

Machine translation system to convert Sinhala and English Braille documents into voice

K. Sasindu Anuradha*

Faculty of Graduate Studies & Research

Sri Lanka Institute of Information Technology, Sri Lanka

sasindu071@gmail.com

Samantha Thelijjagoda

SLIIT Business School

Sri Lanka Institute of Information Technology, Sri Lanka

samantha.t@slit.lk

Abstract: Reading Braille documents are a time-consuming and labor-intensive task. A blind person should touch every Braille letter by his or her fingers. Therefore, high sensitivity in fingers and memorizing every Braille letters are key factors in Braille reading. Due to enhancement in technology, several Optical Character Recognition (OCR) systems have been introduced for different languages in different parts of the world. However, in Sri Lanka, there are no systems that extract Sinhala or English Braille characters using OCR and convert those Braille codes to sound output. The main purpose of the research is to create a system that extracts both Sinhala and English Braille segments from a given Braille document and makes the Braille content into voice. At the beginning Embossed Sinhala or English Braille image which took from webcam or a high-resolution phone will use for image processing techniques such as gray scaling, thresholding, erosion, and dilation. Erosion and gray scaling help to eliminate the noise and the noise-free image used to detect contour. After pre-processing, the image using an OpenCV library do the segmentation and character extraction. Braille character recognition has done by taking binary to decimal equivalent numbers. Before generating voice output, recognized Braille letters should convert to corresponding language letters (either Sinhala letter or English) and mapping English letters can directly generate English words but in Sinhala need an additional step called Unicode Mapping to generate a Sinhala word. When a Braille document is in high quality, the system will reach for 95 percent accuracy level and generally, results have around 85.30 percent success rate. This software provides many usability characteristics to increase simplicity when it's using OpenCV and related technologies. Even teachers can use this to improve their teaching terminologies and explain things more clearly in their lessons.

Keywords: Braille letters, Image processing, OCR, Segmentation, Unicode mapping

I. INTRODUCTION

Information Technology is everywhere. In the last decade, IT has permeate to every corner in society. In this research, the study concentrates on visual impairments and investigates the possibility to support this community with modern information technology while developing an application. Visual impairment is a significant health problem in Worldwide. Blindness is the condition of lacking visual perception due to physiological or neurological factors.

There are a lot of visually impaired people in our community. These people face difficulties with their regular actions such as reading, walking, driving, socializing and writing. In this modern world, blind people spend more time when they are reading Braille documents by hand and

especially for researchers it is a huge disadvantage. This application uses image processing algorithms to take the Braille document (.JPG, .png or .Pdf) as input and Convert those embedded Braille letters to voice would be a great help to them in their education as well as day to day lives.

II. LITERATURE REVIEW

A French educator named Louis Braille developed a system for blind people to read and write which was known as the Braille system. Braille Coding System is a method that is widely used by visually impaired people to read and write. This system uses patterns of raised dots to make the characters on paper, helping visually impaired people read and write with single touch instead of vision. It is a way for blind people to actively participate in social activities and literature. Braille system was first developed in the nineteenth century. Its characters are six-dot cells, with two columns and 3 rows. Any of the dots may be raised above the paper plain thus giving 64 possible characters. Although Braille cells are used worldwide, the meaning of each of the 64 cells depends on the language that they are being used to depict. The concept of Braille was universally adopted [1].

There is an article called "Braille Translation" written by 3 authors, Farhan Bodale, Uddhav Bhide, and Dilip Gore [2]. Braille Characters are extracted from the given text message. This extracted Braille Character is compared to entries and the system returns the related patterns. However, visually impaired people a little bit hard to use the suggested system as it is. It is necessary to convert the Braille code in any local language so that blind people can hear the audio chaos without the effort of reading with touch sensations. In the article entitled "Development of a Bangla Character Recognition (BCR) System for Generation of Bengali Text from Braille Notation" [3], the development of Bangla letters of the Braille code is achieved. The characters go through a translation engine that consists of a suitable processing algorithm to convert into Bangla Character and any local language extraction of Braille script can be possible. According to the language rule table, the input Braille segment will convert into Bangla characters. The concept of translation rule table has been described by Alasdair King in his paper titled "Text and Braille Computer Translation". Considering these pre-existing methods, the paper called "Braille Code Conversion to Voice in Malayalam" by Bindhu K Rajan and Anjitha V [4] describes the Malayalam language extraction from Braille characters and then the message could be read as audio using proper conversion from text to speech. Their database is made for the standard representation of alphabets. A scanned image will be given

as system input and it undergoes several steps such as image enhancement, Image Filtering, Image segmentation, Braille character extraction and finally Translation.

The research paper "Braille Character Recognition Using Find Contour and Artificial Neural Network by Joko Subur, Tri Arief Sardjono and Ronny Mardiyanto uses Open Cv library for image processing and artificial neural network to identify braille letters. This system supports only the English language and it generates plain text from the given English Braille document [5]. The paper "Optical Braille Translator for Sinhala Braille System: Paper Communication Tool Between Vision Impaired and Sighted and Sighted Persons" by Talagalage Don Sanka Harshana Perera and W.K.I.L Wanniarachchi again use OpenCV library to image processing and binary equal decimal value mechanism to generate text output from given braille document. This system supports Sinhala language and directly convert Braille letters to Sinhala letters [6]. In this research analysis, there is no preprocessing technique to generate words from Sinhala letters and as a result of this output, the text is hard to understand. "Sinhala Braille Translator" by N.M.T De Silva and S.R. Liyanage is another research analysis that uses OpenCV library to image processing [7]. Like previous research, they use binary equal decimal value mechanisms to generate text output from a given braille document and use a preprocessing technique called "Normalization" before generating words from letters. As a result of this additional step output text which is generated by giving Sinhala, Braille image is more meaningful.

