

# A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters

Shreekanth.T  
Research Scholar,  
JSSRF, Mysore, India.

V.Udayashankara  
Professor, Department of IT,  
SJCE, Mysore, India.

## ABSTRACT

Optical Braille Recognition (OBR) system is computer software that automates the process of acquiring and processing images of Braille documents. It converts images of embossed Braille characters into their corresponding natural language characters. This involves few steps that include: Image Acquisition, Image pre-processing, segmentation, dot recognition and converting into computerized textual form. This review traces the earlier works carried out by the researchers on the development of OBR. In this study we try to highlight on the existing OBR solutions with special emphasis on dot recognition of the Embossed Braille Image characters.

## General Terms

OBR, SVM, GFV.

## Keywords

Braille, Inter-Point Braille, Braille Cell Dimensions, Grade I, Grade II, Grade III.

## 1. INTRODUCTION

According to the estimation of World Health Organization (WHO) in 2002, it had stated that, there were over 161 million visually impaired people worldwide [50]. As above said 15 million among these 161 million people across the globe who were blind, were from India. It had been found that 1.7% of India's population was disabled, out of which 0.22% were visually disabled. But these visually impaired people also contributed efficiently to the society. The blind community in India and those in the developing world faced a tremendous hindrance in communicating with the sighted community especially in writing, due to the difference in the script systems [13]. Consequently, they faced a stiff challenge in expressing their views in educational as well as in the mainstream employment opportunities. Until and unless there was a smooth flow of information from the visually impaired to the sighted people and vice versa, the common people would not have the necessary information about visually impaired, and due to this, the work of many visually impaired people would have remained un-notified. This would have created wide disparities between both the visually impaired and the vision society [38]. In order to make the bi-directional communication between the visually impaired and sighted community feasible, it was required to transliterate the Braille documents into the text document in the corresponding languages. Louis Braille, the founder of Braille, created a revolutionary form of communication that transcended blindness and transformed the lives of millions. He took a secret code devised for the military and saw in it the basis for written communication for the blind individuals. Braille was a

tactile format of written communication for people with low vision and blindness worldwide, since its inception by Louis Braille in 1829. Braille was a system of embossed or raised signs which used 6 dots to represent different characters, which were arranged in 3 rows and 2 columns as shown in Fig.1. Unique combinations of raised dots would represent different characters [1]. The scanned image of the real Braille was as shown in Fig.2.

A dot may be raised at any of the six positions to form sixty four combinations (including the combination in which no dots were raised). For the reference purpose, a particular combination may be described by naming the positions where dots were raised, the positions being universally numbered 1 through 3 from top to bottom on the left, and 4 through 6 from top to bottom on the right [6]. The Braille dot dimensioning have been formatized according to the tactile resolution of the Fingertips of an individual. The horizontal and vertical distance between the dots in a character, the distance between cells representing a word and the interline distance were also specified by the Library of Congress as shown in Fig.3.

Coming to the Braille size parameters the dot's height was approximately 0.02 inches (0.5 mm); the horizontal and vertical spacing between dot centers within a Braille cell was approximately 0.1 inches (2.5 mm), the blank space between dots on adjacent cells was approximately 0.15 inches (3.75 mm) horizontally and 0.2 inches (5.0 mm) vertically. A standard Braille page was 11 inches by 11.5 inches and it typically had a maximum of 40 to 43 Braille cells per line and 25 lines [22]. Braille has been adapted to write many different languages and was also used for music and Mathematical notation. The concept of Braille contractions were used to reduce the number of cells needed to represent the words and affixes, thereby reducing the size of the Braille document. The Braille system was classified into three grade levels namely, *Grade I Braille*: In this type each letter of the Braille word was fully spelled out. It was generally sufficient to learn Grade I for those who do not read and write Braille extensively. *Grade II*: It represents contracted form of the Grade I Braille. Generally the Braille books for children contain Grade II Braille. *Grade III*: It was a complicated form of Braille, mostly used as short-hand [47].

A Braille document was being embossed not only on single side but also on either of the sides. The latter was also known as Inter-point Braille. The single sided Braille documents generally consume a lot of space and hence to overcome this problem most Braille documents were printed in an inter-point. Format with the embossing done on both the sides of a page, with a slight diagonal offset to prevent the dots on each side of the Braille page from interfering with each other as shown in Fig.4. This made the translation process more

difficult as the recognition technique employed by the translation system was based on the visual perception of the Braille document but not on the tactile sensing as used by the visual impaired users [6].

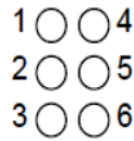


Fig 1: The Braille Cell

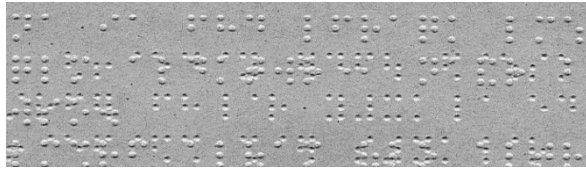


Fig 2: Scanned Image of a Braille Print

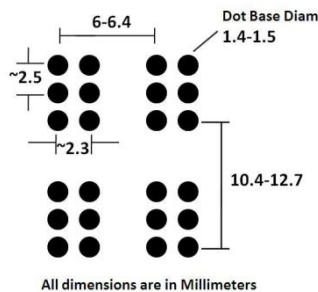


Fig 3: Braille Cell Dimensions.

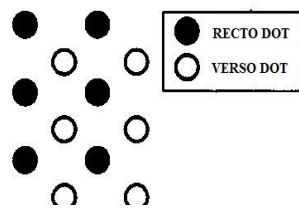


Fig 4: Inter-point Braille.

After many readings the Braille document tends to lose its embossment thereby necessitating for the need of an OBR system [15]. The fact that the thickness of the sheet used for embossing the Braille is directly proportional to the life of the document, this thus restricts the number of readings. So this makes the digitizing of Braille character essential. As a preventive measure the conversion of Braille documents into softcopies was carried out so that it could be reproduced using Braille printer at the later stages. In addition to preserving the out of print Braille texts, OBR offers many benefits to Braille users in facilitating communication and reducing the storage space. Every individual who works with the blind person and does not understand Braille i.e. the non-Braille users would be benefited from using the OBR [22]. This forms a strong platform for written communication between visually impaired and sighted people. In spite of English Braille being very popularly accepted as the Braille standard universally, different countries have adopted their own Braille notations to suit their local languages. As we stated that English Braille has been globalized, the OBR software for the same has been commercialized, but in case of other languages it was less focused. Bharati Braille was a largely unified Braille script for writing the languages of India which was developed in April 1951. The Bharati Braille Code was the collective name given to all the languages of India [48].

