Braille Pattern Recognition using Digital Image Processing

Abhinav A.A, Tarun Eswar R, Pranavram V, Varun Kumar S

Indian Institute of Information Technology, Design and Manufacturing, Kancheepuram

***Abstract*— The project uses image processing techniques for converting an imprinted Braille pattern into natural language character. The processing takes place through a series of steps implemented in MATLAB. The major step is the removal of noise from the scanned image of an imprinted Braille pattern.**
The complete system is fully tested and works efficiently for both manually written as well as computer generated Braille for U.K English language and is found to give accurate results. This novel idea can be implemented for any language, thereby has a profound positive impact on the development of education material for visually challenged individuals together with other applications.

***Keywords*— Image Processing; Braille; Pattern recognition; Segmentation; Noise Removal**

I. Introduction

Information in written form plays an undeniably important role in our daily lives. From education and leisure, to casual note taking and information exchange, recording and using information encoded in symbolic form is essential. Visually impaired (blind and partially sighted) people face a distinct disadvantage in this respect.

Addressing this need, the most widely adopted writing convention among visually impaired people is Braille. Since its inception in 1829, significant developments have taken place in the production of Braille and Braille media as well as in the transcription of printed material into Braille.

Braille is a system of touch reading and writing for blind persons in which raised dots represent the letters of the alphabet. It also contains equivalents for punctuation marks and provides symbols to show letter groupings. Braille is read by moving the hand or hands from left to right along each line. The reading process usually involves both hands, and the index fingers generally do the reading. By using the braille alphabet, people who are blind can review and study the written word. They can also become aware of different written conventions such as spelling, punctuation, paragraphing and footnotes. Most importantly, braille gives blind individuals access to a wide range of reading materials.

Braille is a system of raised dots arranged in cells. Any combination of one to six dots may be raised within each cell, and the number and position of the raised dots within a cell conveys to the reader the letter, word, number, or symbol the cell represents. There are 64 possible combinations of raised dots within a single cell. Due to the varying needs of Braille readers, there are three different grades of Braille.

II. Grade 1 Braille

In the first of the grades of Braille, grade 1, each possible arrangement of dots within a cell represents only one letter, number, punctuation sign, or special Braille composition sign - it is a one-to-one conversion. Individual cells cannot represent words or abbreviations in this grade of Braille. Because of this grade's inability to shorten words, books and other documents produced in grade 1 Braille are bulkier and larger than normally printed text. Grade 1 Braille is typically used only by those who are new to learning the grades of Braille, but as of the early 2000s a new movement was in place among elementary school teachers of Braille to introduce children with sight difficulties to grade 2 Braille immediately after teaching the basics of grade 1 Braille.

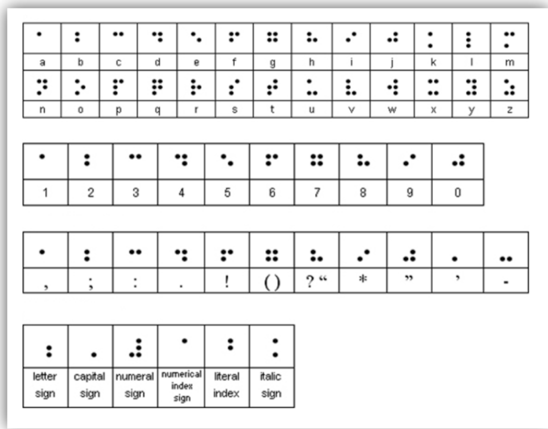

Fig. 1 Grade 1 Braille

III. PROCEDURE OF PROPOSED TECHNIQUE

The complete procedure for Braille pattern recognition is divided in to two steps i.e., preprocessing and Pattern recognition.

a. The Pre-processing Steps are listed as follows:

1) Scanning of the Braille
2) Conversion of RGB image to binary image
3) Complementing the image (bit 1 to 0 and bit 0 to 1).
4) Removal of the noise and unwanted dots.
5) Removal of the unwanted redundant edges of the image.

b. The Pattern Recognition Steps are listed as follows:

1) Resizing of the binary image
2) Generation of a pattern vector on the basis of this result in terms of 0's and 1's.
3) Division of pattern image in to six equal grids of size 3 x 2. The size 3 x 2 is selected because this is the international standard of the Braille.
4) Linking of the pattern vectors with corresponding alphabets.

c. PREPROCESSING

Scanning of Braille

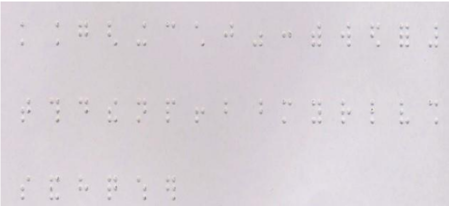

Fig. 2 Input RGB Image

We convert the given RGB Image into Binary Image using inbuilt MATLAB functions, *rgb2gray()* and *imbinarize()* with appropriate sensitivity and foreground polarity.

Fig. 3 Grayscale Image

Next, we complement the image (bit 1 to 0 and bit 0 to 1) using Negation operator.

Fig. 4 Inverted Binary Image

Noise Removal and dot detection is done using '*two stage'* method of *imfindcircles()* command with a sensitivity of 0.95

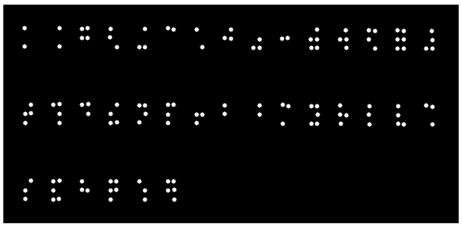

Fig. 5 Image after Noise Removal

Removal of unwanted edges is done next by isolating the rows and columns containing the Braille dot.

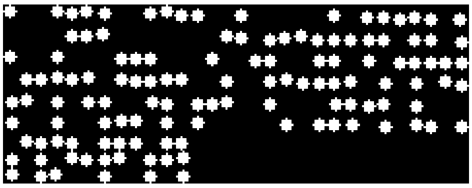

Fig. 6 Image after Removing Unwanted Edges

d. PATTERN RECOGNITION

The image is converted into a pixelated image for Pattern Vector Generation

Fig. 7 Pixelated Image

The Pixelated Image is converted into a Matrix of 0’s and 1’s indicating the position of the Braille dots.

1	0	0	1	1	1	1	0	0	1	1	1	1	0	0	1	0	0	0	0	0	1	0	1	1	1	1	1	0	1
0	0	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	0	1
1	0	0	1	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1	1	0	0	1	0	0	1	1	1	1	1	1	0	1	0	1	0	1	1
1	1	0	1	0	1	1	0	0	1	1	0	1	1	1	0	0	1	0	0	0	1	1	1	1	0	1	0	0	0
1	0	0	1	0	0	1	1	1	0	1	0	1	0	0	0	0	0	1	0	1	1	1	0	1	0	1	1	0	1
0	1	1	1	1	0	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 8 Image Matrix

The Image is divided into 3x2 grids using *mat2cell()* commands with appropriate parameters. Each divide cell is linearized into a 1D Vector and compared with the Database to output the translated Braille Alphabet.

{'K'}	{'P'}
{'(italics)'}	{'!'}
{'G'}	{'B'}
{'C'}	{'M'}
{'J'}	{'Y'}
{':'}	{'R'}
{'W'}	{'L'}
{'#'}	{'V'}
{'T'}	{'S'}
{'D'}	{'H'}
{'N'}	{'Q'}

Fig. 9 Output

VII. LIMITATIONS

The program accepts only RGB Input Images and best results are achieved when the Input Braille Image is properly aligned

VII. CONCLUSIONS

The project has been tested on Braille and is focused on English Braille only. With slight modifications, it can be extended to other languages as well. This makes our project highly dynamic and applicable to almost all the languages available in the world.

REFERENCES

[1] Imprinted Braille-Character Pattern Recognition using Image Processing Techniques 2011 International Conference on Image Information Processing (ICIIP 2011)

[2] Pattern Recognition and Image Analysis (PRImA) group, Department of Computer Science University of Liverpool, Liverpool, L69 3BX, United Kingdom

[3] MathWorks - Makers of MATLAB and Simulink - MATLAB & Simulink (Support Website)