An Application of Eight Connectivity based Two-pass Connected-Component Labelling Algorithm For Double Sided Braille Dot Recognition

Shreekanth T *speak2shree@gmail.com*
Research Scholar,
JSS Research Foundation,
Mysore, India.

V. Udayashankara *v_udayashankara@yahoo.co.in*
Professor,
Department of IT,
SJCE, Mysore, India.

Abstract

The intrinsic noise present in the image during the acquisition phase marks the recognition of Braille dots a challenging task in Optical Braille Recognition (OBR). Further, while the Braille document is being embossed on either side in the case of Inter-Point Braille, this problem of Braille dot recognition is aggravated and it makes the differentiation between recto (convex) dots and verso (concave) dots more complex. Also, the recognition of Braille dots should be carried out by reading information recorded on both sides of paper by scanning only one side. This work proposes a novelty to circumvent this issue for distinguishing convex points from concave points even if they are adjacent to each other by using only the shadow patterns of the dots and by employing the connected component labelling using two-pass algorithm and the eight connectivity property of a pixel. Enthused by the fact that, during the acquisition phase, the reflection of light through the verso dots results in a high pixel count for them when compared to the recto dots, this technique works perfectly well with good quality Braille. Furthermore, due to the natural problems like ageing and frequent usage of the document the Braille dots tend to deteriorate resulting in the down fall of the performance of the algorithm for the Braille image. Besides to this for the recognition of the Braille cell in a Braille document with some special cases an adaptive grid construction technique has also been proposed. The results extracted reveal that the enactment of the proposed technique is much consistent and dependable and that the accuracy is very much comparable to the modern state of the art techniques.

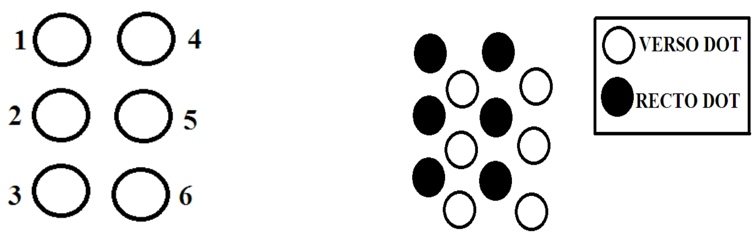
Keywords: Braille, Connected Component Labelling, Eight-Connectivity, OBR**,** Recto, Verso.

1. INTRODUCTION

Braille is a form of written language for blind people, in which characters are represented by patters of raised dots that are felt with the fingertips. The Braille system, devised in1821 by Frenchman Louis Braille consists of patterns of raised dots arranged in cells of up to six dots, with each cell representing a letter, numeral or punctuation mark. The dots in each cell is arranged in three parallel rows having two dots each. The dot positions are identified by numbers from one through six. Sixty-four combinations are possible using one or more of these six dots.

In order to establish a bi-directional communication between the visually impaired and the sighted community workable, it is required to transliterate the Braille documents to the text document in the corresponding language. Optical Braille Recognition comprises of acquiring and processing the images of Braille documents for the purpose of converting the embossed Braille characters into their corresponding natural language characters. The need for OBR is that there are significant number of old Braille documents that need to be reproduced so that they can be preserved and accessed by more people. Like other documents Braille documents need to be converted to digital format in order to facilitate storage, maintenance, duplication and text to speech conversion. Everyone who works with blind people and does not read Braille will benefit from using the OBR. The main reason for developing a system that can read Braille is to preserve and multiply large volumes of manually crafted books. Many books on mathematics or music are very difficult even for skilled copyist to retype due to the specific rules that apply in Braille.

As the dots on both sides of the page are visible from one side, both sides of the page can be recognised in a single scan. Printed Braille documents are very bulky. To mitigate this problem, most Braille documents are printed in inter-point with the embossing done on both sides of each page with a slight diagonal offset to prevent the dots on the two sides from interfering with each other. This makes the translation process more difficult. Depending on the presence of protrusions and depressions the Braille document can be classified as single sided and inter point Braille. If the document contains only the protrusions on single side then it is a single sided Braille document as shown in Fig.1 (a). If the document contains the protrusions and depressions on single side or if the document contains protrusions and depressions on both the sides then it is a double sided Braille document and is as shown in Fig.1 (b). Double-sided output takes less space and uses less paper as compared to single sided Braille document since the information lies on both sides of the Braille document thus volume can be reduced to half of the single sided Braille volume. All dots on a Braille page should fall on an orthogonal grid. When texts are printed double sided (Inter-point), the grid of the inter-point text is shifted so that the dots fall in between the primary side dots. During the recognition of double sided Braille document, the presence of protrusions and depressions may cause interference to recognition system. The dimensions of a Braille dot have been set according to the tactile resolution of the fingertips of person. For both the single sided and double sided Braille document the dot height, cell size and cell spacing are always uniform.



(a) (b)

FIGURE 1: (a) The Braille Cell (b) Inter-point Braille.

Several researchers have made efforts to recognise single sided Braille documents and the recognition rates are to the acceptable level [3],[8],[9],[12]-[15],[18] and [19]. An extensive literature survey indicates that as the recognition of single sided Braille documents is easier when compared to the double sided Braille document, lesser number of works has been done on recognition of Inter-point Braille document. Several authors have as well proposed numerous techniques to differentiate the recto and verso dots in an inter-point Braille document [1], [4]-[7], [10], [11], [16], and [17].

In [1] J. Mennens et.al presented a very first approach for double sided Braille page recognition. This work throws light on the assumption as to how the dots of a digitized Braille page are represented with light and dark areas. Then the areas are classified by making a three value image. The resulting image has five values: regions with value +2(recto) and -2(verso) are called core regions, regions with value -1(recto) and +1(verso) are called side-lobes and 0 is background. Grid lines are then searched by making histograms of rows and columns.

In [4] R.T Ritchings et.al developed a prototype system that after the Braille document is scanned, the system proceeds to identify the location of the protrusions and depressions by exploiting the differences in gray levels in the image. These arise during scanning from the reflected light and the shadows created by the protrusions and the depressions on the document surface. The protrusions first generate a dark area and then a lighter one in the scanning direction (vertical). The algorithm to identify protrusions and depressions proceeds by making an allowance for the dark regions of each dot and check is made to determine whether a light region exits above it. If it does exist within the limits of the predictable Braille character height then a depression is found, if not a check is made to govern whether a light area exits beneath the dark one. Correspondingly if one exists above within the vertical limit, a protrusion is found. If an area is found to be comprising of a protrusion or a depression, then those areas are marked as used and are not considered again and on the other hand regular spacing between the Braille dots and cells are used for Braille character recognition.

In [5] Yoshifumi oyama et.al proposed a method for distinguishing convex points from concave points by the highlight and shadow patterns generated by exposing the Braille points to LED lights. This work used the difference in the strength of the reflected light from convex and concave points to separate the convex points from the concave points. For the Braille character extraction as a substitute for conventional 2*3 point set mask here 3*3 i.e. 9 windows are used as the positioning performance of the mask that determines the character recognition accuracy.