III. METHODOLOGY

The system consists of mainly 3 main steps. In the beginning, Braille image should be processed in the way it is convenient to read and it is done by using methods in OpenCV library, which is open source and freely available on their official web site. Then the image is read by the system and uses several mathematical formulas to extract and identification of Braille characters. The final step is generating audio input using extracted Braille letters. It has several sub-steps such as match six points composed Braille

letters to corresponding Sinhala or English letters and generates words. English words can directly map to sentences, but in Sinhala need an additional step called normalization before generating Sinhala sentences. In English, there are several libraries to convert English sentences to voice, but in Sinhala needs a manual process to voice generation.

A. Capturing Braille images

There are several ways to capture a Braille image and this system highly recommended a scanned image.

- Scan a Braille image
- Use Webcam
- A high-resolution image which is taken by mobile phone

When using a web camera to better accuracy in Braille reorganization needs 650 pixels by 500 images with the distance between web camera and the Braille image should be +/- 15cm. When taking images using a mobile phone, the phone should be flatted and the distance between the image and the camera should be less than 6cm.

B. Image Processing

With the original Braille image, it is hard to identify Braille letters so there are compulsory image processing steps. The main idea underneath image processing is a clear recognition of Braille letters for the next process [5].

C. Image processing using OpenCV

OpenCV is called an Open Source Computer Vision library feature primarily aimed toward a real-time PC vision. The library is cross-platform and free to be used underneath the ASCII text file BSD license. It is initially written in C++ programming language and its primary interface is in C++, however, it still retains a less comprehensive although in-depth older C interface. Currently, it supports Python, Java and Matlab/Octave. The system uses OpenCV library methods for mainly Object Identification, Segmentation, and recognition [8].

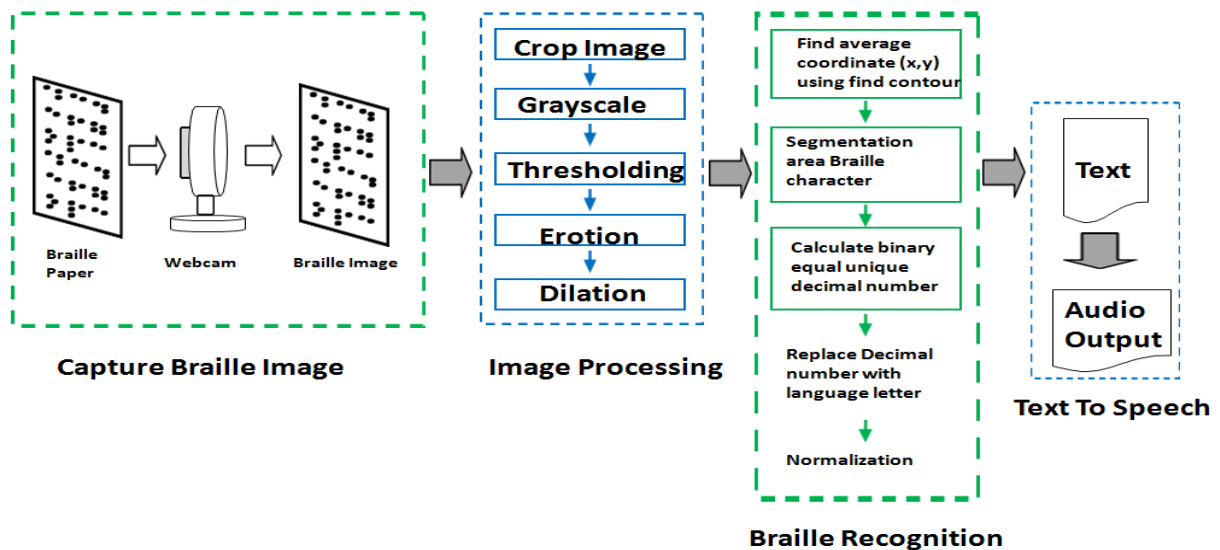


Fig.1.System overview

D. Crop the given Braille image

To achieve better accuracy in Braille reorganization, we need to remove the edges which are not wanted. The entire process is done manually and OpenCV provides below Java method (command) to image cropping [5].

```
Rect rectCrop = new Rect (cropCenterX,  
cropCenterY, cropWidth, cropHeight);
```

CropCenterX and cropCenterY arguments determined the initial point x and y coordinates and cropWidth and cropHeight suggest the cropped image width and height.

E. Gray scaling the image

The original Braille image contains an RGB color scale and the main purpose of this step is to convert the original image RGB color scale into Grayscale images. A grayscale picture is kind of a black and white or gray monochrome; it contains a variety of shades of gray. OpenCV method for the Gray scaling mention bellow and it contains 3 parameters [5].

```
Imgproc.cvtColor(originalImage,imgGrayscale,  
Imgproc.COLOR_BGR2GRAY);
```

The original image represents the giving Braille image and imgGrayscale is the equivalent grayscale image for original image. Imgproc.COLOR_BGR2GRAY is the static attribute that gives a command for OpenCV for gray scaling.

