant connected component in each region is selected. 1 and 0 are assigned according to the availability of dots considering each region in a cell separately. This process is iterated until all the images are process. Finally all binary arrays for all segmented braille characters are stored.

F. Classification

A classification technique (or classifier) is a systematic approach to build classification models from an input data set. The model generated by a learning algorithm should both fit the input data well and correctly predict the class labels of records it has never seen before. Therefore, a key objective of the learning algorithm is to build models with good generalization capability; i.e., models that accurately predict the class labels of previous unknown results.

Table 1: Summary Of Classification Results.

Character	**Number of Characters**	**KNN (%)**	**ANN (%)**	**SVM (%)**
ආ	90	100	97.77	100
ම	101	100	97.029	100
ර	80	98.75	100	100

In this study, the performance was tested with a few classification methods; artificial neural networks (ANN) [18], K-nearest neighbour (K-NN) [19] and Support vector machine (SVM) [20]. Table 1 summarizes the results obtained for these classifiers in two class classification (one vs rest) in classifying three selected characters.

Even though ANN smoothly classifies features of braille characters into groups, to fully implement standard neural network architecture with 62 outputs, requires a lot of computational resources. This makes ANN very complex and expensive method in this situation.

The performance of the SVM classifier was also investigated for this setup. The classification results for K-NN for two class classification was not significantly different from those obtained with SVM. Since there are 62 classes, formulating a multi-class classifier based on binary SVM classifier is time consuming and monotonous.

K-NN is a type of instance-based learning. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors [19].

Choice of K value is very important as the performance of a K-NN classifier is primarily determined by K. Generally, larger values of K are more immune to the noise presented, and make boundaries smoother between classes. In this study 3 is chosen as the K value, because when higher K values such as 5, 7 is used the accuracy decreased.

After analyzing all factors such as complexity, time consumption, processing power, and accuracy of results K-NN was selected as the classifier for this scenario because of its simplicity and high accuracy.

G. Unicode mapping

Unicode mappings are done according to character groups during translation. In the developed system, Unicode mappings for 52 Sinhala characters, 9 punctuations and 10 numbers were implemented.

In Braille characters the numerals were identified using "⠼" braille character. If "⠼" character appears prior to any braille character it is considered as a number. A problem occurred as the Sinhala characters "අ, ඞ ච, ද, ඵ, ෆ, ග, හ, ඉ, ඡ" also has the same braille character "⠼". Hence, a different identification has to be formulated for numerals instead of Sinhala Unicode mapping.

Similarly, for the Sinhala character "්" hal kereema there exists a separate braille character " ⠈". This character can appear prior to any consonant character. Furthermore, spaces and punctuation marks also have special leading characters in Braille to distinguish them. Therefore, numerals, hal kereema, spaces, and punctuations marks have to be identified before applying Unicode mapping.

H. Normalization

In Sinhala braille, mapping Sinhala vowels has been used in two different ways according to the position they appear. Usually in Sinhala language vowels appear at the beginning of words. But in Sinhala braille, vowels can appear at the beginning of a word or in the middle of a word. Therefore, a few scenarios must be considered.

If vowels appear after punctuations such as a space, full stop, brackets, braces, exclamation mark etc. these vowels are considered as the beginning of a sentence and mapped as they are, i.e. ආ,ඇ,ඈ,ඉ,ඊ,උ,ඌ,එ,ඒ,ඓ,ඔ,ඕ,ඖ mapped to their corresponding Unicode values. But, if these vowels occur in the middle of a word these must be mapped differently, considering the character before that vowel. As an example Table II shows the above vowels mapped for the consonant ම.

Table 2: Vowels Mapping In Middle of a Word For The Consonant ම.

Vowels In The Middle of a Word With Consonant "ම"	**Mapping**
ම+ආ	මා
ම+ඇ	මැ
ම+ඈ	මෑ
ම+ඉ	මි
ම+ඊ	මී
ම+උ	මු
ම+ඌ	මූ
ම+එ	මෙ
ම+ඒ	මේ
ම+ඔ	මො
ම+ඕ	මෝ
ම+ඖ	මෞ

III. RESULTS

The implemented method was tested on the data set. K-NN classifier was used to test all the characters. Table 3 shows the results obtained for 52 Sinhala alphabetic characters. The corresponding International Phonetic Alphabet (IPA) [21] character for each Sinhala character is also shown. Table 4 summarizes the results obtained for all the characters. Results for all numerals 0 to 9 and punctuations such as comma, colon, full stop etc…are summarized in table 5 and 6.