The Bharati Braille, the Braille scheme adopted by India and some of the Asian Countries, for instance used phonetic equivalents from the English Braille to represent the Hindi texts. Bharati Braille confirms the syllabic writing system for all the Indian languages, and hence developing the OBR was not a straight forward task for Indian languages as compared to English language, as the basic character set of all the Indian Languages was more than 64 and hence to represent a character two Braille cells were required. The Braille notations for the Hindi Braille was as shown in Fig.5 [48]. The alphabet, Phonetic representation of the alphabet and the raised dot numbers of the Braille notations were shown inside the braces respectively. Also the Braille notations for the special symbols of Bharati Braille were shown in Fig.6 [48]. Section II presents the General methodology of OBR system. Section III presents the review of Literature. Section IV provides the performance analysis of the various algorithms proposed by the researchers. In Section V, we have provided conclusions and directions for future work.

#### Alphabet: Hindi

|         |           |          |          |           |            |          |          |           |            |           |            |               |                |              |               |          |           |            |            |           |          |           |           |            |          |             |             |            |               |            |            |             |           |             |            |            |            |          |           |           |             |            |           |           |            |           |             |           |           |                |              |              |                  |           |            |
|---------|-----------|----------|----------|-----------|------------|----------|----------|-----------|------------|-----------|------------|---------------|----------------|--------------|---------------|----------|-----------|------------|------------|-----------|----------|-----------|-----------|------------|----------|-------------|-------------|------------|---------------|------------|------------|-------------|-----------|-------------|------------|------------|------------|----------|-----------|-----------|-------------|------------|-----------|-----------|------------|-----------|-------------|-----------|-----------|----------------|--------------|--------------|------------------|-----------|------------|
| अ a (1) | आ ā (345) | इ i (24) | ई ī (35) | उ u (136) | ऊ ū (1256) | ए e (26) | ऐ ē (15) | ऐ ai (34) | औ o (1346) | औ ō (135) | औ au (246) | ऋ ṛ (5, 1235) | ॠ ṝ (6, 1235) | ऌ ḷ (5, 123) | ॡ ḹ (6, 123) | क k (13) | ख kh (46) | ग g (1245) | घ gh (126) | ङ ṅ (346) | च c (14) | छ ch (16) | ज j (245) | झ jh (356) | ञ ñ (25) | ट ṭ (23456) | ठ ṭh (2456) | ड ḍ (1246) | ढ ḍh (123456) | ण ṇ (3456) | त t (2345) | थ th (1456) | द d (145) | ध dh (2346) | न n (1345) | प p (1234) | फ ph (235) | ब b (12) | भ bh (45) | म m (134) | य y (13456) | र r (1235) | ल l (123) | ळ ḷ (456) | व v (1236) | श ś (146) | ष ṣ (12346) | स s (234) | ह h (125) | क्ष ks (12345) | ज्ञ jñ (156) | झ ṝ (12456) | ढ ṭh̄ (5, 12456) | फ f (124) | ज z (1356) |
|---------|-----------|----------|----------|-----------|------------|----------|----------|-----------|------------|-----------|------------|---------------|----------------|--------------|---------------|----------|-----------|------------|------------|-----------|----------|-----------|-----------|------------|----------|-------------|-------------|------------|---------------|------------|------------|-------------|-----------|-------------|------------|------------|------------|----------|-----------|-----------|-------------|------------|-----------|-----------|------------|-----------|-------------|-----------|-----------|----------------|--------------|--------------|------------------|-----------|------------|

#### Diacritics

|                  |                    |                 |
|------------------|--------------------|-----------------|
| ◌̣ Virama (4)    | ◌̤ Visarga (6)     | ◌̥ Avagraha (2) |
| ◌̦ Anusvara (56) | ◌̧ Candrabindu (3) |                 |

#### Numbers

|                 |                  |
|-----------------|------------------|
| १ 1 (3456, 1)   | ६ 6 (3456, 124)  |
| २ 2 (3456, 12)  | ७ 7 (3456, 1245) |
| ३ 3 (3456, 14)  | ८ 8 (3456, 125)  |
| ४ 4 (3456, 145) | ९ 9 (3456, 24)   |
| ५ 5 (3456, 15)  | ० 0 (3456, 245)  |

Fig 5: Hindi Braille Notations

#### Punctuation: Indian Languages

|                              |               |  |
|------------------------------|---------------|--|
| , comma                      | (2)           |  |
| ; semicolon                  | (23)          |  |
| : colon                      | (25)          |  |
| । danda, . period, full stop | (256)         |  |
| ॥ double danda               | (256, 256)    |  |
| ? question mark              | (236)         |  |
| ! exclamation                | (235)         |  |
| ' apostrophe                 | (2)           |  |
| "..." quote                  | (236...356)   |  |
| [...] brackets               | (2356...2356) |  |
| - hyphen                     | (36)          |  |
| — dash                       | (36, 36)      |  |
| * asterisk                   | (35, 35)      |  |
| capital *                    | (6)           |  |
| number sign                  | (3456)        |  |
| italics                      | (46)          |  |

\* Capital used exclusively for English

Fig 6: Special Symbols of Bharati Braille

## 2. GENERAL METHODOLOGY

The general methodology for OBR is as shown in the Fig.7. It involves Image Acquisition; Image De-skewing, Image Pre-Processing, Segmentation, Dot recognition and Conversion. Below we shall be describing each of these steps.

### 2.1 Image Acquisition

The primitive step in the OBR system is acquiring the Images of the Embossed Braille pages, this can be accomplished using a number of distinct techniques. Researchers have CCD camera, Mobile Phones, Digital camera and Scanner's for acquiring the Braille Image [1]-[12], [14]-[44].