In [6] C M Ng et.al proposed a regular feature extraction for recognition of Braille. The illumination characteristics of the recto and verso dots are efficiently made use of by this algorithm. Expending these illumination characteristics the position of the illuminated hole can be castoff as the feature to make a distinction between the front faced dots and the back faced dots. Based on the boundary co-ordinates information and the illumination characteristics, two standard templates were constructed to represent the front-face dots and the back face dots. To assess the correlation at each pixel position these templates were applied to every position of the image. The front faced and the back faced dots are then extracted depending on the correlation values attained.

In [7] Antonacopoulos et.al. Proposed an algorithm for double sided Braille dot detection which is analogous to the one debated above. At this juncture a novel technique is proposed for grid formation. The system described here constructs a relatively flexible grid by allowing variations in the position of characters between different lines. First, the grouping of dots that have the same y co-ordinates are done in order to identify the rows of dots. Having identified the rows of Braille dots, a frequency histogram of the vertical distances between adjacent rows is calculated. The histogram ought to have two main peaks, one indicating the inter-character vertical distance between dots and other the inter-line distance. i.e., vertical distance between the bottom row of a Braille character line and the top row of the next.

In [10] Abdul Malik Al-Salman et.al developed an Arabic OBR system that is competent enough to recognize both single-sided and double-sided Arabic Braille documents from a single scan. This algorithm takes into account the implication that if the dark region comes at the top and the bright one comes at the bottom then it is a verso dot, the inverse situation results in a recto dot. Also it takes into account that the average dot height is 8 pixels. Bright pixels are allotted the value +1 and dark pixels are assigned the values -1. If pixel (1) + pixel (2) <0 and pixel (7) + pixel (8) < 0 then this is part of a recto dot. If pixel (1) + pixel (2) <0 and pixel (7) + pixel (8) > 0 then this is part of a verso dot. At this point the Braille cell recognition is done using the horizontal and vertical projection and as well the average distance between the rows holding the dots and between the columns holding the dots.

In [11] Abdul Malik S, Al-Salman presented an innovative algorithm for Braille cells recognition using image processing technique. The Braille image segmentation is done by means of a stability thresholding method with a beta distribution. A grid containing the Braille dots is moulded to ensure accurate detection and extraction of dots composing Braille cells. Then the recto dot is identified by a light region that exits underneath a dark region. Once the recto dots and verso dots are being recognised, Braille cells are then identified based on the standard regrouping of dots.

In [16] Amany-al-soleh et.al proposed a method for dot detection of Braille images using a mixture of beta distributions. In this work it is presumed that the scanned Braille page consists of three classes of pixels; a mid-gray background, a pair of light area and dark area for each recto and verso dot. At first in order to segment the scanned Braille image into three classes, thresholding is proposed onto it. Then the initial threshold values T_1^0 and $T_2^0$for stability thresholding is estimated. To do this the maximum value of the histogram is calculated first. T_1^0 will be the average gray level value of image starting from 0 to maximum value, T_2^0 will be the average gray level value of image starting from maximum value to 255. The stability thresholding procedure is repeated until the error is zero. By the culmination of this algorithm the optical values of T_1^{New} and T_2^{New} are obtained. Then for detecting the recto and verso dots from double sided Braille documents a grid is formed first to accomplish the detection of dots for every box in the grid. Further to decide whether it holds a recto dot or a verso dot a test is being carried out. The test checks if the upper half contains a light region and the lower half contains a dark region. If this was the case then this is considered a recto dot and will be drawn on the output range in the same location.

In [17] Bhattacharya.J et.al proposed an algorithm which makes use of the regular inter-dot and inter-cell spacing. In order to detect each cell a sliding window with a fixed interval is used to slide over the entire Braille image. Each window consists of both side dots. These dots are differentiated as front or back sided through a sliding method. Firstly every window is sub divided into 3regions R1, R2 and R3. All the dots which lie in region R1 entirely are accepted, whereas dots which lie completely in the region R3 are rejected. Dots lying in both R2 and R3 are either front or backside merged dots or front side dots. For the former case the dot centroid is modified using the merged dot centroid and bottom extreme point of the dot. Yet again dots lying in both R1 and R2 are either merged dots or backside dots. Here for the former case the dot centroid is modified using the merged dot centroid and bottom extreme point of the dot whereas for the later scan the dot is rejected.

The downsides of the above works are i) The increased average processing time introduced by the template matching procedure for differentiating the recto dots from verso dots ii) The error introduced due to the merging of Braille dots and thus causing an ambiguity and iii) The need for restructuring of templates depending on the height of the Braille character as the spatial resolution of the Braille image varies. Thus these techniques do not lead to an adaptive approach.

Driven by the above actualities, an attempt has been made in the present work to cultivate an adaptive algorithm using the concept of eight connectivity based two-pass connected-component labelling algorithm to differentiate the recto and verso dots from the double sided Braille document. The experimental results show that the proposed technique can deliver an enhanced performance when equated to the other techniques mentioned for Recto and Verso Braille dots separation. This paper is divided into four sections: wherein the section II discusses in detail the projected work. Section III presents the results and discussions. Section IV, offers conclusions and directions for the future work.

2. METHODOLOGY

The overview of the proposed OBR system is illustrated in Fig. 2. The objective here is to develop an optical Braille character recognition system which takes the different resolution of the scanned Braille document and the different Braille image quality into deliberations. The proposed system can be developed as follows

2.1 Extraction of Shadow Patterns and Median Filtering

The Braille images are scanned using the HP Scan jet 3400C A4 size scanner, with the image resolution of 200dpi, the spatial resolution of 1501 x 2121 and with the bit depth of 24 bit. The algorithm begins by converting the innate colour image to the gray scale image. The diverse gray levels result in the image which is due to the variations of the reflected light and the shadows created by the protrusions and the depression on the document surface during the scanning process. In the scanning direction the protrusions generated dark area first and then a lighter one. It was our observation that the depression produced the opposite. The shadow produced by the depressions relatively account for a lesser number of pixel count when compared to the shadow produced by the protrusions. This motivated us to retain only the light areas of the protrusions and depressions and thus eliminating the dark regions. With a motive to preserve only the light areas of the dots and to remove any inherent noise present in the image thresholding is being performed on the gray scale image.

The thresholding function used to do this is given below:

$$Y(I, J) \begin{cases} = X(I, J); & \forall X(I, J) > \max(X(I, J)) - 10; \quad 1 \le i \le M, 1 \le j \le N \\ = 0; & \forall X(I, J) \le \max(X(I, J)) - 10; \quad 1 \le i \le M, 1 \le j \le N \end{cases} \quad (1)$$

Those pixels having threshold value less than this are made zero and those pixels with values greater than threshold are retained as they are.

