Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 09, September-2019

Smart OCR for Visually Challenged People

Nikunj Desai, Devarsh Bhupatkar
Guide: Prof. Anuja Nair
Institute of Technology, Nirma University,
Ahmedabad, India

***Abstract*:- Braille is a language used by visually challenged people to read and write, it consists of 6 dots which are organised in a 3x2 matrix and hence can have 64 (2^6) symbols. Software technologies nowadays focus primarily on English however for visually challenged people, it is not feasible to type and operate keyboards, nor can they have their text directly scanned and read via computer devices as computer software are not designed to read scanned images in braille and hence special provisions need to be added to computer devices for understanding braille language, decrypt the meaning of the dots and then provide the output in feasible form to the braille user.**

One way to do this would be to take input from the visually challenged user in the form of a scanned image in braille, obtain the precise position of the dots in the 3x2 matrix, decrypt the dots based on the rules of braille and then use a text-to-speech software to give the output of the computer back to the user.

1. INTRODUCTION

Braille is an integral part of communication for the visually challenged and to accomplish the goal of communicating with the computer via braille language can be done in 4 steps which would be divided as pre-processing, segmentation (horizontal and vertical), error correction and the conversion of the output file from text to speech but before that it is necessary to understand the working of the braille language, the 3x2 matrix of braille gives a different character depending on the position of dots where the absence of dot is taken as 0 while its presence is taken as 1. Now as all 6 dots have 2 possibilities (0 or 1) and hence there will be 2^6 possibilities of characters. In English it includes 26 English alphabets, punctuations and numbers, etc.