Number of attempts has been carried out to recognize the Braille documents by using the relatively complex setups of the camera and the oblique lighting methods. In comparison with the non-standard setup and equipment, the image captured by the camera frequently suffer from many problems such as irregular lightness, relatively low resolution etc., while in the case of using a commercially available flatbed scanner, will certainly provide a better and a cost-effective solution [15]. Using the scanner as an acquiring device, both the single and double sided documents could be scanned with varying gray level resolution and the images with high resolution could be obtained. The first approach to use a flatbed scanner to appear in the literature was of Ritchings et al. [6]. Images were scanned at 100dpi and 16 gray level values were used. The approach of Mennens et al. [2],[3] using the flatbed scanner for image acquisition appeared in the literature, but in there, high quality images which were scanned at 200dpi and at 256 gray level values were used. Experimentally it has been authenticated by the authors, that a scanning resolution between 80dpi and 200dpi was most appropriate for the Braille documents [15].

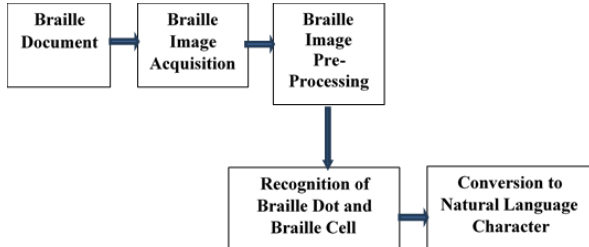


Fig 7: Block Diagram of the OBR system

### 2.2 Image De-skewing

As the arrangement of the cells in Braille document were in horizontal and vertical directions, and hence the dots in Braille document were also arranged in the similar fashion. But these dots and cell arrangements could be disturbed due to the machine dithering or human error in copying the Braille document, which would make the processing of the dots and the cell detection more difficult. Also the skewed Braille documents will affect the performance of the Braille recognition. There were many Braille documents, where cells were not arranged in horizontal and vertical directions and they are as shown in Fig.8 (a) [22]. To resolve this problem, Braille image was rotated for arranging the cells in the right directions using the information in segmented image [28]. In Fig.8 (b), the original Braille Image was rotated where all the dots were arranged in horizontal and vertical redirections. In the literature, Linear regression method, binary search algorithm, Hough transforms, Radon transforms, Discrete Fourier Transforms (DFT) have been proposed for the image skew correction. Mennens et al. [2] used deviation over the sum of rows rather than DFT to calculate the rotation angle. Their reason was the fact that

image's structure caused important and recognizable data peaks in the DFT image to lie too close to the origin and this gave an error on the calculation of the rotation angle.

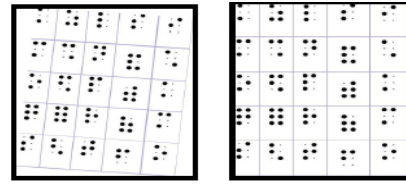


Fig 8: (a) Original Braille image where cells are rotated (b) Original Braille Image where all dots are arranged in horizontal and vertical redirections

In the case of deviation over the sum of rows, the image was slanted over an angle. Each time the image was slanted by one pixel in the vertical direction, deviation over the sum of rows was calculated, and a maximum was obtained when the dots were aligned horizontally. Hough transform was used by Antonacopoulos et al. [15] for de-skewing the skewed Braille image, while Nestor Falcon et al. [17] detected the skew angle of the scanned document by means of horizontal histogram and mass centers calculation and corrected by rotating the original image. A novel binary search algorithm was developed by Abdul Malik S et al. [22] to correct various kinds of de-skewing in the scanned images which were tilted. Zhenfei Tai et al. [25] has employed the radon transform, since they were able to transform two dimensional images with lines, into domain of possible line parameters. This was because the Braille images had inherited in them the line structures in vertical as well as horizontal directions. A fresh approach for adjusting skewness of Braille image in optical Braille recognition system based on image processing technique was proposed by Abdul Malik S [28]. This approach was able to detect the skewed Braille image and adjust it efficiently. The results showed that the method was able to correct the skewed Braille images upto 100% accuracy in a faster way. This has further increased the performance of Braille recognition system to recognize all Braille cells efficiently in Braille images. Linear regression method was used by Jia yin et al. in [31] to calculate the tilt angle and then the clockwise or anticlockwise rotation of the image was done to achieve the objectives and requirements.

### 2.3 Image Pre-Processing

In the path of obtaining a good quality digitized image of a Braille print a number of different image pre-processing techniques have been put to use. In order to meet the precision requirement of the Braille recognition this step was essential in removing the inherent noise present in the image during the image acquisition phase. Some of the previous research works done on Braille character recognition reports the use of various image processing techniques like template matching with cross correlation [2], canny edge detection [30] and image thresholding with beta distribution [37] for dot detection from the Braille image. The performance of these procedures depend on the digital image quality, and it may also give a suitable results on a very high quality image but fail to give the desired output for an ordinarily scanned image. Hence the Generalized Feature Vector (GFV) algorithm was proposed by Bhattacharya. J et al. which successfully detects the dots even for an ordinary scanned image [34].

An early effort was presented by Mennens et al. in 1993 [2] and they mainly addressed the problem of false shadows in the image caused by imperfectly flat Braille pages. This frequently happens due to tension in the paper's surface, and hence the authors suggested the subtracting of a locally

averaged image from the original image in order to obtain the enhanced image.

Hentzschel and Blenkorn [4] introduced a system for optical Braille recognition based on twin shadow technique, which subtracted two images of the same Braille page, where each image was taken under different illumination conditions. This system could locate and extract the Braille dots in Braille images as pairs of white and black spots, where each pair of white and black spots represent a single Braille dot, but if some pairs are lost, then false ones are formed.

In [8], pre-processing consisted of dual operations namely: noise filtering and edge enhancement. During image acquisition, impulse noise was introduced in the image and in relation to this noise filtering was achieved via a low-pass Gaussian filter to attenuate the high spatial frequencies from the image while at the same time, the detailed edge information of the Braille dots were being preserved. Edge detection was achieved using Sobel operator, on the other hand the objective of the edge enhancement was to sharpen the fine details of the image that was blurred.