The impulse noise like components contained in a thresholded image is eliminated using the median filtering approach. Now the median filtered image contains only the recto and verso dot components. Further the morphological dilation is performed in order to increase the area of the dot and this is more effective for increasing the area of the verso dot as compared to recto dot. In order to differentiate between the recto and verso dot the eight connectivity property of the pixel relationship is employed. It is perceived that the eight connectivity pixel count for the recto dot is less compared to the verso dot. Then to differentiate between the recto and verso dots, the thresholding operation is performed on the basis of the eight connectivity based pixel count value. The scanned document sample is shown in Fig. 3 and also the portion of the recto and verso dots are shown in Fig. 4(a) and (b) respectively. It can be observed that the dots on the front side are protruded above the page and those on the back side form depressions. These concave and convex characteristics of the dots reflect the light into two different angles, creating a light region at the top half of the captured dots for front sided Braille dots and at the bottom of for those back sided dots [6].

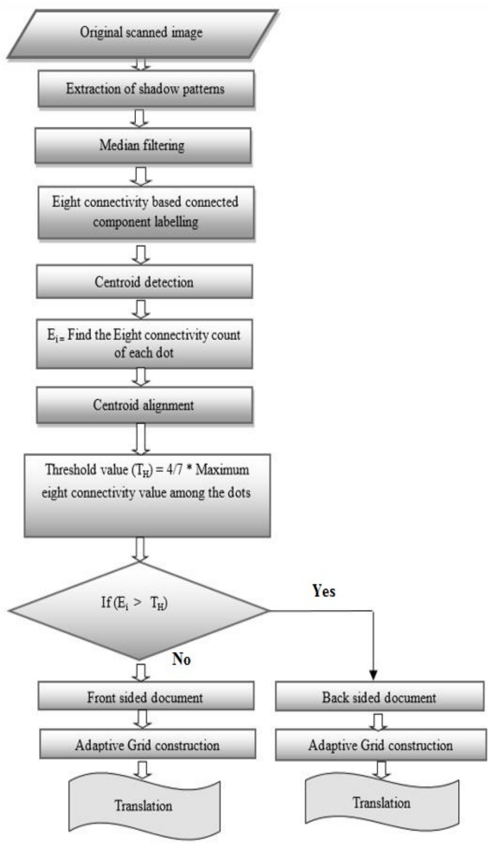


FIGURE 2: Flowchart of the Braille Recognition System.

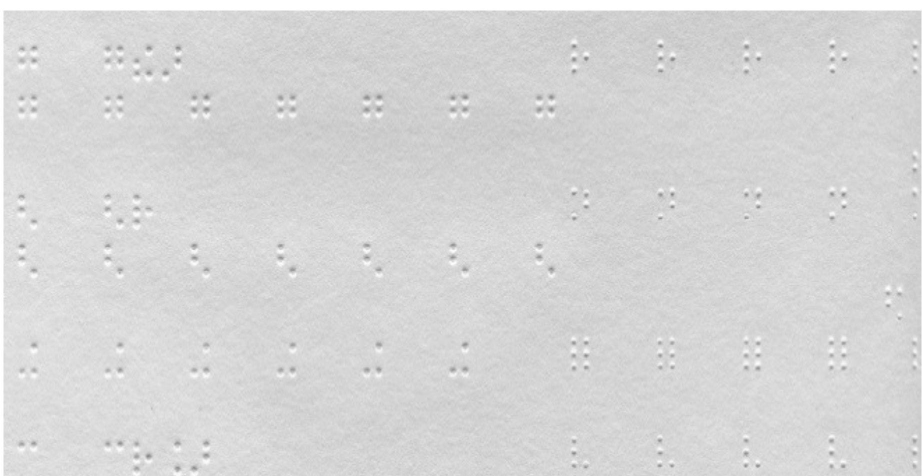
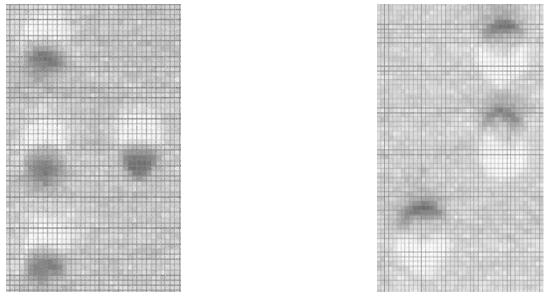
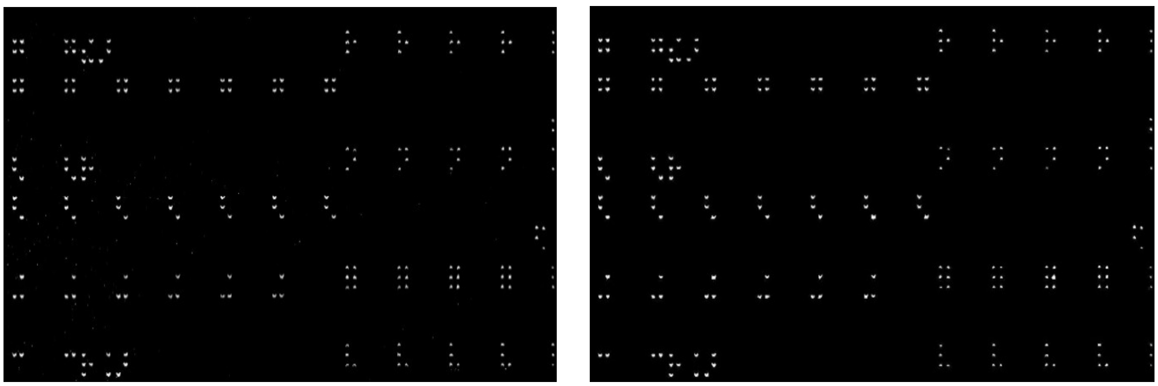


FIGURE 3: Original Scanned Braille Image.



(a) (b)

FIGURE 4: a) Recto Dot: Light area first followed by the dark area b) Verso Dot: Dark area first followed by the light area.



(a) (b)

FIGURE 5: a) Thresholded image with impulse noise b) Median filtered image.

2.2 Eight connectivity based two pass connected component labelling algorithm and centroid calculation

If any pixel is connected horizontally, vertically and diagonally then it is called an eight connected pixel. For any pixel p(x, y) the definition of an 8-connected component is that the pixel p(x, y) should be connected with any of the pixels described below [21]:

$$(x+1,y),(x-1,y),(x,y+1)(x,y-1),(x+1,y-1),(x+1,y+1),(x-1,y-1),(x-1,y+1) \quad (2)$$

The extraction and labeling of various disjoint and connected components in an image is the most vital part in a number of automated image analysis applications. As the connected component labeling works either on binary or gray images, accordingly different measures of connectivity are possible. These images are typically the output from an alternative image-processing step, such as segmentation.