F. Thresholding the image

This process helps to convert grayscale image to binary image; an image that has the color rage of 0-255. Before segmentation thresholding should be done [5].

```
threshold(src, dst, thresh, maxval, type)
```

The original method signature looks like above and it has five parameters. Parameter src represents a math object of an input image and dst represents the result, math image object after the method execution. The thresh variable contains a double value of the threshold and maxval is a double variable that holds the value that to be replaced when the pixel value exceeds the threshold value. The final parameter is an integer character describing the kind of threshold to be practiced.

G. Erosion

In this analysis, the erosion method is applied to reduce white area dots and eliminates noise from the Braille picture that emerged from the thresholding method [5].

```
erode(src, dst, kernel)
```

Similar to previous image processing methods, src denote to a math object of an input image, and dst denote the result math image object after the method execution. The parameter kernel represents a Mat object of the kernel.

H. Dilation

According to this research, the dilation process applied to expand white dots on image Braille.



Fig. 2. Original Scanned Image

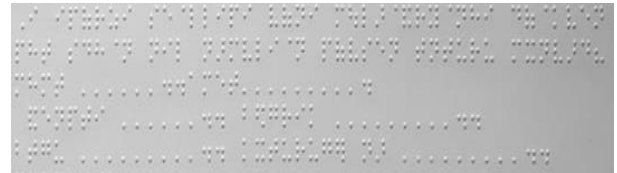


Fig. 3. Output of grayscale method.

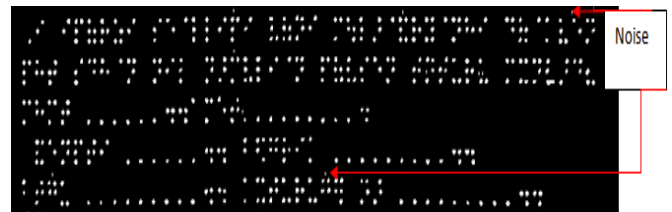


Fig. 4. Output of thresholding method



Fig. 5. Output of erosion method. (Noise removal)

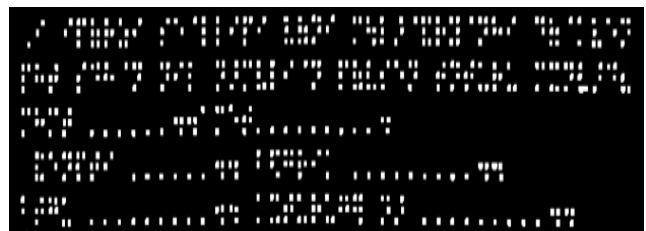


Fig. 6. Output of dilation method. (large Braille dots)

```
dilate (src, dst, kernel)
```

The dilate method signature consists of 3 parameters and src and dst represent the input Mat image object and output Mat image object. The parameter kernel represents a Mat object of the kernel. Below show the results images of image processing steps.

I. Braille reorganization

The next major step is locating each and every center coordinates of Braille dots. It also has several steps. Normally Braille dot covers a circular area and OpenCV provides a method to find position coordinates of each dot.

findContours(Mat image, List<MatOfPoint> contours, Mat hierarchy, int mode, int method, Point offset)

Above method uses the algorithm called CITE: Suzuki85 for retrieves contours from a binary image. The contours are a valuable tool for shape review and object detection and identification. FindCountours method has 6 parameters; it is a static method in Imgproc class. Image represents the output image of image processing step, contours is a list of MatOfPoint which saved detected contours as points of vectors, hierarchy includes information regarding the image topology, mode determines the Contour retrieval tone and, in this research, uses CHAIN_APPROX_SIMPLE as retrieval tone and finally offset is an optional parameter which helpful if the contours are obtained from the image ROI.

J. Mechanism of findContours method

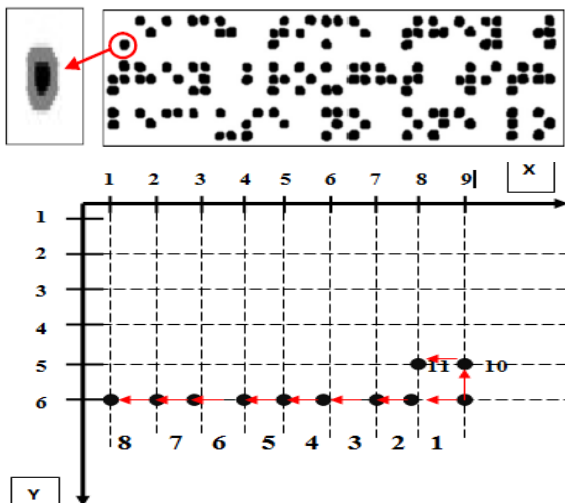


Fig. 7. Dot identification by findContours method

FindContours method identifies Braille dots from the rightest bottom corner and traverses through the x-axis up to the top in the way shown in figure x. Every black dot point consists of several members and those members can be identified from this process. All members have unique X and Y coordinates and it will help to find the center point of Braille dots. If members of a Braille point printed out it would look like bellow.

- 1) X ->: 180.0 Y -> 276.0
- 2) X ->: 180.0 Y -> 277.0
- 3) X ->: 181.0 Y -> 277.0
- 4) X ->: 181.0 Y -> 276.0

Member count: 4

K. Center X and Y Coordinates of a Braille Dot

By analyzing the above information, it can be recognized that the white dot is one point there were four or three members, or it can be concluded that there are four to three white pixels. Therefore, the need to calculate average XY coordinates of every member of a white dot to determine the center point of a Braille dot.