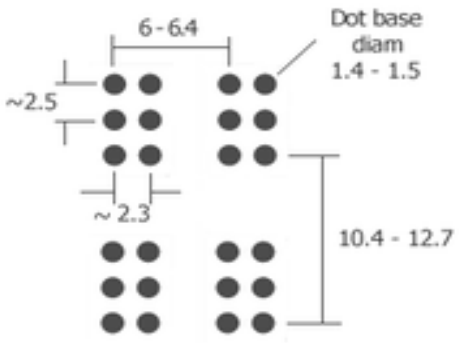


Fig 1: Standard spacing followed in braille document

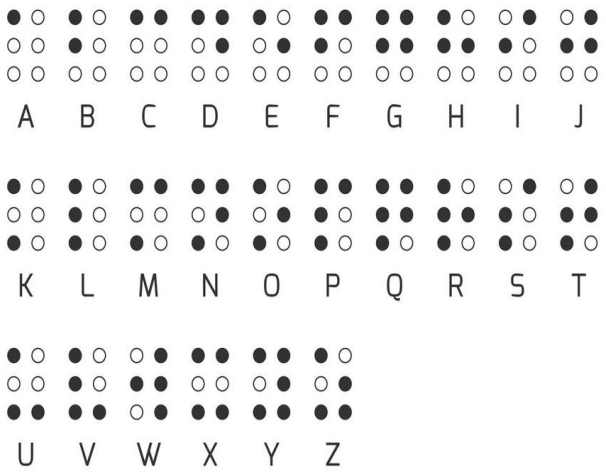


Fig 2: Braille character for alphabets

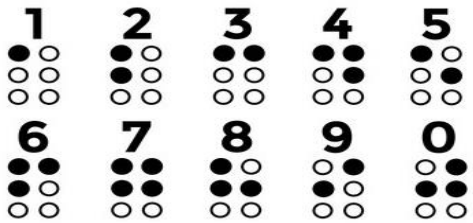


Fig 3: Braille character for integers

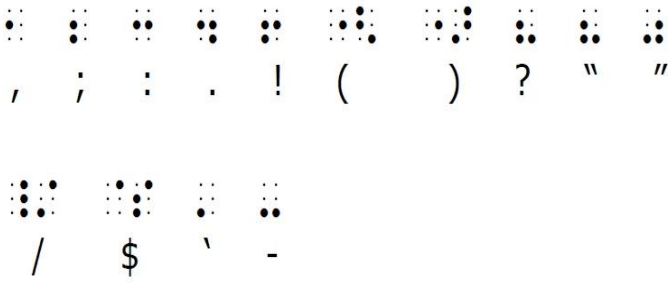


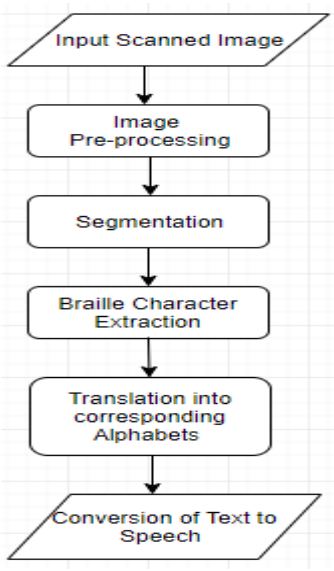
Fig 4: Braille character for punctuations

Once the image is scanned, the intensity range of the image will have to be adjusted to detect the presence of dots which are present in the image. Then the image goes through the process of horizontal and vertical segmentation to detect the edges of the 3x2 matrix and finally after the presence of dots in the matrix is noted, they will be checked against the given symbols for the particular language and the dots would have been converted into their corresponding characters, numerals and punctuations. Even after applying these processes, some errors might have crept in due to failure of detection of a few dots or because white noise in the image is interpreted as a dot and so the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 09, September-2019

error correction step tries to identify such errors to eradicate problems in the output. Finally, the text generated will be written in an output file which will be converted to speech using the text-to-speech API and hence the process will be complete.

2. BLOCK DIAGRAM



3. PRE-PROCESSING

3.1 Image Enhancement

For the detection of dots, there needs to be reduction of noise and enhancement of dots and only then cam efficient readings be found. Initially the image could be in any shade as shown in Figure-5(which will be used as the sample input in the process) but the image needs to be converted into grey scale where dots will appear in a darker shade as compared to the black background and a histogram will have to be used to identify these intensity ranges and they have to be enhanced to identify the dots. Contrast stretching and intensity sketching are two piece-wise enhancement techniques which are used to enhance the dots and are represented by

$$S = T(r)\ldots[2]$$

Where S is the grey scale after modification, r is the grey scale before enhancement and T is the enhancement function.

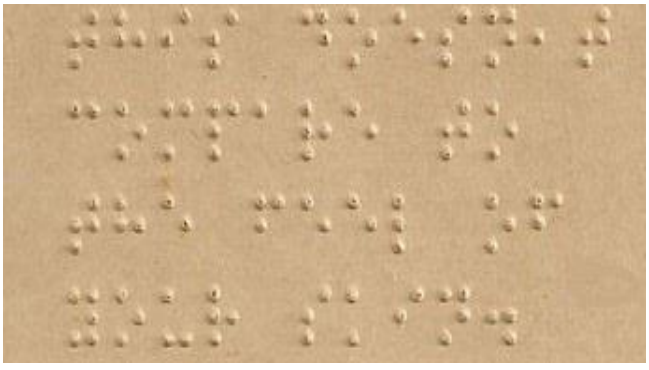


Fig 5: Original scanned braille document

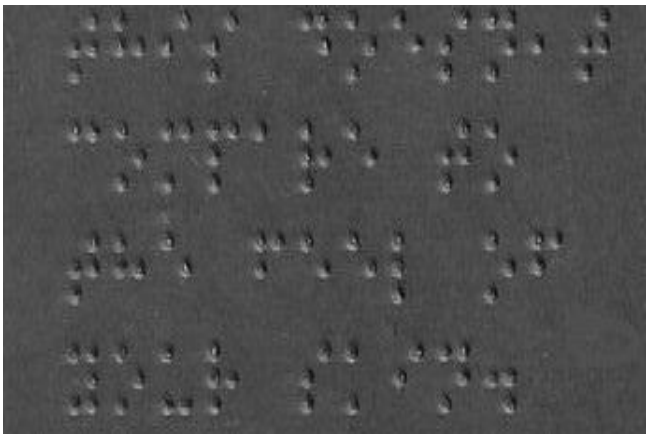


Fig 6: After applying enhancement process

After the image is enhanced, normalized box filter method is used which uses a 5x5 matrix that blurs the image after convolution and removes the noise from image as shown in figure 7.

$$K = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} *A\ldots[4]$$

Where * denotes convolution, A is the grey scale image, K is the final blurred and noiseless image.