The process of grouping the *connected* pixels in an image for assigning an unmatched label to each object in an image is usually done by the *Connected Component Labeling* technique .The reason being that these labels are the key for various other analytical procedures, an indispensable part of most applications in pattern recognition and computer vision, such as character recognition. The basic approach is to scan the image and group its pixels into components based on pixel connectivity, *i.e.* all pixels in a connected component share similar pixel intensity values and are in some way connected with each other. There are two much known ways of defining connectedness for a 2D image: 4-connectedness and 8-connectedness. In this paper, we use the 8-connectedness as illustrated in expression (2). Once all groups have been determined, assign labels to each pixel until the labels for the pixels no longer change. As a result of the scan, no temporary label is attributed to the pixels belonging to different components but on the contrary different labels may be associated with the same component. Consequently, equivalent labels are sorted into equivalence classes and a unique class identifier is designated to each class after the completion of the first scan. Then, a second scan is run over the image so as to substitute each temporary label by the class identifier of its equivalence class [20].

The number of pixels having the same label is counted and these values are used for discerning the recto and verso dots. The centroid of a labelled component is determined using the equations given below.

$$a\,(i,1) = 1 \;\forall\; 1 \le i \le M \tag{3}$$

$$a\,(1,j) = i \forall\; 1 \le i \le N \tag{4}$$

$$b\,(i,j) = a\,(i,1) * a\,(1,j) \tag{5}$$

$$\text{Area} = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1} z(i,j) \;\forall\; 1 \le i \le M,\; 1 \le j \le N \tag{6}$$

$$\text{Mean } x = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[z(i,j) * b\,(i,j)] \,/\, \text{area} \tag{7}$$

$$c\,(i,1) = I \;\forall\; 1 \le i \le M \tag{8}$$

$$a\,(1,j) = 1 \;\forall\; 1 \le i \le N \tag{9}$$

$$d\,(i,j) = c\,(i,1) * c\,(1,j) \;\forall\; 1 \le i \le M,\; 1 \le j \le N \tag{10}$$

$$\text{Mean } x = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[z(i,j) * b\,(i,j)] \,/\, \text{area} \tag{11}$$

$$\text{Mean } y = \sum_{i=0}^{M-1}\sum_{j=0}^{N-1}[z(i,j) * d\,(i,j)] \,/\, \text{area} \tag{12}$$

The centroid extracted Braille image is shown in Fig.6. Either due to the little skewness of the document or due to the deteriorated Braille dots the centroids of the dots are not aligned properly. To circumvent this problem the centroids must be aligned vertically and horizontally by defining the threshold for the alignment. In this work we have used the following equation for aligning the centroids of the Braille dots.

$$x_{i+1} = x_i;\forall\; x_i - \Delta x \le x_{i+1} \le x_i + \Delta x;\; 1 \le i \le M \tag{13}$$

$$y_{i+1} = y_i\;;\;\forall y_i - \Delta y \le y_{i+1} \le y_i + \Delta y;\; 1 \le i \le N \tag{14}$$

This has been designed considering the very little skewness in the document as this work does not take the rotation angle into consideration. The eight connectivity component values are assigned to the respective centroid.

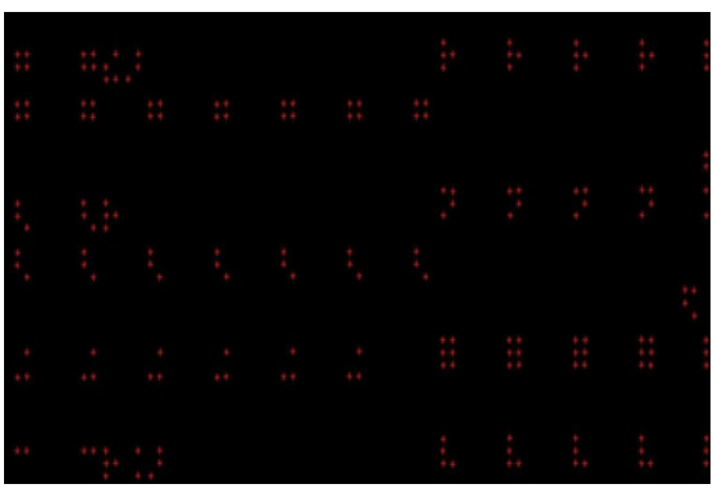


FIGURE 6: Centroid Detected Image.

2.3 Recto / Verso dot Deparation

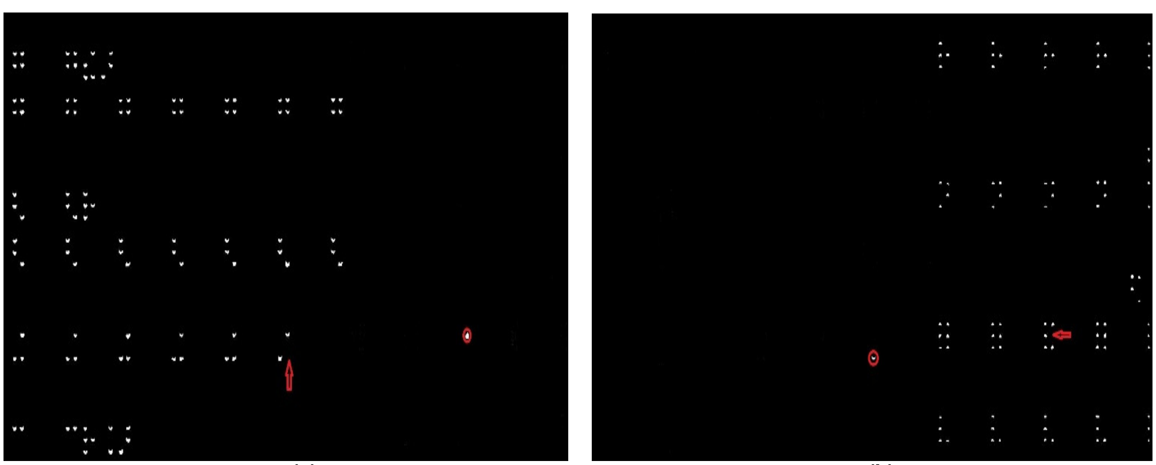
The principal objective of this work is the separation of recto/verso dots. The incentive to execute this is the variance in the eight connectivity pixel count for the shadow region of the recto and verso dots. This is attributed due to the reflection property of the light. Also, this difference is independent of the spatial resolution of the Braille image. An in depth analysis has reviewed that the thresholding using the relation described below can be used for differentiating the recto dots from verso dots in an inter-point Braille image.

$$T_{H} = 4/7 * \text{Maximum eight connectivity count among the dots} \quad (15)$$

By considering the average eight connectivity pixel counts of un-deteriorated Braille dots from the developed Braille database this threshold value has been designed. Those dots with eight connectivity count value greater than the threshold T_H are considered as the verso dots and those dots for which the eight connectivity count value lesser than the threshold T_H are considered as the recto dots. The only detriment of the direct thresholding is that if the dots are deteriorated due to frequent usage and ageing then the eight connectivity count for such dots are less and thus it leads to false recognition of dots which is illustrated in Fig.7 (a) and (b). If the dots are deteriorated then the verso dots take the appearance of the recto dots and vice-versa. With the location of the dots being same as that of the original document, the recognised dots are then separated into recto and verso dots and placed in a different document. Each side of the Braille dots are transcribed into their corresponding natural text when the output of this stage is fed into the adaptive grid construction block.