After the pixel coordinate data of every member of the white dot concerned need to find minimum X and Y coordinates as well as maximum X and Y coordinates. For that following formulas can be used. [5]

$$\text{Center X} = \frac{(\text{Minimum X} + \text{Maximum X})}{2} \quad \text{--- 01}$$

$$\text{Center Y} = \frac{(\text{Minimum Y} + \text{Maximum Y})}{2} \quad \text{--- 02}$$

L. Find the average middle of rows and columns

After finding the center XY coordinate values of all the Braille dots next step is determining the average middle values in each and every column and row. The main reason for calculating the average middle values of each column and row is column to column and row to row X and Y coordinate values change significantly. Therefore, results have better accuracy in terms of Braille identification. There is a mathematical formula to calculate average X and Y coordinates and it mentioned bellow.

AvgXk stands for average x coordinates for each column, Centerxn denotes for summation of all centers X coordinates for each column, k denotes for column number and n is the number of dots in a column. Therefore, the 3rd equation obtained average X coordinates in each column.

$$\text{AvgXk} = \frac{\sum_{k=1}^n \text{CenterXn}}{n} \quad \text{--- 03}$$

$$\text{AvgYb} = \frac{\sum_{b=1}^n \text{CenterYn}}{n} \quad \text{--- 04}$$

AvgYb stands for average y coordinates for each row, CenterYndenotes for summation of all centers Y coordinates for each row, b denotes for the row number and n is the number of dots in a row. Therefore, the equation number 4 obtained average Y coordinates in each row. [5]

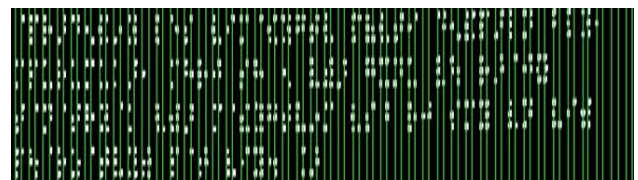


Fig. 8. Average column indexes

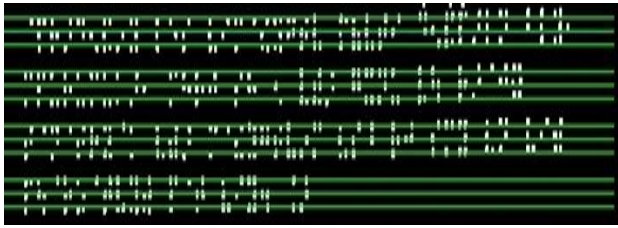


Fig. 9. Average row indexes

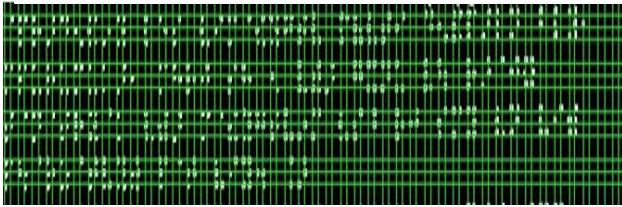


Fig. 10. Average row and column indexes

M. Braille character segmentation

The segmentation scenario is the most significant step in this research project because Braille characters have 2 columns and 3 rows. When determining the area of Braille of 2 columns and 3 rows based on average X and Y coordinates the Segmentation process comes into the action. There is a mathematical equation for the Braille characters segmentation process, and it is mentioned below. [5]

$$S = \sum_{b=1}^b \sum_{k=1}^k [t = t + 1] \left(\sum_{h=b}^{b+3} \sum_{d=k}^{k+2} ([avgXd][avgYh]) \right)$$

05

The letter S determined the area of each Braille letter and b and k stands for row index and column index, respectively. The letter t stands for the value counter of a Braille figure, the letter h represents the counter of 3 rows in every Braille sign and d means 2 column counters. The variables avgX and avgY denotes for average X coordinate value and average Y coordinate value respectively. Fig 11 highlights the outcome of the segmentation method and the segmented region of one Braille letter can already be labeled with a blue border. After the segmentation method, every segment area managed to do a letter identification process.

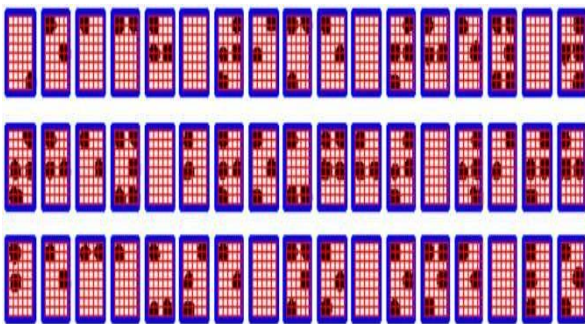


Fig. 11. Braille letter segmentation

N. Braille character identification

The next major step is Braille dot identification in the segmentation areas and consists of several steps. In this research analysis, the average X and Y coordinates are considered and need further Braille character verification process as an additional step. According to Fig 12 Braille dot not situated in the exact average X and Y coordinate value. It may shift to 1-pixel right, left, top or bottom and need to check whether adjacent pixels have a white color pixel. According to this research check one pixel in major 4 directions (top, bottom, left and right) whether it contains RGB value of the white color [9].