Fig 7: Blurred noiseless image

3.2 Erosion

Normalised box filtration method is further used where the first step involves cutting off noise with the help of erosion and the next step involves the process of dilation which increases the size of dots which have been detected in the image. In the step of erosion, the image is divided into smaller segments called kernels action areas and they are convoluted with convolution matrix which is also known as kernel or mask matrix.[3], this step works to lessen the boundaries of the foreground so that the pixels near the boundary are discarded depending solely on the size of the kernel used as shown in Figure-8.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 09, September-2019

$$E(erosion) = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} * A \quad \ldots[4]$$

Fig 8: Image after erosion process

So, a pixel in the original image is considered 1 if and only if the pixels around the boundary of the pixel are 1 otherwise the pixels are simply eroded, i.e., simply set to 0 and in this way the small white dots which depict noise are removed and the image is ready for the next step of dilation.

3.3 Dilation

The process of dilation is almost inverse the process of erosion as it works on increasing the size of the dots by making the pixel element 1 even if there is just one-pixel element under the kernel as 1 and since the white noise has already been removed.[3], there won't be any increase of noise in the image even though the area of dots in the image are being increased as shown in Figure-9

$$D(dilation) = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix} * A \quad \ldots[4]$$

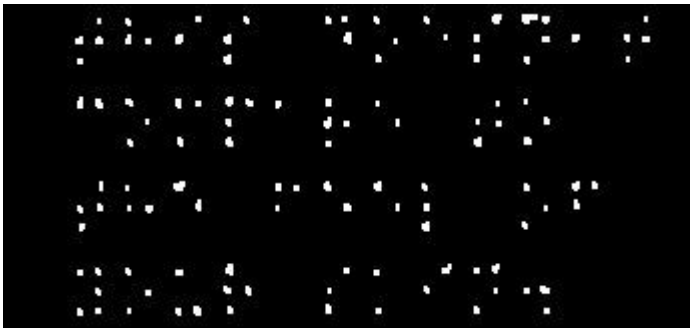


Fig 9: Image after dilation process

Now the image is ready to go through the process of horizontal and vertical segmentation before getting it ready for error correction.

4. SEGMENTATION

4.1 Horizontal segmentation

In horizontal segmentation basic and most useful algorithm is canny edge detection algorithms and following are the steps of the canny edge detection algorithm.[5]

4.1.1 Finding region of interest:-

Moving along the x-axis, dots are to be detected and once a dot is detected, the entire x-axis is changed in colour and in the horizontal direction the entire line is made of high pixels, i.e., the entire line is made of pixels with value 1. If not a single dot is found in the entire line while moving along the x-axis, then that row is left as it was in the original image.

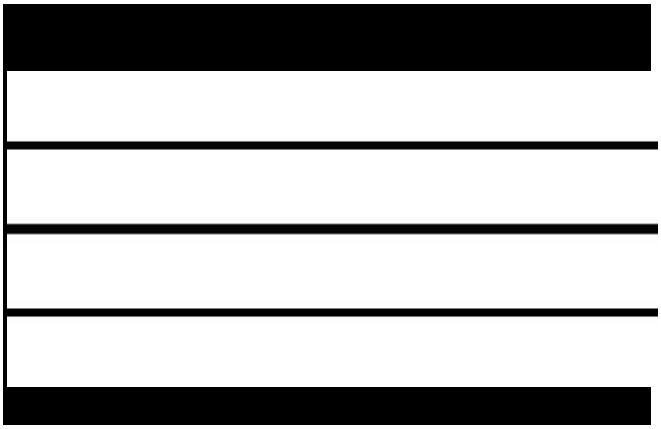


Fig 10: Region of Interest

4.1.2 Finding Gradient:-

To extract information from the image, image gradient needs to be found because image gradient is used to detect edges in the image. For this the Prewitt operator is used which uses two 3X3 kernels which are convolved with image A and the approximations of change in the horizontal direction G_X and G_Y in the vertical direction are found.