Since there are only three classes of useful information (shadows, light areas and background), Antonopoulos has proposed a pre-processing step which has reduced the grey levels in the image to three [15]. To cope with significant variations in lightness across the whole image, a local adaptive thresholding method was introduced. This method worked by dividing the image into 32x32 pixel regions (the window size was experimentally derived) and assessed whether each region contained whole dots, highlight(s) only, shadow(s) only, or just background. The image resulting from this would have only black region (corresponding to the shadows), white region (corresponding to the highlights) and mid-grey region (the majority, corresponding to the background).

Jia Yin et al. [29] have applied the adaptive histogram equalization method to enhance the image.

Jie Li et al. [31] have used a simple technique to adjust the brightness, in which the image was partitioned into small rectangular blocks of 200X200 pixels block size, and for each block, a mean was calculated, and subtracted from the pixels in the block. The rationale behind this method was that, the mean of a block was an optimal estimation of the intensity of the background, and hence, by subtracting the mean from each block left behind only the dots and the possible noise data. It was noted that using this simple technique may cause intensity discontinuity for the pixels lying on the boundary of two adjacent blocks. A better approach would be to use a bilinear interpolation to smooth out the boundary pixels, at the price of increased computational cost.

Rawan Ismail et al. in [33] used the median filter to de-noise the image. Later on, the enhancement of dot shape is necessary. Therefore, the use of morphological operations – i.e. series of erosion and dilation operations using the disk shape- was suggested.

## **2.4 Segmentation**

The separation of dots from Braille image that can further be grouped into a cell was termed as the process of segmentation. Vivid thresholding techniques have been formulated by the researchers for the detection of the front and the back Braille dots and also to separate the Braille dots from the background. Since, Braille pages contained dots (foreground) and page (background) only, analysing a binarised image was much simpler than that of gray-scale images. For an uncompounded global threshold, the selection of the threshold value was straightforward, as the image histogram was bimodal or had

easily identifiable peaks and valleys. However, the digitized Braille page images were noisy, and there was considerable spread in the grey level values, thus the selection of threshold value was a problematic task [8]. An approach towards improving the segmentation process was to consider a histogram made up of only those pixels that were lying at or near the edges of the Braille dots. The resulting histograms contained sharper peaks and lower valleys. With an effort to determine which pixels were laying on or near an edge, the Laplacian of Gaussian operator  $\Delta^2 G$  was ordinarily applied to the entire image to identify the edge pixels. The average grey-level value of these edge pixels would then be used as the global threshold value. By experimental results, the threshold value was 100 for single sided Braille pages and 85 for double-sided pages [8].

The Otsu algorithm has been adapted by Jia Yin et al. for the purpose of thresholding [29]. Image threshold segmentation was an important image segmentation technique, wherein its crux was the threshold selection method. Threshold Segmentation was particularly useful in dealing with an image of a strong contrast of the foreground and background. Bimodal histogram method was the most commonly used method in threshold segmentation, in which if the gray scale histograms showed a clear bimodal shape, then the trough corresponding to the grey level between two peaks is selected as the threshold value. Here the focus was on Otsu algorithm. Otsu method was an adaptive threshold calculation segmentation method.

By taking into consideration the characteristics of the Braille image, a simplex global threshold was used to segment the Braille images, but this might have brought bad results too. For this reason, the partial and local threshold method to finish the segmentation was adopted, namely Braille image could be separated into numerous sub-blocks of appropriate sizes, and then be applied with the Otsu algorithm in each sub-block to segment, and so the sub-block with the size of 60x70 pixels was being utilised.

Nestor Falcon et al. [17] have used an iterative algorithm which looked for the best threshold according to the areas of Braille dots. This area criterion for thresholds selection offered an accurate way to get the optimum levels to separate black, white and gray level values.

Amany Alsaleh, et al. [37] developed a threshold value stability technique with Beta distribution to estimate the thresholds. Segmented Braille image was then used to form grids that contained the recto dots and the verso dots.

In [22], image thresholding involved the examining of each pixel in the 2-D array representing the scanned Braille document and classifying it to one of the following three categories: (a) Elements having values 32 and above were considered bright and were assigned with a threshold value +1. (b) Elements having values 23 and below were considered dark and given a threshold value -1 and (c) Elements having values between 23 and 32 were considered grey and were given a threshold value 0.

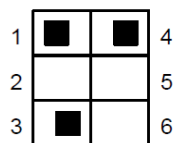
Thresholding in [37] was used to perform segmentation, which progressively subdivided an image into its constituent regions or objects. During this step, the prior calculated threshold values would be used. Each pixel in the image was checked, and if its value was less than the threshold value  $T_1$ , then this was a dark region pixel and would be assigned the value 0 in the output image and on the other hand, if its value happened to fall between the threshold values  $T_1$  and  $T_2$  then this was a background pixel and would be assigned the value 100 in the output image. The terminal case was when the pixel value was greater than the threshold value  $T_2$ , in which this

was a light region pixel and would be assigned the value 255 in the output image. The main query of the process of thresholding was the estimation of the threshold values T1 and T2. In this study, a statistical method based on Beta distribution to estimate the thresholds was put to use.

Segmenting the braille image into lines, and also segmenting the lines into character pattern was proposed by Mohd Wajid et al. in [36]. Histogram representation of the image was being used for segmentation portrayal. The 3 peaks in the vertical histogram correspond to a sole Braille line. Two peaks (adjacent, one odd and one even numbered) of horizontal histogram corresponds to a single 3 X 2 cell or an alphabet pattern.

## 2.5 Dot Recognition and Conversion

The main motive of this phase is to assemble the Braille dots into cells and to convert them into their corresponding natural characters. The various dot recognition techniques proposed by the authors in the literature will be discussed in detail in the next section. During the dot recognition process all the valid Braille dots have been detected on either sides of the document. Now in order to convert the recognised dots into their corresponding natural characters, most of the researchers have used the concept of dividing each cell into grids consisting of six parts and corresponding code for each cell was 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. Within each cell, the dot pattern was determined and was also represented by a String. These bit strings were then converted into their equivalent decimal code by using the expression: Decimal code =  $b_1 + b_2 * 2 + b_3 * 4 + b_4 * 8 + b_5 * 16 + b_6 * 32$  [22]. For example Fig.9 [6] shows the Braille cell in which recognised dots were represented by black pixels and the absence of dots were represented by white pixels. Fig.10 shows the dot position, bit strings and the equivalent decimal codes for the Braille cell shown in Fig.9. In order to retrieve the natural characters corresponding to the Braille characters, a matching algorithm was employed in which, an input decimal code generated from the processed image could be searched against the lookup table wherein the natural letters corresponding to the Braille characters are being stored. Similarly the word reconstruction was then carried out from the generated array of the decimal code using matching algorithm to match each of the input decimal arrays corresponding to a word against the Braille word lookup table [30]. If the word could not be found, then the word with the highest percentage of similarity would be selected and would be subjected to further corrections.