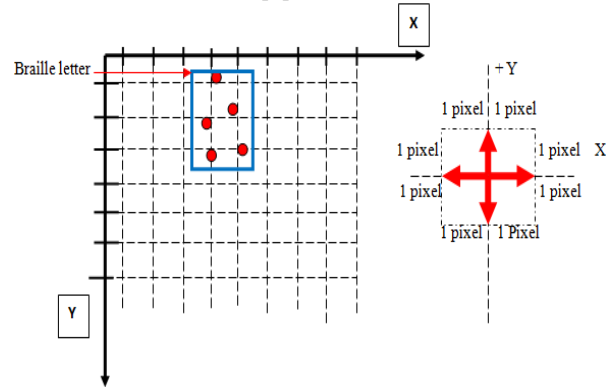


Fig. 12. Pixel Loop

There is a method in OpenCV to get the RGB value of a pixel which is in given X and Y coordinates and it looks like below.

getRGB(int x, int y)

The letter x and y denote the average x and y coordinates respectively, and it returns 32 bits integer value. Normally pixel is considered as the smallest unit in an image. Braille dots appear in white and use pixel loop to check whether the adjacent pixel has white color RGB values or not. If it has a white color RGB values, it registered as a Braille dot otherwise omit it.

O. Calculate Binary equal decimal values

A Binary letter has 6 points and the way those points presences or absent generate a range of mixtures. Every Braille letter has a unique 6-bit binary number and after getting the 6-bit binary number, then convert it to a decimal number. For that, there is an arithmetic operation to generate the binary equal decimal number. According to fig 13 6-point notation for single Braille, the letter can create 63 different combinations and all most all the languages' alphabets in the world have less than 63 letters.

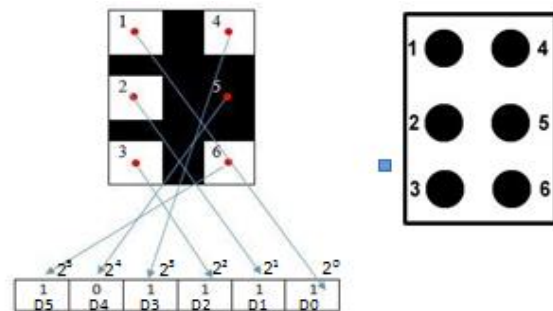


Fig. 13. Braille letter equals 6-bit Binary value

$$\begin{aligned}
 \text{Decimal number} &= (D_0 \cdot 2^0) + (D_1 \cdot 2^1) + (D_2 \cdot 2^2) + \\
 &\quad (D_3 \cdot 2^3) + (D_4 \cdot 2^4) + (D_5 \cdot 2^5) \\
 &= (D_0 \cdot 1) + (D_1 \cdot 2) + (D_2 \cdot 4) + \\
 &\quad (D_3 \cdot 8) + (D_4 \cdot 16) + (D_5 \cdot 32) \\
 &= (1 \cdot 1) + (1 \cdot 2) + (1 \cdot 4) + (1 \cdot 8) + \\
 &\quad (0 \cdot 16) + (1 \cdot 32) \\
 &= 1 + 2 + 4 + 8 + 0 + 32 \\
 &= 32
 \end{aligned}$$

Therefore 6 points for the Braille letter mechanism can apply all the languages in the world and it has universally accepted. In this research project mainly focus on English and Sinhala Braille reorganization [6].

TABLE I. NUMERIC BINARY EQUAL DECIMAL VALUES

Braille Notation	Binary Equal Decimal Value	Number
·	01	01
⠆	03	02
⠃	09	03
⠔	25	04
⠨	17	05
⠠	11	06
⠤	27	07
⠠	19	08
⠨	10	09
⠔	26	00

TABLE II. SINHALA CONSTANT LETTERS BINARY EQUAL DECIMAL VALUES

Braille Notation	Binary Equal Decimal Value	Sinhala Letter
·	01	අ
⠨	28	ආ
⠤	55	ඇ
⠠	59	ඈ
⠨	10	ඉ
⠨	20	ඊ
⠠	37	උ
⠤	51	ඌ
⠨	17	එ
⠨	34	ඒ

Braille numbers and some English Braille letters have the same Braille symbols and there is another mechanism to distinguish English Braille and numeric Braille. Braille has a special indicator for numbers, it shows in fig 14. When it is in the Braille document it means that on that point onwards all letters have numeric values until finding a space or letter indicator symbols which show in fig 15 [10].

TABLE III. ENGLISH BINARY EQUAL DECIMAL VALUES

Braille Notation	Binary Equal Decimal Value	Number
·	01	A
⠆	03	B
⠃	09	C
⠔	25	D
⠨	17	E
⠠	11	F
⠤	27	G
⠠	19	H
⠨	10	I
⠔	26	J
⠆	05	K
⠆	07	L
⠠	13	M
⠔	29	N
⠨	21	O
⠤	15	P
⠤	31	Q
⠤	23	R
⠨	14	S
⠔	30	T
⠠	37	U
⠨	39	V
⠤	58	W
⠠	45	X
⠔	61	Y
⠔	53	Z