$$Gx = \begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} * A \;\&\; Gy = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A \quad \ldots[2]$$

Where * denotes convolution

The resultant gradient approximations are compared to give the resultant gradient which will be given by:

$$Edge\ Gradient\ (G) = \sqrt{Gx^2 + Gy^2} \ldots[2]$$

$$\text{Angle } (\Theta) = \tan^{-1}\left(\frac{Gy}{Gx}\right) \ldots[5]$$

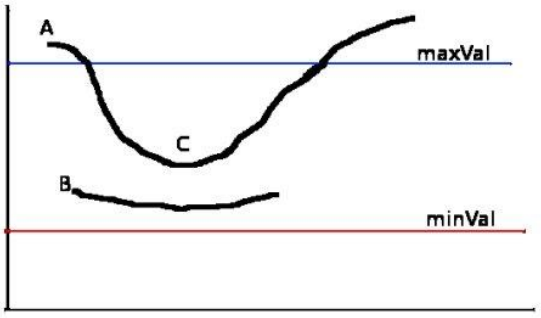
International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 09, September-2019

4.1.3 Non-maximum suppression:-

This method is used to make the edge thinner as the edge found after finding the regions of interest would be blur going in the direction of the gradient. Hence, only the local maximum is taken as the edge while the other neighbouring edges need to be discarded. The process is as follows: According to the value of θ (always 0^0 in the case of detection of braille dots), the edge strength of the current pixel is compared with the edge strength of the pixels in positive and negative directions and if and only if it is greater than its neighbours, it will be used otherwise it will be supressed, i.e., the value of its pixels is set to 0 if it is not local maxima. So the pixels pointing in the x-direction will be compared to the pixel above and below it in the horizontal axis and if it has a greater gradient value, it is accepted and if not, it is supressed.

4.1.4 Hysteresis Thresholding:-

After all the possible edges in the image are found, it is to be found which of them actually have high enough intensity to qualify as an edge. For the explicit purpose of doing it, two threshold intensity values are taken, namely maxVal and minVal. To identify the edges, any edge whose value is equal or greater than the maxVal is qualified as an edge and is called "sure-edge" and any edge with value below the minVal is discarded as an edge. If the value of intensity of the edge is greater than minVal but less than maxVal, the vicinity of the edge needs to be seen, if the given edge is connected to another "sure-edge" then it will be considered as "sure-edge" otherwise it is plainly discarded as shown in figure 11. From this it can be concluded that the process assumes that edges are long lines and because of that the small pixel noises which may have crept in are removed and the final image only has strong edges while the blur edges in its vicinity are removed.



…[5]

Fig 11: Hysteresis Thresholding graph

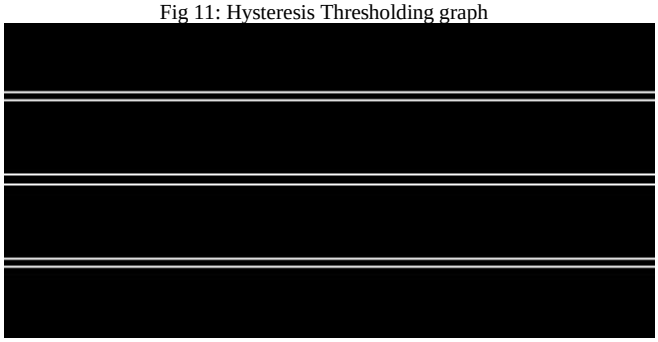


Fig 12: Canny Edges

Then the edges which are found to be "sure-edges" are qualified as Hough lines and the original image is partitioned into smaller images on the basis of the Hough lines where only the regions of interest are selected while the area between the Hough lines which do not contain dots are discarded from the image.