2.4 Adaptive Grid Construction

After the separation of the recto and verso dots the grids are constructed discretely for the front and back sides of the document. The grid construction is essential to recognise a Braille cell in the Braille document. In former works the grids were constructed according to the standard Braille dimension and the summation of pixels within the dot frame was done to separate the recto dots from verso dots. This is a dreary and a time consuming process. Also, the constructed grids were not adaptive, as in it has not considered the different resolution of the scanned Braille document. Henceforth in this work we propose the adaptive grid construction technique which, in general can be applied to any sort of Braille document and is shown in Fig.8. This reduces the computer time and also any possible miss classification of the dot is avoided. In order to construct the adaptive grid an algorithm has been designed by considering the three factors such as, the distance calculation, horizontal projection profile and vertical projection profile of the Braille image.



(a) (b)

FIGURE 7: a) Recto dots extracted image. Dot enclosed by a circle is a part of verso dot and the arrow indicates the missing recto dot.

b) Verso dots extracted image. Dot enclosed by a circle is a part of recto dot and the arrow indicates the missing verso dot.

The Braille document to illustrate the process of adaptive grid construction is shown in Fig.9 (a). Here we have considered only the image consisting of recto dots, as the output from the previous step is a separate document consisting of either recto dots or verso dots only. We notice from Fig.9 (b) that, the third row of dots are missing from both line 4 and line 6. The solution to this has been incorporated and is described in the following.

It is our observation that any Braille document in general will satisfy the below characteristics

Horizontal Inter-dot (HID) distance > 2*dot width and < 3*dot width
Horizontal Inter-cell (HIC) distance > 3*dot width and < 4*dot width
Vertical Inter-dot (VID) distance > 2*dot width and< 3*dot width
Vertical Inter-cell (VIC) distance > 4*dot width and< 5*dot width

The diagram showing the Horizontal and Vertical Inter-dot and Inter-cell distances is as shown in the Fig.10.The horizontal and vertical projection profiles are drawn from the dot extracted image and are shown in Fig.11 (a) and (b). The horizontal projection profile is due to the sum of all the pixels in the row direction and vertical projection profile is due to the sum of all the pixels in the column direction. Three consecutive peaks in the horizontal projection profile indicates one complete line of a Braille document and two consecutive peaks in the vertical projection profile indicates one complete cell. From the horizontal and vertical projection profile information a grid is constructed as in Fig.12 (a). The constructed grid contains the information about the probable dot position only if the particular row or column contain the dot components. In real time all the Braille documents cannot possess all 3x2 dot information. In some cases it may be 2x2, 3x1 and so on. So as to solve this issue, distance calculation algorithm has been employed to locate the possible positions of the dots. The horizontal and vertical width of any dot in the document is calculated first and then the algorithm proceeds as follows.

It is found from the experimentation that the horizontal and vertical inter-dot distances are almost the same. However, the vertical inter-cell distance i.e., the distance between two consecutive Braille lines is slightly greater than the horizontal inter-cell distance i.e., the distance between two cells within the same Braille line.

The filled possible dot positions are shown in the Fig. 12(b). The above algorithm is used to complete the grid as shown in Fig. 12 (c). Fig.12(c) shows the concatenation of the extracted Recto/Verso dots with the adaptive constructed grid. Then the dots are scanned as shown to convert the grid to binary number followed by the decimal conversion.

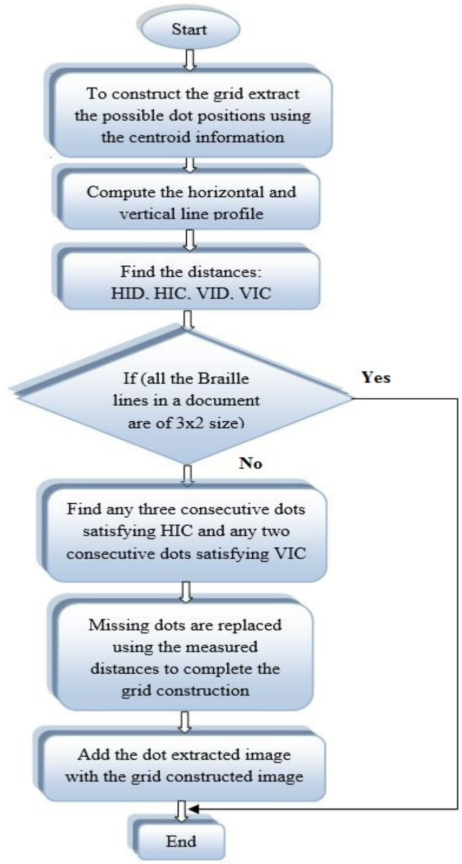
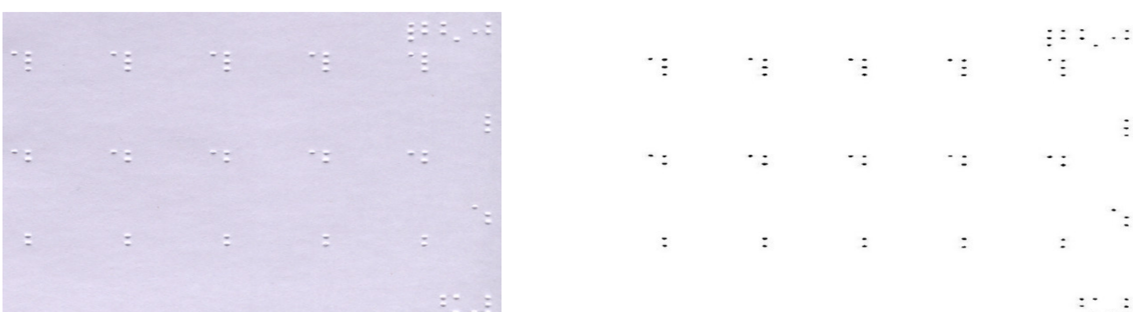


FIGURE 8: Flowchart of Adaptive Grid Construction.



(a) (b)

FIGURE 9: (a) Specimen document for illustrating the adaptive grid construction (b) Dot extracted image.