Table 3: Results Obtained for Each Character

Character	IPA	Number of samples	No of Correctly identified Samples	%
අ	[a]	52	50	96.15
ආ	[aː]	90	83	92.22
ඇ	[æ]	48	43	89.58
ඈ	[æː]	36	34	94.44
ඉ	[i]	101	96	95.05
ඊ	[iː]	44	44	100
උ	[u]	35	33	94.28
ඌ	[uː]	41	41	100
එ	[e]	48	46	95.83
ඒ	[eː]	49	46	93.88
ඓ	[ai]	37	33	89.19
ඔ	[o]	30	30	100
ඕ	[oː]	39	33	84.62
ඖ	[au]	37	34	91.89
ක	[ka]	47	45	95.74
ඛ	[ka]	37	32	86.48
ග	[ga]	56	53	94.64
ඝ	[ga]	36	31	86.11
ච	([t͡ʃa])	40	36	90
ඡ	[t͡ʃa]	28	25	89.28
ජ	[d͡ʒa]	44	39	88.63
ට	[ʈa]	46	41	89.13
ඨ	[ʈa]	27	24	88.88
ඩ	[ɖa]	39	35	89.74
ඪ	[ɖa]	36	31	86.11
ත	[t̪a]	55	52	94.54
ථ	[t̪a]	32	29	90.62
ද	[d̪a]	56	50	89.28
ධ	[d̪a]	43	40	93.02
න	[na]	72	68	94.44
ප	[pa]	54	54	100
ඵ	[pa]	37	33	89.18
බ	[ba]	48	43	89.58
භ	[ba]	38	36	94.74
ම	[ma]	73	70	95.89
ය	[ja]	65	59	90.77
ර	[ra]	80	79	98.75
ල	[la]	48	43	89.58
ළ	[ɭa]	38	34	89.47
ව	[ʋa]	70	62	88.57
ශ	[sa]	37	26	70.27
ෂ	[sa]	39	33	84.61
ස	[sa]	80	77	96.25
හ	[ha]	56	47	83.92
ෆ	[fa,ɸa, pa]	37	30	81.08
ණ	[ɳa]	43	37	86.05
ඟ	[^{n}ga]	37	30	81.08
ඬ	[^{n}ɖa]	37	34	91.89
ඳ	[^{n}d̪a]	38	31	81.57
ඹ	[^{m}ba]	37	33	89.18
ඥ	[gna]	36	31	86.11
ක්	[k]	118	112	94.91
කං	[kan]	40	34	85

Table 4: Results Of Classifying Characters With KNN

Number of characters	Correctly classified Characters	Accuracy (%)
2604	2375	91.20

Table 5: Results of Classifying Numerals With KNN

Number of Numerals	Correctly classified Numerals	Accuracy (%)
490	474	96.73

Table 6: Results Of Classifying Punctuations With KNN

Number of Punctuations	Correctly classified Punctuations	Accuracy (%)
410	388	94.63

IV. DISCUSSION

The conversion from Braille documents into Sinhala text is a new area where much research has not been pursued. The proposed system has used some new techniques to recognize Braille cells using a standard scanner. The system has been tested with a wide variety of different sizes and different color single sided scanned Braille documents written in Sinhala.

The system takes advantage of the regular spacing between Braille dots within a cell, and the regular spacing between cells. The Braille documents can be embossed on a range of media and with different specifications such as page size, dot size, inter Braille dot distance and, inter Braille cell distance.

It is expected for most documents to have some defects such as slight variations in background color and small number of dark spots. In addition, image quality depends on the extent of scanning artifacts, such as the effect of non-uniform illumination. These problems generally cause the lightness of the highlighted part of Braille dots to be much less distinct from the page background and could be misclassified as the highlighted or shadow parts of dots.

Overall, the majority of the errors can be attributed to the quality of the image of the Braille document. Also, it should be pointed out that the quality of the Braille document itself is important. Incorrectly recognized characters will cause mainly due to very old documents with some of the protrusions flattened because of heavy use.

Various methods being carried out to improve the performance of the system. Experiments include scanning at different resolutions and color and grey level ranges.

However in this system, it does not require expensive or complicated hardware. It uses a flatbed scanner, which can be shared with other applications. Robustness to cope with low quality scans and defective worn-out documents is built-in at different levels, from the initial Threshold through to the flexible text.

The execution times of the implemented SBR system was evaluated on an Intel core i5 processor with 4GB of RAM.

The system took less than 10s classification of a free document with 720 braille characters.

CONCLUSION

This proposed system is capable of extracting Braille characters from a Braille document followed by decoding them into Sinhala characters and then normalization of the decoded Sinhala characters into legible Sinhala text.

An algorithm to detect dots in an image of embossed Braille material obtained by an optical scanner was proposed. Although the Braille dots have the same color as the background, they cast soft shadows when scanned with a standard flatbed scanner. These shadows are used to locate the dots on the page.

As a whole, the approach is successful as a cost-effective, fast and easy method to detect dots in Braille documents. It does not require expensive or complicated hardware.