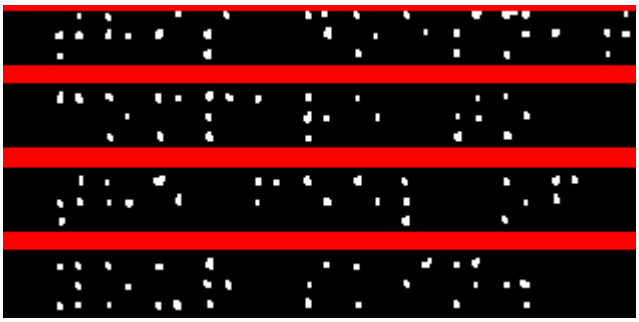


Fig 13: Hough line

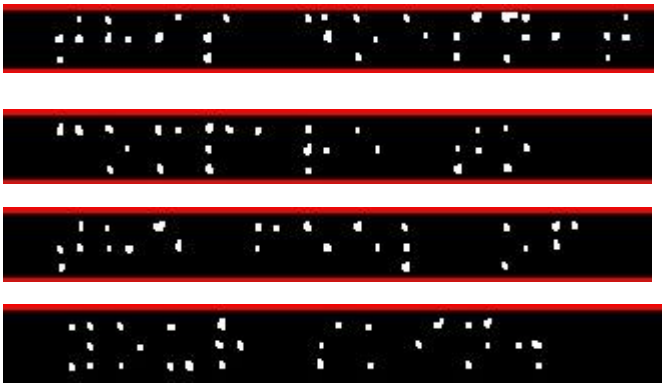


Fig 14: Sample partition of the above Fig 13

4.2 Vertical Segmentation: -

The canny edge algorithm is further used on the partition images which are produced after horizontal segmentation but the algorithm is used in the vertical direction instead of the horizontal direction.

4.2.1 Finding region of interest:- Instead of moving along the x-axis and detecting dots, the partitioned image which is the input image for the case will have vertical regions of interest and vertical blur edges as shown in figure 15

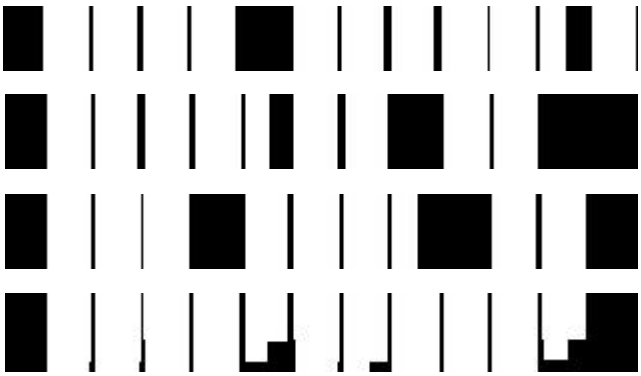


Fig 15: Region of interest

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 09, September-2019

4.2.2 Finding Gradient: - The image gradient directions of G_X and G_Y in the horizontal direction will become G_Y and G_X respectively in the vertical direction. Hence G_Y and G_X would be given by:

$$Gx = \begin{bmatrix} +1 & +1 & +1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A \;\&\; Gy = \begin{bmatrix} +1 & 0 & -1 \\ +1 & 0 & -1 \\ +1 & 0 & -1 \end{bmatrix} * A \quad [5]$$

$$Edge\ Gradient\ (G) = \sqrt{Gy^2 + Gx^2} \ldots [5]$$

$$\text{Angle } (\theta) = \tan^{-1}\left(\frac{Gy}{Gx}\right) \ldots [5]$$

The formula for finding resultant gradient and the angle θ would be the same and in vertical segmentation, the value of θ would always be 90^0

4.2.3 Non-maximum suppression: - The process is similar to that done in horizontal segmentation with a slight difference as the value of θ is 90^0, pixels pointing in the y-direction will be compared with the pixels to its left and right and if it is the local maxima for gradient values, it will be accepted otherwise it will simply be supressed.