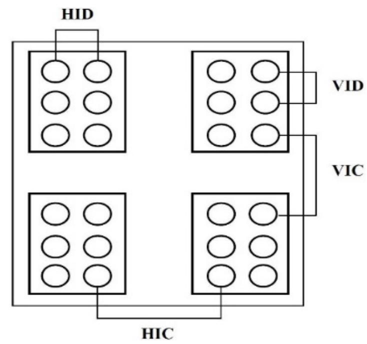
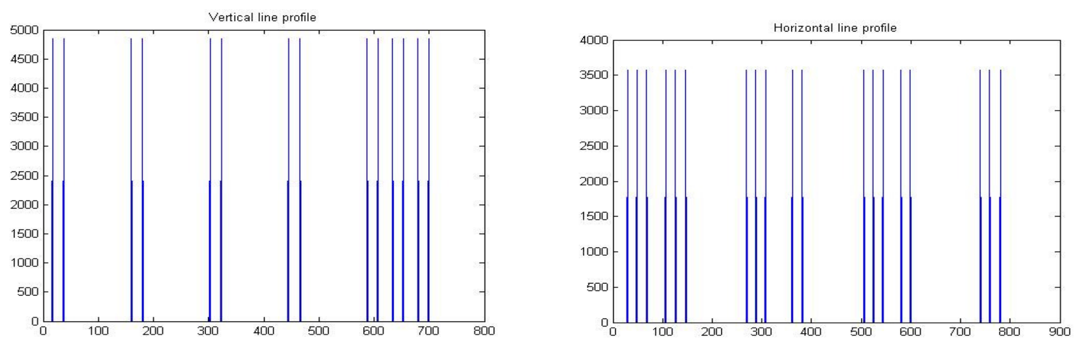
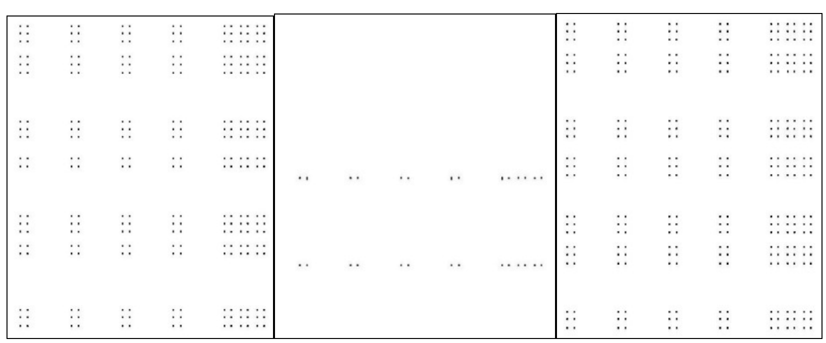


FIGURE 10: Horizontal and Vertical Inter-dot and Inter-cell distances.



(a) (b)

FIGURE11: (a) Vertical line profile and (b) Horizontal line profile for the image shown in Fig 11(b).



(a) (b) (c)

FIGURE 12: (a) Grid constructed using the possible dot positions, (b) Missing dots detected and (c) Showing the adaptive grid construction.

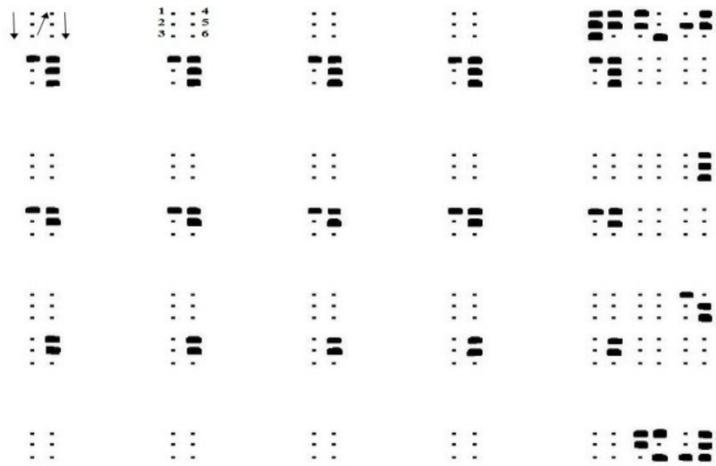
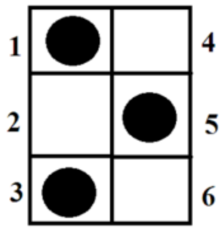


FIGURE 13: Overlapping of reconstructed grid with the recognised dots and the representation of the Braille cell scan pattern.

If there are three consecutive lines in the horizontal line profile satisfying the horizontal inter-dot distance and if there are two consecutive lines in the vertical line profile satisfying the vertical Inter-dot distance property then the algorithm does nothing. Only when Horizontal Inter-dot distance, Horizontal Inter-cell distance, Vertical Inter-dot distance and Vertical Inter-cell distance properties are violated the algorithm comes into picture. In order to address this matter an algorithm has been designed in such a way that the algorithm looks for any three dots satisfying the Horizontal Inter-cell distance and any two dots satisfying the vertical Inter-cell distance. Then if there are any missing dots of lines either in the horizontal direction or in the vertical direction it will be filled taking into considerations the distance calculation. This completes the adaptive grid construction.

Far ahead the grid constructed images and the dot extracted images are added to get an image as shown in the Fig.13. The presence or absence of all the valid Braille dots is found by multiplying the grid constructed image with the dot extracted image. If the product is true then it indicates the presence of dot and its value is indicated through 1 and if the product is false then it indicates the absence of dot and its value is indicated through 0.

During the dot recognition process all the valid Braille dots have been detected on either sides of the document and the two images are formed for each side of the document. Currently in order to convert the recognised dots into their corresponding natural characters, the scanning pattern as depicted in the Fig. is used for dividing each cell into grids consisting of six parts and corresponding code for each cell is generated according to the presence or absence of a dot in each grid. Here the binary 1 and the binary 0 represent the presence of the dot and the absence of the dot respectively. The dot positions are determined through number 1 to 6. The positions being universally numbered 1 to 3 from top to bottom on the left, and 4 to 6 from top to bottom on the right. Within each cell, the dot pattern is determined and is also represented by a bit string. These bit strings are then converted into their equivalent decimal codes by using the expression: Decimal code = $b1+b2*2+b3*4+b4*8+b5*16+b6*32$ [10]. For example, Fig.14 (a) shows the Braille cell in which recognised dots are represented by black pixels and the absence of dots are represented by white pixels. Fig.14 (b) shows the dot position, bit strings and the equivalent decimal codes for the Braille cell shown in Fig.14 (a). In order to retrieve the natural characters corresponding to the Braille characters, a matching algorithm is employed in which, an input decimal code generated from the processed image could be searched against the lookup table wherein the Unicode corresponding to the Braille characters are being stored. These Unicode's are then converted into their corresponding natural text using the Matlab function unicode2native. The entire process has to be repeated for the other side of the Braille document too.



(a)

Dot Position	1	2	3	4	5	6
Binary number	1	0	1	0	1	0
Decimal code	21					

(b)

FIGURE 14: (a) Braille Cell after Dot recognition (b) Braille code generation.

3. RESULTS AND DISCUSSION

In this section, we present the materials on which the Recto and Verso Braille dot separation using Eight Connectivity based two-pass Connected-Component Labelling Algorithm and a Novel Thresholding Technique is evaluated, the performance measures used to evaluate the algorithm, and the results obtained. The methodology has been gauged quantitatively and qualitatively expending two locally established databases (DB1 and DB2). The database DB1 contains 25 colour images of the Inter-point Braille. This image set sans overlapping of Recto-Verso dots and Inter-dot deformation which is one of the vital hitches that is most commonly introduced during the Braille embossing process. Thus the Recto and Verso Braille dots from this database can be separated workably. The database DB2 contains 25 colour images of the Inter-point Braille consisting of scanned Braille documents with Inter-dot deformation and overlapping of Recto-Verso dots. The images of these two databases were captured by a HP Scan jet 3400C scanner. Table 1 gives the complete description of the databases used in this work.