The recognition rates of the characters have room for improvement. One way to improve accuracies is by implementing a Sinhala spell check algorithm. In the implemented Sinhala Braille Translator after all valid dots are extracted they are passed to the classifier and then recognition has been made. Post processing techniques (with the help of dictionary, thesaurus and grammar) can be incorporated to make the system more effective and robust. Such a system can automatically correct grammatical and spelling errors in recognized words. Automated character reading applications and character-to-speech engines can also be developed to improve the quality of life of blind people.

References

[1] J. Mennens, L. Tichelen, G. François and J. J. Engelen, “Optical recognition of Braille writing using standard equipment”, IEEE Trans. of rehabilitation engineering, II(4), pp.207-211,1994.

[2] Al-Saleh, Amany, Ali El-Zaart, and Abdul Malik Al-Salman,"Dot Detection of Braille Images Using A Mixture of Beta Distributions." Journal of Computer Science.VII (11), 2011.

[3] R. T. Ritchings, A. Antonacopoulos and D. Drakopoulos, “Analysis of scanned Braille documents”, Document Analysis Systems: World Scientific Publishing Company, pp. 413–421, 1995.

[4] C.M. Ng, Vincent Ng and Y. Lau, “Statistical Template Matching forTranslation of Braille”, In Proceedings of the Spring Conference on Computer Graphics (SCCC 1999) , pp.197-200, 1999.

[5] A. Antonacopoulos and D. Bridson,”A Robust Braille Recognition System”, Document Analysis Systems VI, A. Dengel and S. Marinai (Eds.), Springer Lecture Notes in Computer Science, LNCS 3163, pp. 533-545,2004.

[6] Lisa Wong, Waleed Abdulla and Stephan Hussmann, “A software algorithm prototype for optical recognition of embossed Braille”, Proceedings of the 17th International Conference on Pattern Recognition, pp.23-26, 2004.

[7] AbdulMalik Al-Salman, Yosef Alohali, Mohammed AlKanhal and Abdullah AlRajih, “An Arabic Optical Braille Recognition System”, Proceedings of the First International Conference in Information and Communication Technology and Accessibility(ICITA 2007), Hammamet, Tunisia, pp.81-87,2007.

[8] Saad D. Al-Shamma and Sami Fathi,”Arabic Braille Recognition and Transcription into Text and Voice”, Proceedings of the 5th Cairo International Biomedical Engineering Conference Cairo, Egypt, December 16-18, pp 227-231,2010.

[9] Santanu Halder, Abul Hasnat, Amina Khatun, Debotosh Bhattacharjee, Mita Nasipuri,”Development of a Bangla Character Recognition (BCR) System for Generation of Bengali Text from Braille Notation”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), III(1), pp. 5-10, 2013.

[10] T.W. Hentzschel and P. Blenkhorn, “An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach”, Journal of Microcomputer Applications, III(2),pp. 341-345, 1995

[11] Y.Oyama, T. Tajima, and H. Koga, “Character Recognition of Mixed Convex- Concave Braille Points and Legibility of Deteriorated Braille Points”, System andComputer in Japan, XXVIII (2), pp. 44-53, 1997.

[12] R.Moreno, M.Grana., D. M Ramik, & K.Madani, “Image segmentation on spherical coordinate representation of RGB colour space”, IET Image Processing, VI (9), pp. 1275-1283, 2012.

[13] R.Fisher, S.Perkins,A.Walker & E.Wolfart, “Spatial Filters – Mean Filter”,2003,[Online],Available: http://homepages.inf.ed.ac.uk/ rbf/HIPR2/mean.htm,[Accessed: June.12, 2016]

[14] R.Fisher, S.Perkins,A.Walker & E.Wolfart, “Spatial Filters – Median Filter”,2003,[Online],Available :http://homepages.inf.ed.ac.uk/rbf/HIPR2/median.htm,[Accessed: June.20, 2016]

[15] Bob Powell,”Convert Color to gray scale”,2003,[Online],Available:http://www.bobpowell.net/grayscale.htm,[Accessed: June.17, 2016]

[16] Canny, J.,” A Computational Approach To Edge Detection”, IEEE Trans. Pattern Analysis and Machine Intelligence, VIII (6), pp. 679–698, 1986.

[17] Irwin Sobel, History and Definition of the Sobel Operator, February 2,2014

[18] McCulloch, Warren; Walter Pitts,"A Logical Calculus of Ideas Immanent in Nervous Activity". Bulletin of Mathematical Biophysics,V (4),pp 233–247, 1943.

[19] Song, Yang, Jian Huang, Ding Zhou, Hongyuan Zha, and C. Lee Giles. "Iknn: Informative k-nearest neighbor pattern classification." In Knowledge Discovery in Databases, Springer Berlin Heidelberg, PKDD,pp. 248-264,2007 .

[20] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." In Proceedings of the fifth annual workshop on Computational learning theory, ACM, pp. 144-152,1992.

[21] Sinhala Alphabet. (n.d.). Wikipedia.,[Online], Available:http://en.wikipedia.org/wiki/Sinhala_alphabet. [Accessed: June. 17, 2016.]