4.2.4 Hysteresis Thresholding: - The process is exactly the same as that in horizontal segmentation with the difference that "sure-edges" are in the vertical direction.

Again the "sure-edges" are replaced by Hough lines and the partitioned images are further partitioned into smaller images by selecting the area between Hough lines. The only difference from horizontal segmentation is that if there is a region between two Hough lines which doesn't have any dots, that region is not discarded because it depicts the space between two words in the scanned image.

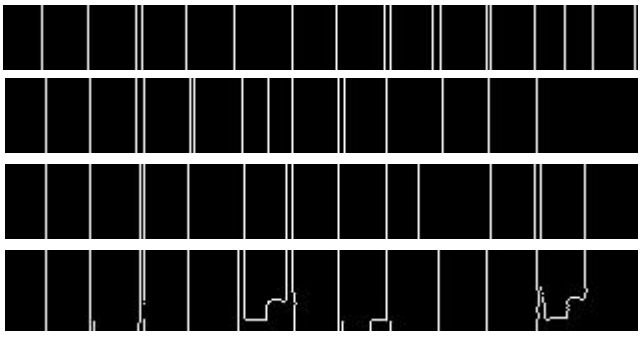


Fig 16: Canny edges

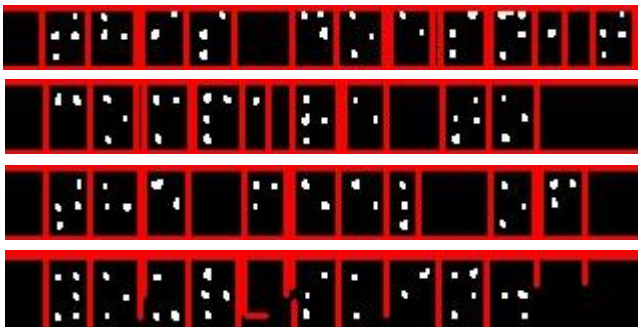


Fig 17: Hough lines

4.3 Extraction of dots

The partitioned images are seen to be of different sizes but before extraction of dots, all the images need to be made of the same size. A standard size is fixed and if the partitioned image is bigger than the standard size then the image is compressed to the standard size and if it is smaller than the standard size then the image is enlarged.

The new set of partitioned images will now be divided into 6 equal blocks in the form of a 3X2 matrix and if there is a single pixel in the block which has the value 1, then the block is said to be filled and the value of the block will be taken as 1.

The values of the block are checked against the standard symbols of the language and the partitioned images are converted into alphabets, numerals or a symbol depending upon the value of the dots and the entire image (which is a collection of all the partitioned images) has now been converted into English and stored in an output file.

(Before compression) (After compression)

Fig 18: Standard size 3x2 block

(Before enlarge) (After Enlarge)

Fig 19: Enlarged 3x2 block

5. ERROR CORRECTION

Even after removing noise through pre-processing and removing blur edges after segmentation some blocks might have noise elements and a dot might be detected even though it was not meant to be there and hence errors might come in the output file even after these many precise steps and hence error correction needs to be done in the final output file by using standard algorithms for checking spellings and identifying and correcting grammatical errors.