Number of Braille Documents	50 (25 DB1 + 25 DB2)
Braille Document Type	Inter-Point Braille, Grade-*I*
Digital Format	24 bit Color Image
Resolution	300 dpi
Pixel Resolution	2300x1700
Image Format	Bit-Map (bmp)
Document size	26cm (Horizontal) , 30cm (Vertical)
Total number of dots in DB1	6359
Average number of characters per sheet in DB1	254
Total number of dots in DB1	14855
Average number of dots per sheet in DB1	594

TABLE 1: Description of the Braille Database created.

The dot separation using a novel combination of Eight Connectivity based two-pass Connected-Component Labelling algorithm and a novel thresholding technique is insensitive to majority of the noise present in the acquired image.

For evaluating the efficiency of the proposed method, we have considered four events; two classifications and two misclassifications. The classifications are the True Positive (TP) where a recto/verso dot is identified as recto/verso dot in both the ground truth and dot extracted image, and the True Negative (TN) where a recto/verso dot is classified as a non-dot in dot extracted image. The two misclassifications are the False Negative (FN) where a recto dot is classified as verso-dot in dot separated image but as a recto dot in the ground truth image, and the False

Positive (FP) where a verso dot is marked as recto dot in the dot separated image but as verso-dot in the ground truth image.

True Positive Rate (TPR) is the fraction of recto/verso dots correctly recognised as recto/verso dots respectively. True Negative Rate (TNR) is a fraction of recto/verso dots which are classified as non-dot in the dot extracted image. False Negative Rate (FNR) is the fraction of recto dots erroneously detected as verso-dots. False Positive Rate (FPR) is the fraction of verso-dots detected as recto-dots.

FNR and FPR may be attributed due to degradation of the dots which in turn is due to ageing of the Braille document. Ageing in sense as the Braille writing is read using the finger touch over the document, after multiple readings it is possible that the dots may deteriorate. Also, it may be attributed due to the surface imperfection of the Braille document and also due to the defacing of the Braille document by any means. The performance of the proposed algorithm is evaluated on manual basis with TPR, TNR, FNR and FPR. This paper also presents a new way of calculating the accuracy which differs from the previously used traditional method where in the accuracy was calculated as the ratio of the total number of correctly extracted dots to the total number of dots in the image field of view. The expression used in this paper is as follows

$$\text{Accuracy} = TPR - (FNR+FPR+TNR) * 100 \qquad (16)$$

Table.2 gives the average TPR, FNR, FPR TNR and accuracy for database DB1 and DB2. Fig.15 shows the plot of accuracy for DB1 and DB2 Fig.16 and Fig.17 give the plots of FNR, FPR and TNR for the two databases respectively. Fig.18 and Fig.19 show up the accuracy plots of all the images of DB1 and DB2.

Parameters	**Database DBI**	**Database DB2**
True Positive Rate	1.00	1.00
False Negative Rate	0.006	0.009
False Positive Rate	0.002	0.003
True Negative Rate	0.001	0.002
Average Accuracy	0.991	0.986

TABLE 2: Performance Evaluation for DB1 and DB2.

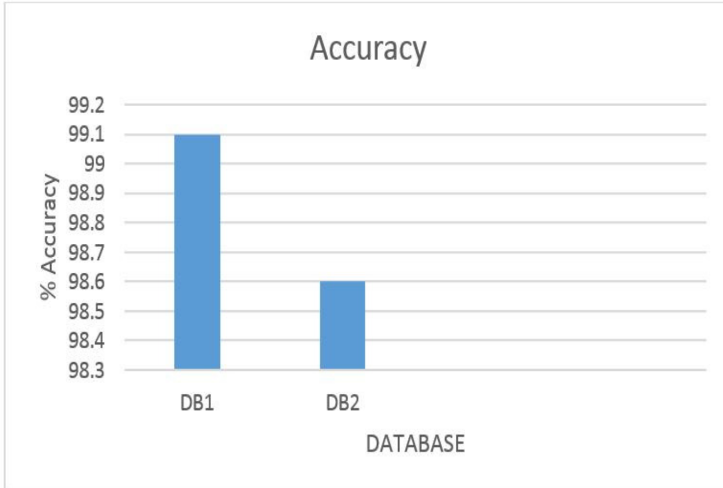


Figure 15: Graph showing Average Accuracy of DB1 and DB2.

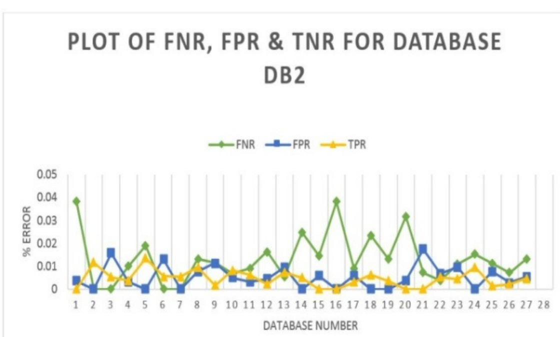
A significant drawback of the proposed technique is the merging of recto-verso dots, which is due to the fading shade patterns in the double sided Braille document being used for dot extraction. Although this transpires hardly ever, the overall error rate of the proposed system can be ascribed to the quality of the acquired image of the Braille document.

Additionally in this work, for evaluating the performance of this technique we have considered merged dots as the error and has been assigned as FNR and error due to dirty mark and blemishes as FPR.

The proposed method fails to work if the rotation angle for the document is more.

The proposed system not only possesses the excellent detection rates up to 99.1% and 98.6% for DB1 and DB2 databases respectively but it could as well be applied over any Braille document regardless to the writing grade or language.

All the experiments were done under MATLAB environment. The average execution time on an I7 machine with 8 GB of memory for separating the Recto and Verso dots of the Braille image for DB1 database is 5.24 sec and is 5.42 sec for database DB2 respectively.



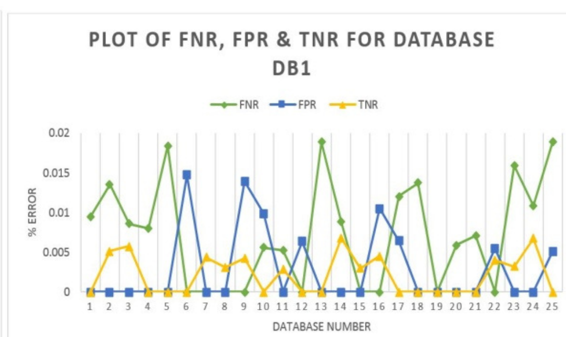
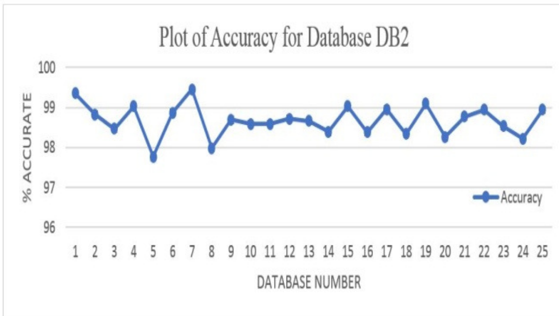


FIGURE 16: Plot of error sources for DB1.