**Fig 9: Braille Cell after Dot recognition**

| Dot Position  | 1  | 2 | 3 | 4 | 5 | 6 |
|---------------|----|---|---|---|---|---|
| Binary number | 1  | 0 | 1 | 1 | 0 | 0 |
| Decimal code  | 13 |   |   |   |   |   |

**Fig 10: Braille code generation**

## 3. LITERATURE REVIEW

A primitive effort was presented by Mennens et al. and in their work to recognize the Braille dots, wherein a grid was placed on the image where the Braille dots could possibly be

present. In order to calculate the grids, this approach searched grid lines by generating histograms of rows and columns in five value ranges (-2,-1, 0, 1, 2). With an effort to recede the computer time and to provide an additional protection against other disturbances, firstly the histograms for rows were calculated and which were then followed by calculation of histograms for columns [2].

Hentzchel and Blenkhorn proposed a new approach, where a profile of the line of text was first generated using the column counting method. In the process of development of this segmentation algorithm, there were several such profiles being examined to determine the types and frequency of peaks and troughs which appeared on a typical line of text. In the above mentioned algorithm the characters consist of either one or two high groups of column counts. The determination of the sizes and the number of peaks in the scanned line was the main motive of this approach. A presence of the character was confirmed, if it had at least one peak and not more than four peaks of the appropriate size. If a character was detected and it began greater than half a character, but less than one character from the end of the previous character, then it was assumed that the character was a single column character with the dots in the right hand column. Otherwise the characters were assumed to start in the left column [4].

The system which was proposed by Ritchings et al. [6] was able to identify the locations of the protrusions and the depressions. This could be achieved, by exploiting the differences in grey levels in the image. These differences arouse from the reflected light during the scanning process and the shadows created by the protrusions and the depressions on the document surface. This used simple rules to distinguish between Braille dots on the two different sides. The grouping together of the dots on either sides were done in order to form characters which could be then recognised and encoded accordingly. These characters were identified based on the location of the lines of the braille character and also on the average height of the text line. Having located the lines of the Braille, the process of segmentation was performed based on the analysis of the horizontal spacing between the columns within dots. Finally the recognition of the segmented characters were then based on the above features extracted.

The dot detection module in the OBR system proposed by Oyama et al. [7] in 1997 was designed to detect both the recto and verso Braille dots. This detection was possible due to the difference in light reflectance between concave and convex dots. Dot extraction was then achieved using a 2x3 convolution mask.

In 1999 Ng. et al. [8] proposed a system that translated Braille characters to their equivalent English characters. The regular spacing between the cells and between the Braille dots in a cell was taken as a vital advantage in this proposed method. The concept of feature extraction was imbibed to extract the Braille dots from the binarised image. This proposed system has in it four distinct activities of feature extraction namely: boundary detection to highlight the dots, back-face removal to remove the dots embossed on the back side, centroid determination to locate the central point of the dots and the dot alignment to align the extracted dots with the coordinate system. On the basis of boundary coordinates and the illumination characteristics, two standard templates were then constructed to represent the recto dots and verso dots.

The robust approach proposed by Antonacopoulos et al. [15] has the ability to cope with low quality scanning and defective documents. This system worked effectively on both single-sided and the inter-point Braille documents. When considering two-sided documents, the Braille characters on both the sides were recognized from the image of only one



side of the page. In order to reduce the grey levels in the image a pre-processing step was employed and then the image was thresholded so that only three classes of regions existed namely: dark, light and background. An initial identification of Braille dots was performed by having labelled each of the different types of regions. Those dots that were not previously detected were recovered by constructing a flexible grid of possible dot locations. And lastly the Braille characters were subsequently recognized and followed by the translation into the equivalent printed text.

In 2004, Wong et al. [16] proposed an OBR system that was capable of recognizing a single sided Braille document. In this algorithm the computation time was reduced significantly by processing the image, one row at a time. The proposed system included in it the three major steps namely: half-character detection to determine the whereabouts of the character by detecting the possible dot positions, half-character recognition to determine the half characters, that the dots represent by using a probabilistic neural network and grid determination using transcription algorithm to produce a Braille text file.

The approach by Nestor Falcon et al. [17] highlights the dot detection using following concepts: dynamic thresholding, pattern detection, Braille grid creation and dot recovery using Braille grid. The research also aimed at applying iterative thresholding algorithms for classifying the grey scale image into black and white and an adaptive algorithm in order to make mesh from the detected dots. Since the distances between the dots are normalized, the adaptive algorithm builds columns in the first stage and the process begins to search for the groups of dots in the same vertical plane that follows these distances. On the basis of this criterion the protrusion and depression areas are identified.

An effort to recognize Braille image recognition using Cellular Neural Networks (CNN) was presented by Namba et al. in 2006 [20]. In this work CNN is used as constructing element of the system where the CNN for associative memory was used for Braille cell recognition. The distances of two dots were modified so that they could correspond to the size of CNN. Moreover, the values are transformed ( $0 \rightarrow 1$ ,  $255 \rightarrow -1$ ) in order to generate an input pattern of CNN. The pre-processed image was then downsized in effect to bring it close to the size of memory pattern.

A system was developed by Abdul Malik et al. in the year 2007 [22] to recognize an image of embossed Arabic Braille and this was then converted into the text format. It particularly aimed to build a fully functional Optical Arabic Braille Recognition system. It had two main tasks, first one was to recognize printed Braille cells, and the later was to convert them to regular text. Conversion of the Braille to text was a Complicated task as one cell did not represent one particular cell and also it was not a mere one to one cell mapping (It may be an alphabet, letter, digit or special characters). In addition to it the multiple cells would represent only a single symbol.