Fig. 14. Numeric indicator



Fig. 15. Letter indicator

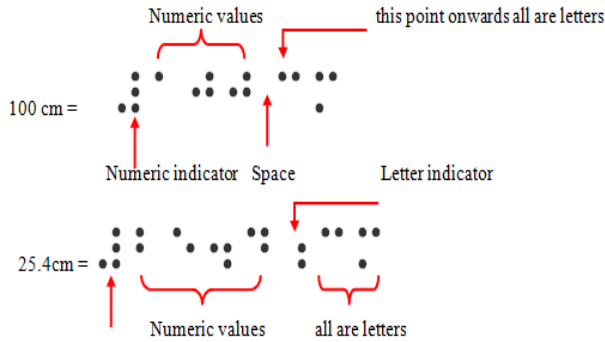


Fig. 16. Example

P. Sinhala text normalization

This step only needs for when converting Sinhala Braille document to voice. For English Braille does not need normalization because from decimal values can create words, but in Sinhala after generating Sinhala letters from decimal values need an additional step to create words. Eg1 shows a text before normalization and can't get a proper meaning at once. Eg2 shows a text after normalization and it can easily understand. In practically Eg1 shows the text a blind person initially identifies when reading a braille document and from his brain, he normalizes it into a more readable format which shows in Eg2. Therefore, normalization is a compulsory step in this research analysis and needs to maintain normalization tables for normalizing [7].

TABLE IV. SINHALA TEXT NORMALIZATION

Letter 01	Letter 02	Normalized Letter
ක	ආ	කා
ක	ඇ	කෑ
ක	ඈ	කූ
ක	ඉ	කී
ක	ඊ	කී
ක	උ	කු
ක	ඌ	කු
ක	එ	කෙ
ක	ඒ	කේ
ක	ඔ	කො
ක	ඕ	කෝ

ක	ඔ	කො
ක	ක	කී
ක	උ	කා
ක	ඌ	කාා

Normal text:

Eg1: පර ආද ඒ ශ රී ය ල එස වලුමටයගන ව
 පඤ්ච න ල ද ඒ ශ ප ආ ල න බ ල ය

After normalization

Eg2: පරාදේශීය ලෙස විමධියගනව පැවති
 දේශපාලන බලය

Q. Sound outputs

For English Braille, there are open source libraries to convert English text to voice and in this research analysis, the library called Freetts used for text to voice translation. For Sinhala it is a manual process and similar to fig 17 all audio outputs save in one folder. All the paths of the sound clips save in a HashMap and when converting Sinhala text to sound read one letter at a time and match with Hashmap. Then find the sound clip address corresponding to Sinhala letter and play it.

```
public static HashMap<String, String> sinhalaSounds=new HashMap<String, String>();
sinhalaSounds.put("ක", "/src/main/resources/static/audios/ki.mp3");
sinhalaSounds.put("කී", "/src/main/resources/static/audios/kii.mp3");
sinhalaSounds.put("ආ", "/src/main/resources/static/audios/a.mp3");
Sinhala Letter
Sinhala Letter Path
```

Fig. 17. Sound clips and their path in a HashMap

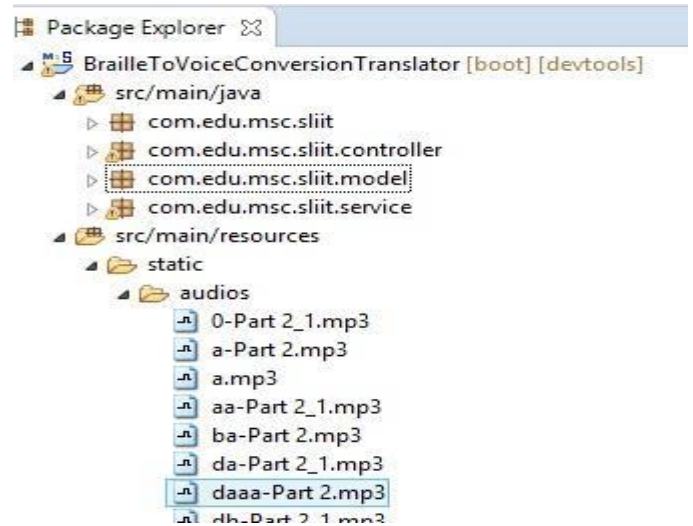


Fig. 18. Sound folder

IV. RESULTS AND DISCUSSION

The system has one single interface and the project is deployed as a web application. The outcomes of the system can be explained using an illustration and fig 19 shows the only interface of the system. There is a Choose File option to upload a Braille image and uploaded a Braille image display like in fig 20. The next step is to scan the image and when the client clicks the Scan Image button system generates a text document corresponding to the Braille image as an intermediate outcome.

Binary equal decimal values

[56, 50, 0, 0, 25, 55, 29, 62, 0, 19, 13, 37, 0, 39, 20, 0, 30, 10, 3, 17, 29, 0, 14, 28, 46, 5, 0, 1, 29, 37, 39, 0, 13, 17, 23, 62, 0, 25, 10, 61, 37, 53, 37, 0]

[5, 16, 23, 41, 10, 5, 28, 8, 23, 13, 10, 5, 0, 14, 13, 28, 26, 61, 5, 0, 28, 23, 8, 13, 24, 61, 0, 1, 39, 13, 0, 39, 47, 61, 17, 8, 29, 0, 8, 5, 23, 10, 50, 0]

[15, 51, 50, 0, 60, 3, 25, 26, 26, 0, 15, 13, 53, 39, 8, 30, 0, 59, 30, 62, 0, 25, 10, 39, 17, 61, 10, 50, 0, 17, 13, 0, 61, 37, 27, 61, 34, 0, 14, 10, 62, 0, 13, 0]