FIGURE 17: Plot of error sources for DB2.



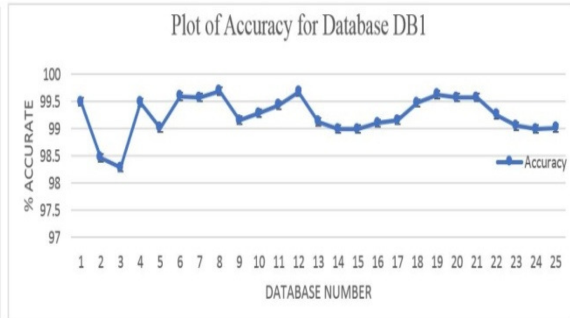


FIGURE 18: Plot of Accuracy for DB1.

FIGURE 19: Plot of Accuracy for DB2.

4. CONCLUSION

In this projected work, the separation of Recto and Verso dots from Inter-point Braille Image using a novel combination of Eight Connectivity based two-pass Connected-Component Labelling algorithm and a novel thresholding technique is presented. With an aim to recognise the Braille cells with some special cases, an adaptive grid construction technique too has been employed. The competence of this technique is 99.1% for database DB1 and 98.6% for database DB2. The proposed technique has proved very advantageous for processing the pages of Braille documents bearing the inter-dot noise attributed by the deformation during the Braille embossing process. The precincts being merging of Recto-Verso dots, rotation angle of the Braille document. These limitations are not very severe and with some additional pre-processing on the input image these noises can be effortlessly dealt with in the future work. Furthermore our chore is to deepen the study and to come up with a novel algorithm to reduce the ambiguity in recognizing the true recto and verso dots and to upsurge the recognition rate. Irrespective of the language and the writing grade, this algorithm can be applied to any Braille document. Herein the processing was executed under MATLAB implementation environment and this can be extended in real time as well.

5. ACKNOWLEDGEMENTS

We would like to thank Council of Scientific and Industrial Research (CSIR), New Delhi, for providing the financial assistance for this work under the research scheme No: **22(0613)/12/EMR-II** and also the work is supported by JSS Research Foundation, Mysore, Karnataka.

6. REFERENCES

[1] J. Mennens, L. Van Tichelen, G. Francosis and J. Engelen. “Optical recognition of Braille writing”, *Proceedings of second international conference on Document analysis and Recognition,* IEEE Oct-1993, pp 428-431.

[2] J. Mennens, L. Van Tichelen, G. Francosis and J. Engelen. “Optical Recognition of Braille Writing Using Standard Equipment”, IEEE transactions of rehabilitation engineering, Vol. 2, No. 4, Dec 1994.

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

[4] R .T. Ritchings, A. Antonacopoulos and D .Drakopoulos. “Analysis of Scanned Braille Documents”, In: Dengel, A., Spitz, A.L. (eds.): Document Analysis Systems, World Scientific Publishing Company 1995, pp. 413–421.

[5] Y. Oyama, T. Tajima, and H. Koga. “Character Recognition of Mixed Convex- Concave Braille Points and Legibility of Deteriorated Braille Points”, System and Computer in Japan, Vol. 28, No. 2, 1997.

[6] C. M. Ng, V.Ng and Y.Lau. “Regular feature extraction for recognition of Braille”, Third International conference on computational Intelligence and Multimedia Applications, 1999, pp. 302—306.

[7] 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, 2004, pp. 533-545.

[8] L. Wong, W. Abdulla and S. Hussmann. “A Software Algorithm Prototype for Optical Recognition of Embossed Braille”, 17th Conference of the International Conference in Pattern Recognition, Cambridge, UK, IEEE-2004, pp. 23–26.

[9] N. Falcon, C. M. Travieso, J. B. Alonso and M. A. Ferrer. "Image Processing Techniques for Braille writing Recognitor", EUROCAST 2005, LNCS 3643.

[10] A. Malik Al-Salman, Y. ALOHAI, M. Alkanhal and A. Airajith. “An Arabic Optical Braille Recognition System”, ICTA Apr 2007, pp.12-14.

[11] A. Malik S. Al-Salman, A. El-Zaart, Y. Al-Suhaibani, K. Al-Hokail and A. O. Al-Qabbany. “An Efficient Braille Cells Recognition”, IEEE-2010.

[12] J. Yin, L. Wang and J. Li. “The Research on Paper-mediated Braille Automatic Recognition Method”, Fifth International Conference on Frontier of Computer Science and Technology, IEEE-2010, pp 619-624.

[13] J. Li, X. Yan. “Optical Braille Character Recognition with Support-Vector Machine Classifier”, International Conference on Computer Application and System Modelling (ICCASM 2010).

[14] S. D. Al-Shamma and S. Fathi. “Arabic Braille Recognition and Transcription into Text and Voice”, 5th Cairo International Biomedical Engineering Conference Cairo, Egypt, IEEE-Dec 2010.

[15] Z. Tai, S. Cheng, P. K. Verma and Y. Zhai. "Braille document recognition using Belief Propagation", Journal of Visual Communication and Image Representation 21(7): 722-730 (2010)

[16] A. Al-Saleh, A. El-Zaart and A. Malik Al-Salman. “Dot Detection of Braille Images Using A Mixture of Beta Distributions”, 2011 Journal of Computer Science ISSN 1549-3636 pp-1749-1759.

[17] J. Bhattacharya, S.Majumder and G.Sanyal. “Automatic Inspection of Braille character: A Vision based approach”, International Journal of computer and Organization trends – volume1, Issue3 -2011, ISSN: 2249-2593, pp. 19-26

[18] M. Wajid, M. Waris Abdullah and O. Farooq. “Imprinted Braille-Character Pattern Recognition using Image Processing Techniques”, International Conference on Image Information Processing, IEEE- 2011.

[19] R. Ismail Zaghloul and T. Jameel Bani-Ata. “Braille Recognition System – With a Case Study Arabic Braille Documents”, European Journal of Scientific Research, ISSN 1450-216X Vol.62 No.1 (2011), pp. 116-122.

[20] L. Di Stefano and A. Bulgarelli. "A Simple and Efficient Connected Components Labeling Algorithm". Proceedings ICIAP, IEEE- 1999, Venice, Italy, pp. 322-327.

[21] R.C.Gonzalez and R.E. Woods. “Digital Image Processing”, 2nd edition, Prentice Hall, 2002.