A new algorithm for Braille cells recognition using image processing technique was proposed in 2010 by Abdul Malik et al. [28]. A formation of grid was carried out which contained the Braille dots and ensured correct detection and extraction of dots. A recto dot was recognized by a light region that existed above the dark region using the image segmentation. In the similar fashion, a verso dot was identified in double sided Braille document by a light region that existed below a dark region. After having detected recto and verso dots, the standard regrouping of dots were done to recognize the Braille cells. Experimentation inferred that Braille cells were automatically identified from those grids with excellent accuracy.

Saad D Al-Shamma and Sami Fathi [30] presented Image processing technique in 2010 to convert Arabic Braille into equivalent Arabic text and also to voice. Many digital image processing have been performed on the Braille scanned document like binary conversion, edge detection, holes filling and finally image filtering before dot extraction.

Jie Li et al. proposed a system [31] which makes use of the standard distance between the Braille dots. Using the pixel value in the original image and the step size of the sliding window used in Binary Conversion Module, the relative positions of Braille dots in the binary image is calculated. With the first character's coordinate on the page determined, the grid position information can be used to determine the value of each character.

Again Jie Li et al. proposed a new system in 2010 [32] in which Braille documents were scanned into full-color image. The images were then passed through a pre-processor which converts the images into grayscale images, and performs geometric correction. Then a sliding window is applied to the image to crop out the sub images. For each sub image, Haar feature vector is calculated and then sent to Support Vector Machine SVM to decide whether the sub image contains a Braille dot, this translated the original grayscale image into a binary image. To translate Braille characters into English letters a simple searching algorithm was applied to the binary image. This method was simple, convenient, and feasible to operate, and it was also able to extract dots online in real time. The experiments showed that the method was effective and accurate for Braille extraction.

Bhattacharya et al. proposed a novel approach in 2011 [34] which used a feature detection algorithm called Generalized Feature Vector (GFV) and this successfully detected the dots even for an ordinary scanned image. To detect each cell a sliding window with a fixed interval was used to slide over the entire Braille image. The dot position for each cell was then detected using the GFV.

Histogram representation was used by Mohd Wajid et al. to recognise the Braille dots [36] and this representation for the images was given wherein the length of every peak was proportional to the number of dots present in the line. The minimum separation between the peaks was equal to the vertical separation between the dots and further, the maximum width of the peak was equal to the vertical dimension of the dot. As  $3 \times 2$  was the Braille Cell standard, wherein 3 peaks in the vertical histogram correspond to one Braille line. The end of the 3rd peak indicates the end of 1st Braille line, presence of the 4th peak gives the start of the second braille line while the 6th peak depicts the end of the 2nd line and so on. The next step after partitioning the Braille image into lines was the segmentation of the lines into alphabet patterns and the horizontal histogram for the whole image was plotted. Two peaks (adjacent one odd and one even numbered) of horizontal Then the division of alphabet pattern image in to six equal grids of size  $3 \times 2$  was performed, and the size  $3 \times 2$  was selected because this was the international standard of the Braille. Amany Alsaleh et al. developed a threshold value stability technique with Beta distribution to estimate thresholds. Segmented Braille image was then used to form a grid that contains recto dots and another one that contains verso dots. The dots in each box of the grid was checked, and if the presence of dot was found, then a text file that contains the coordinates of the center of each found dot was given as output [37].

PCA-Based Image Recognition of Braille Blocks for Guiding the Visually Handicapped has been developed by Sang-Jun Park and Dongwon Shin in the year 2012 [42]. According to the algorithm presented by them the braille blocks were

classified into four classes according to the type and orientation of the pattern on the block: vertical, right-slanted, left-slanted, and dotted stop blocks. PCA was used for generating the feature vectors of the braille block images for classification.

**Table 1. Performance Evaluation of different OBR algorithms**

| Alg<br>orit<br>hm | Image<br>acquisition<br>mode | Execution<br>speed/page   | Single/double<br>sided Braille<br>page   | Recognition Accuracy  |                         | Platform  | Braille<br>Language |
|-------------------|------------------------------|---|--|---|-------------------------|---|---------------------|
|                   |                              |   |  | Single<br>sided<br>Braille  | Double sided<br>Braille |   |                     |
| [3]               | A3 Scanner                   | 60 sec for<br>single sided<br>Braille and<br>80 sec for<br>Double<br>sided<br>Braille | On sets of<br>different<br>textures and<br>colors of<br>single/double<br>sided Braille<br>paper.Each<br>set contained<br>equal number<br>of samples. | Not specified the recognition<br>accuracy separately for single<br>and double sided Braille.<br>Group 1: (4 sets) Converted<br>with no errors.<br>Group 2: (7 sets) Average<br>error rate: .25% per character.<br>Error caused by Defects or<br>stains on the paper's surface.<br>Group 3: (1 set) Error rate not<br>Specified but it was not<br>acceptable |                         | MATLAB on VAX<br>and Apple Macintosh<br>IIfx Computer                                     | English             |
| [16]              | Scanner                      | 32.6 sec  | Single sided<br>Braille  | 99%   | Not implemented         | AMD2000+<br>CPU, 256MB RAM,<br>using a Matlab<br>implementation                           | English             |
| [8]               | Digital<br>Camera            | Not<br>specified  | On both<br>single and<br>double sided<br>Braille   | 100%  | 97%                     | Not specified   | English             |
| [15]              | Flatbed<br>Scanner           | Not<br>specified  | On both<br>single and<br>double sided<br>Braille   | 100%  | 99%                     | Not specified   | English             |
| [17]              | Flatbed<br>Scanner           | 26 sec for<br>double<br>sided<br>Braille  | On both<br>single and<br>double sided<br>Braille   | 99.9%   | 99.9%                   | Not specified   | English             |
| [33]              | Scanner                      | 25 sec  | Single sided<br>Braille  | 99%   | Not Implemented         | MATLAB<br>environment on a PC<br>with 3GHz Pentium<br>IV processor with<br>1GB of memory. | Arabic              |
| [22]              | A4 Scanner                   | Not<br>specified  | On both<br>single and<br>double sided<br>Braille   | 99%   | 99%                     | Not specified   | Arabic              |
| [20]              | Camera<br>phones             | Not<br>specified  | Single sided<br>Braille<br>numerical   | 87.9%   | Not Implemented         | Not specified   | Numerical           |
| [6]               | Flatbed<br>Scanner           | Not<br>specified  | On both<br>single and<br>double sided<br>Braille   | 98.5%   | 97.6%                   | Not specified   | English             |
| [31]              | Scanner                      | 20 minutes  | Single sided<br>Braille  | 95%   | Not implemented         | MATLAB<br>implementation  | English             |