Replacing those decimal values with letters

[ළ, ., ., ., ද, ඇ න, ට, , හ, ම, උ, , ව, ඊ, , න, ඉ, බ, ඒ, න, , ස, ආ, බ, ක, , අ, න, උ, ව, , ම, ඒ, ර, ට, , ද, ඉ, ය, උ, ණ, උ,][ක, 16, ර, ෂ, ඉ, ක, ආ, p, ර, ම, ඉ, ක, , ස, ම, ආ, ජ, ය, ක, , ආ, ර, p, ම, හ, ය, , අ, ව, ම, , ව, ශ, ය, ඒ, p, න, , p, ක, ර, ඉ, .,][ප, උ, ., , null, 2, 4, 0, 0, , ප, ම, ණ, ව, p, න, , ඇ න, ට, , ද, ඉ, ව, ඒ, ය, ඉ, ., , ඒ, ම, , ය, උ, ග, ය, ඒ, , ස, ඉ, ට, , ම,]

Normalization

දැනට හමු වී
 තිබෙන සාධක
 අනුව දමරට
 දියුණු
 කෘෂිකාර්මික
 සමාජයක
 ආරම්භය අවම
 වශයෙන් කිරී.
 යු. 2400
 පමණවත්
 ඇතට දිවයි.
 එම යුගයේ සිට
 ම

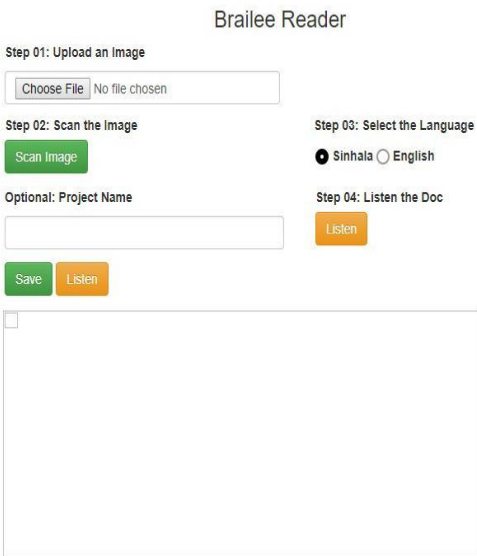


Fig. 19. System interface

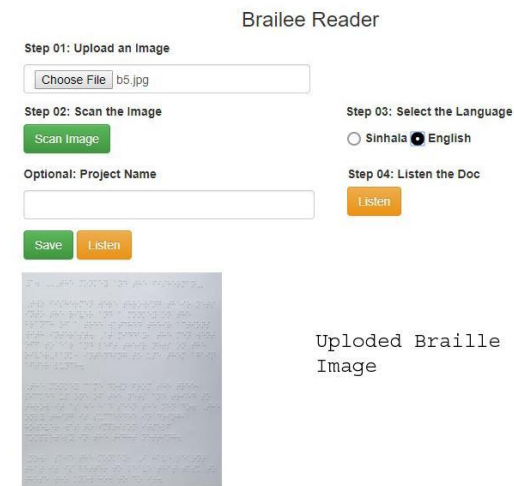


Fig. 20. Uploaded Braille Image



Fig. 21. Original Braille Image



Fig. 22. Output of Image Processing

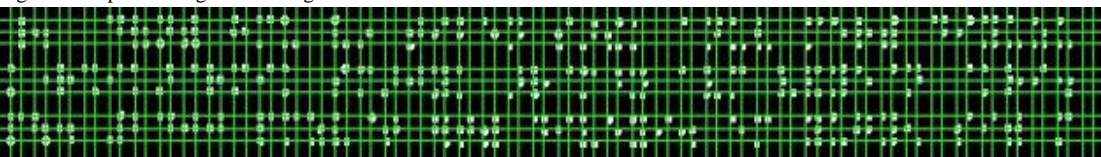


Fig. 23. Find Average X and Y Coordinates.

Results which are generated by the system can be classified as below. It was subjected to identification of 6 braille documents and 3 of them are Sinhala and the rest of them is English. Each Braille paper has a standard size of 8.5" x 11 printed on a single side. Three different English short stories have been embossed in lines on three English Braille papers and the three Sinhala Braille papers have been embossed with 3 pages in Grade 11 Social Studies book. The Sinhala letters such as “ඔළු, ඔඹ, ඔ, ඔ, ඡ, ඤ, ධ, ඩ, ද, ඟ” seldom appeared in Sinhala documents and thus such letters could not be captured in test samples.