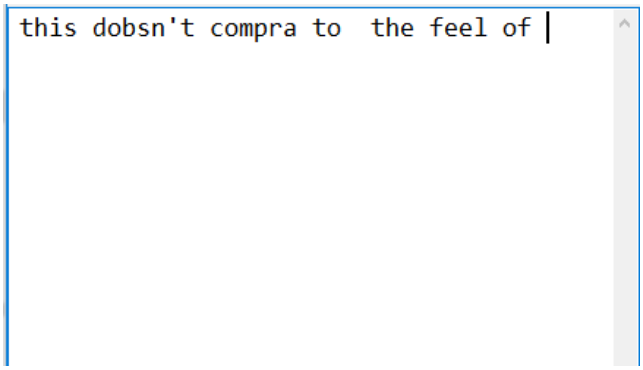


Fig 20: Output file before error correction

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 09, September-2019

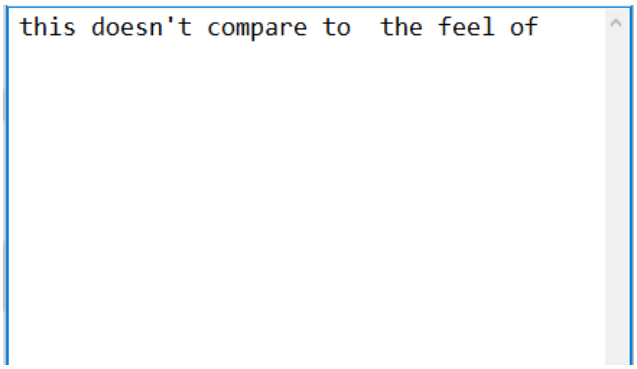


Fig 21: Output file after error correction

6. TEXT-TO-SPEECH

After extracting the dots and putting the output file through error correction, the final step will be converting the output text to speech to give the required output. The file is broken into sentences and the sentences are further broken into words which are concated in the string which is to be sent as input in standard text-to-speech software.

7. CONCLUSION

In this paper the process of extraction of dots from the image is discussed after the removal of noise and the details of identification of the edges and Hough lines are explained. Further details of error correction are discussed to increase the efficiency of the project. The process is divided into different steps which involve scanning the document, pre-processing to remove noise and change the image to grey scale, detect edges by segmentation in both the horizontal and vertical direction, extract the dots by dividing the segmented images into blocks, convert the dots detected to English and finally try and remove errors which might have crept into the output file.

8. REFERENCES

[1] Abdul Malik S. Al-Salman, Ali El-Zaart, Yousef Al-Suhaibani, Khaled Al-Hokail, Abdu Gumaei, "Designing Braille Copier Based on Image Processing Techniques", International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-4 Issue-5, November 2014.

[2] S.Padmavathi, Manojna K.S.S, Sphoorthy Reddy .S and Meenakshy.D, "CONVERSION OF BRAILLE TO TEXT IN ENGLISH, HINDI AND TAMIL LANGUAGES", International Journal of Computer Science, Engineering and Applications (IJCSEA) Vol.3, No.3, June 2013.

[3] https://docs.opencv.org/3.4/d9/d61/tutorial_py_morphological_ops.html

[4] https://docs.opencv.org/3.3.0/d4/d13/tutorial_py_filtering.html

[5] opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_canny/py_canny.html

[6] 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 Volume 14, Number 1, April 2016

[7] AbdulMalik S. Al-Salman, Yosef AlOhali, and Layla O. Al-Abdulkarim, "Trends And Technologies In Optical Braille Recognition".

[8] Itunuoluwa Isewon, Jelili Oyelade, Olufunke Oladipupo, "Design and Implementation of Text To Speech Conversion for Visually Impaired People", *International Journal of Applied Information Systems (IJAIS) – ISSN: 2249-0868 Foundation of Computer Science FCS, New York, USA Volume 7– No. 2, April 2014*

[9] SHREEKANTH.T, V.UDAYASHANKARA, "An Algorithmic Approach for Double Sided Braille Dot Recognition Using Image Processing Techniques", International Journal of Image Processing and Visual Communication ISSN (Online) 2319-1724 : Volume 2 , Issue 4 , June 2014.

[10] Megha Gadag, Dr. V Udayashankara, "Generation of English Text from Scanned Braille Document".

[11] Himali Parekh, Stuti Shah, Feni Patel and Hardik Vyas, "Gujarati Braille Text Recognition: A Review", IJCSE, Vol. 7, No. 1, pp.19-24, 2016

[12] Samer Isayed, Radwan R.Tanboub, "A Review of Optical Braille Recognition", IEEE, pp. 1 – 6, 2015.

[13] https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html

[14] Farhan Bodale, Uddhav Bhide, Dilip Gore, "Braille Translation", *International Journal of Research in Advent Technology, Vol.2, No.4, April 2014, E-ISSN: 2321-9637*

[15] https://docs.opencv.org/3.4.1/da/d22/tutorial_py_canny.html