|      |                 |          |                      |      |                 |   |        |
|------|-----------------|----------|----------------------|------|-----------------|---|--------|
| [30] | Flatbed Scanner | 32.6 sec | Single sided Braille | >99% | Not Implemented | Intel core 2duoCPU, 2GBRAM under MATLAB implementation. | Arabic |
|------|-----------------|----------|----------------------|------|-----------------|---|--------|

#### 4. PERFORMANCE EVALUATION

The comparison among the different OBR algorithms developed by the researchers throughout the Literature is provided in Table.1. Various parameters like, the Image acquisition mode, Execution speed, whether the recognition is performed on single sided or double sided Braille page, Recognition Accuracy, platform and the Braille language on which the algorithms are applied are taken into consideration. It is clear from Table.1 that the focus on optical braille recognition of double sided braille is less as compared to single sided braille. Also, majority of the work is on English Braille and next to that stands the Arabic Braille.

#### 5. CONCLUSION

Braille is understandable by visually impaired people, however, vision people need not be able to understand these codes, and thus the development of Braille recognition systems can bridge the communication gap between the visually impaired and vision people. An immense effort has been put in world-wide by different researchers to overcome this gap. During this span of study, efforts of various researchers in developing optical Braille character recognition system were explored. From this intensive survey we found that the existing literature has not addressed the issue of OBR for Hindi language and many efforts were focused on developing OBR system for single sided Braille. Making it able to detect dots on both sides of the document with only one scan will definitely have a huge advantage. OBR for Bharathi Braille is still an area that requires the contribution of many enhanced research works. Development of OBR for every language will help the blind people to communicate with the sighted world in much better way. Comparative study of one algorithm with the other could not be done as each researcher uses his own set of database, and this dataset used to perform a test is too small. Hence we are now involved in developing the standard database for Hindi Braille which will also be made available online. Continuation of efforts and research in the area of OBR is inevitable since such systems have become an essential part of many Braille document reproduction facilities.

#### 6. ACKNOWLEDGMENTS

We would like to thank Council of Scientific and Industrial Research (CSIR) India to have funded this work under the Research scheme No: 22(0613)/12/EMR-II.

#### 7. REFERENCES

- [1] J.P Dubus, M. Benjelloun, V. Devlaminck, F. Wauquier, and P. Altmayer, "Image Processing techniques to perform an autonomous System to translate relief Braille back into ink called LectoBraille", IEEE 10<sup>th</sup> International Conference in Medicine and Biology Society, New Orleans, Nov. 1988, pp 1585-84.
- [2] Jan Mennens "Optical recognition of Braille writing", IEEE 1993, pp 428-431.
- [3] Mennens, J., et al, "Optical Recognition of Braille Writing Using Standard Equipment", IEEE transactions of rehabilitation engineering, Vol. 2, No. 4, December 1994
- [4] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.
- [4] Hentzschel, T. W., and P. Blenkhorn, "An Optical Reading Systems for Embossed Braille Characters using a Twin Shadows Approach", Journal of Microcomputer Applications, pp. 341-345. 1995.
- [5] Blenkhorn, P., "A System for Converting Braille into Print", IEEE transactions on rehabilitation engineering, Vol. 3, No. 2, June 1995.
- [6] Ritchings, R.T., Antonacopoulos, A., Drakopoulos, D.: "Analysis of Scanned Braille Documents", In: Dengel, A., Spitz, A.L. (eds.): Document Analysis Systems, World Scientific Publishing Company (1995) 413–421.
- [7] Oyama, Y., 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.
- [8] C. Ng, V. Ng and Y. Lau, "Regular feature extraction for recognition of Braille". Third International conference on computational Intelligence and Multimedia Applications, 1999. ICCIMA '99 Proceedings, Page 302–306, 1999.
- [9] Tomoaki Yoshida Akihisa Ohya Shin'ichi Yuta, "Braille Block Detection for Autonomous Mobile Robot Navigation" Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 633-638
- [10] I. Murray and T. Dias. "A portable device for optically recognizing braille - part I: hardware development". In The Seventh Australian and New Zealand Intelligent Information Systems Conference 2001, pages 129–134, 2001.
- [11] I. Murray and T. Dias. "A portable device for optically recognizing braille - part II: software development". In The Seventh Australian and New Zealand Intelligent Information Systems Conference 2001, pages 141–146, 2001.
- [12] G. Morgavi and M. Morando, —"A neural network hybrid model for an optical Braille recognition", International Conference on Signal, Speech and Image Processing 2002 (ICOSSIP 2002), 2002.
- [13] Anupam Basu, Debasish Sen, Shiraj Sen and Soumen Chakraborty "An Indian Language Speech Synthesizer – Techniques and Applications" National Systems Conference, Indian Institute of Technology, Kharagpur, december 17-19, 2003
- [14] Al-Salman, A. and Al-Khalifa, H., "Towards a Computerized Arabic Braille Environment," Software Practice and Experience. Vol. 33. No.6. May 2003. pp. 497-508.
- [15] 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.