TABLE V. LETTERS IDENTIFICATION

Letter	No of occurrence	Correct identification	Accuracy Percentage
අ	35	29	82.85
ආ	90	81	90.0
ඇ	41	37	90.24
ඈ	08	06	75.0
ඉ	162	140	86.41
ඊ	20	16	80.0
උ	76	68	89.47
ඌ	08	07	87.5
ඍ	78	66	84.61
ඎ	41	36	87.80
ඔළු	0	0	0.0
ඔඹ	27	23	85.18
ඔ	10	08	80.0
ඔඹ	0	0	0.0
ක	89	80	89.88
ඛ	5	04	80.0
ග	42	35	83.33
ඝ	2	01	50
ඞ	0	0	0.0
ඟ	0	0	0.0
ච	1	1	100.0
ඡ	0	0	0.0
ඣ	7	6	85.71
කඩ	1	1	100.0
ඤ	0	0	0.0
ට	42	35	83.33

ධ	0	0	0.0
ඳ	16	13	81.25
ආ	0	0	0.0
ඈ	14	12	85.71
ඊ	42	37	88.09
උ	0	0	0.0
ඌ	64	57	89.06
ඍ	10	8	80.00
ඎ	152	135	88.81
ඏ	0	0	0.0
ඐ	34	26	76.47
එ	01	01	100.0
ඒ	25	20	80.0
උ	02	01	50.0
ඌ	102	79	77.45
ඍ	95	84	88.42
ඎ	78	69	88.46
ඏ	53	47	88.67
ඐ	106	88	83.01
එ	19	14	73.68
ඒ	04	03	75.0
උ	75	66	88.0
ඌ	56	51	91.07
ඍ	08	06	75.0
ඎ	0	0	0.0
A	145	138	95.17
B	23	20	86.95
C	39	32	82.05
D	79	70	88.6
E	245	201	82.04
F	33	29	87.84
G	32	29	90.62
H	125	103	82.4
I	87	75	86.20
J	07	06	85.71
K	20	16	80.0
L	60	47	78.33

M	36	32	88.88
N	104	83	79.80
O	105	85	80.95
P	24	20	83.33
Q	02	02	100.0
R	95	82	86.31
S	107	85	79.43
T	162	135	83.33
U	43	35	81.39
V	18	14	77.77
W	42	35	83.33
X	04	03	75.0
Y	42	35	83.33
Z	02	01	50.0

TABLE VI. NUMERIC IDENTIFICATION

Number	No of occurrence	Correct identification	Accuracy
01	3	3	100.0
02	1	1	100.0
03	3	3	100.0
04	0	0	0.0
05	0	0	0.0
07	2	1	50.0
08	2	1	50.0
09	3	2	66.66

Convert Sinhala and English Brille document is a new research area and very little research has conducted so far. The main technologies used to detect Braille letters are open source as well as inexpensive. The system is tested by different sizes and different qualities of Sinhala and English braille documents and scored more than 85 percent accuracy. Conversion of identified braille letters to corresponding voice is straightforward and in Sinhala play letter by letter but in English, it plays words at once. The quality of the image is the most significant factor because it affects Braille identification and when the image is in high-quality accuracy will reach up to 95 percent. If the braille document is old and heavily used it would reduce the accuracy and the system accept single side braille printed documents.

However, hardware components utilized by the system is less and no need for expensive and complicated devices. The system runs on top of any platform and tested in different browsers and different operating systems.

V. CONCLUSION

This system is capable of extracting Braille letters from Braille image and converts it into the corresponding language voice. Initially, the process of extracting Braille letters is associated with several image processing steps such as Crop, Grayscale, Thresholding, Erosion, Dilation, and segmentation. Then extracted Braille letters are decoded into Sinhala characters and then through the normalization process reorder the Sinhala characters into legible Sinhala text. Finally, legible Sinhala text is converted into sounds like the final outcome. In the case of the English Braille image, the normalization process is not applied and directly converts to legible English text.

The strategy is a successful, cost-effective, quick and straightforward method to recognize dots in Braille papers. It does not require costly or complex hardware. The current system gains around 85.30 percent accuracy level in both languages and even teachers can use this to improve their teaching terminologies and explain things more clearly in their lessons.

REFERENCES

- [1] Braille Alphabet, [online] Available at: <http://www.who.int/countries/lka/en>. [Accesses: 02.06.2019].
- [2] Farhan Bodale, Uddhav Bhide, Dilip Gore, "Braille Translation". International Journal of Research in Advent Technology, E-ISSN:2321- 9637 Vol.2, No.4, April 2014.
- [3] 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) ISSN: 2278-3075, Vol- 3, Issue-1, June 2013.
- [4] Bindhu K. Rajan and Anjitha V., "Braille Code Conversion to Voice in Malayalam", International Conference on Communication and Signal Processing, April 6-8, 2017, India
- [5] Joko Subur, Tri Arief Sardjono, Ronny Mardiyanto "Braille Character Recognition Using Find Contour and Artificial Neural Network", JAVA, International Journal of Electrical and Electronics Engineering 19, Vol.14, Number 1, April 2016.
- [6] T. D. S. H. Perera and W. K. I. L. Wanniarachchi "Optical Braille Translator for Sinhala Braille System: Paper Communication Tool between Vision Impaired and Sighted Person", The International Journal of Multimedia & Its Applications (IJMA), Vol.10, No.1/2/3, June 2018.
- [7] N.M.T De Silva and S.R Liyanage, "Sinhala Braille Translator", International Journal of Trend in Research and Development, Vol.3(4), ISSN 2394-9333www.ijtrd.com.
- [8] Open CV Documentation, [online] Available at: <https://docs.opencv.org/2.4.13.7/modules/core/doc/intro.html>. [Accesses: 02.06.2019]
- [9] How to get and set pixel value in Java, [online] Available at: <https://www.dyclassroom.com/image-processing-project/how-to-get-and-set-pixel-value-in-java> [Accesses: 08.06.2019]
- [10] The Braille Alphabet, [online] Available at: <https://www.pharmabraille.com/pharmaceutical-braille/the-braille-alphabet> [Accesses: 09.06.2019].