- [16] Wong, L., Abdulla, W., Hussmann, S.: “A Software Algorithm Prototype for Optical Recognition of Embossed Braille”. IEEE-2004. In: 17th Conference of the International Conference in Pattern Recognition, Cambridge, UK, pp. 23–26.
- [17] Nestor Falcon, Carlos M. Travieso, Jestis B. Alonso and Miguel A Ferrer, "Image Processing Techniques for Braille writing Recognitor", EUROCAST 2005, LNCS 3643.
- [18] Mihara, Y., A. Sugimoto, E. Shibayama and S.Takahashi. “An interactive braille-recognition system for the visually impaired based on a portable camera”. 2005
- [19] Zhang, Xuan; Ortega-Sanchez, Cesar; Murray, Iain; “Hardware-based Text-to-Braille Translator”, Eighth International ACM SIGACCESS Conference on Computers and Accessibility, ASSETS 2006, pp. 229-230.
- [20] Namba, M., Zhang, Z.: “Cellular Neural Network for Associative Memory and Its Application to Braille Image Recognition”. IEEE-2006. In: Proc. of International Joint Conference Neural Networks 2006, pp. 4716–4721
- [21] Zhang S and Yoshino K “A braille recognition system by the mobile phone with embedded camera”. ICICIC 2007: 1321-1324
- [22] Abdul Malik Al-Salman, Yosef ALOHAI, Mohammed ALKanhil and Abdulla AlRajith. “An Arabic Optical Braille Recognition System”. ICTA 2007 April 12-14
- [23] Hammamet, Tunisia. Tai, Z., Cheng, S., Verma, P.: “Braille Document Parameters Estimation for Optical Character Recognition”. Springer, Berlin(2008).
- [24] Amany Al-saleh, Ali El-Zaat, and AbdulMalik AlSalman. “Dot Detection of Optical Braille Images for Baille Cells Recognition”. K.Miesenbeger et al.(Eds.):ICCHP 2008,LNCS5105, 2008, pp.821-826.
- [25] Zhenfei Tai, Samuel Cheng, and Pramode Verma. “Braille Document Parameters Estimation for Optical Character Recognition”. G. Bebis et al. (Eds.): ISVC 2008, Part II, LNCS 5359, 2008, pp. 905–914.
- [26] Yoosefi Babadi, M., Nasihatkon, B., Azimifar, Z., Fieguth, P.: “Probabilistic estimation of Braille document parameters”. IEEE International Conference on Image Processing, 2001–2004(2009).
- [27] M. Abdelmonem, M. El-Hoseiny, A. Ali, K. Emara, H. Abdel Hafez, A. Gamal, 2009. “Dynamic Optical Braille Recognition (OBR) System” in Proceedings of the International Conference on Image Processing, Computer Vision, & Pattern Recognition IPCV’09, Las Vegas Nevada, USA, pp. 779-786
- [28] AbdulMalik S. Al-Salman, Ali El-Zaart, Yousef Al-Suhaibani, Khaled Al-Hokail, AbdulAziz O. Al-Qabbany. “An Efficient Braille Cells Recognition”. IEEE-2010.
- [29] Jia Yin, Lirong Wang and Jie Li. “The Research on Paper-mediated Braille Automatic Recognition Method” IEEE-2010 Fifth International Conference on Frontier of Computer Science and Technology, pp 619-624.
- [30] Saad D. Al-Shamma and Sami Fathi. “Arabic Braille Recognition and Transcription into Text and Voice”. IEEE-2010, 5th Cairo International Biomedical Engineering Conference Cairo, Egypt, December 16-18 2010.
- [31] Jie Li, Xiaoguang Yan, “Optical Braille Character Recognition with Support-Vector Machine Classifier”. 2010 International Conference on Computer Application and System Modeling (ICCSM 2010).
- [32] L. Jie; Y. Xiaoguang; Z. Dayong, 2010. “Optical Braille recognition with Haar wavelet features and Support-Vector Machine”, in Computer, Mechatronics, Control and Electronic Engineering International Conference (CMCE’10), Changchun, pp. 64 – 67.
- [33] Rawan Ismail Zaghloul Tomader 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.
- [34] J. Bhattacharya, S.Majumder and G.Sanyal “Automatic Inspection of Braille character: A Vision based approach”. International Journal of computer and Organisation trends – volume1, Issue3 -2011, ISSN: 2249-2593, pp. 19-26
- [35] Srinath S., C. N. Ravi Kumar, “An Insight into Optical Braille Character Recognition since its Conceptualisation”, International Journal of Computer Applications, Vol. 33, No. 6 November 2011.
- [36] Mohd Wajid, Mohammad Waris Abdullah and Omar Farooq, “Imprinted Braille-Character Pattern Recognition using Image Processing Techniques”. IEEE-2011. International Conference on Image Information Processing.
- [37] Amany Al-Saleh, Ali El-Zaart and Abdul 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.
- [38] Shumet Tadasse, A Masters thesis on “Feature Extraction and Classification schemes for Enhancing Braille Recognition system”, submitted to the school of graduate studies of Addis, June 2011.
- [39] Li Nian-feng Wang Li-rong “A kind of Braille paper automatic marking system” 2011 International Conference on Mechatronic Science, Electric Engineering and Computer August 19-22, 2011, Jilin, China
- [40] M. Yousefi , M. Famouri, B. Nasihatkon, Z. Azimifar and P. Fieguth “A robust probabilistic Braille recognition system” IJDAR (2012) 15:253–266
- [41] AbdulMalik AlSalman, Ali El-Zaart, Saleh Al-Salman, Abdu Gumaei, “A Novel Approach for Braille Images Segmentation” IEEE-2012.
- [42] Sang-Jun Park and Dongwon Shin, “PCA-Based Image Recognition of Braille Blocks for Guiding the Visually Handicapped”, Springer-2012. International Journal of Precision Engineering and Manufacturing December Vol. 13, No. 12, Pp. 2115-2120.
- [43] Zainab. Authman, Zamen F. Jebr, “Algorithm Prototype for Regular Feature Extraction of Arabic Braille Scripts” International Journal of Scientific Knowledge

Computing and Information Technology ISSN 2305-1493, February 2013. Vol. 2, No.1, pp. 20-27

- [44] 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, Volume-3, Issue-1, June 2013
- [45] Neovision s.r.o, 2004. “Optical Braille Recognition System Version 3.7 User manual”.
- [46] Censusof India website –Census 2001, Statement 5, [http://censusindia.gov.in/Census\\_Data2001/Census\\_Data\\_Online/Language/Statement5.html](http://censusindia.gov.in/Census_Data2001/Census_Data_Online/Language/Statement5.html)
- [47] “World Braille Usage”. Third Edition. Perkins. International Council on English Braille. National Library Service for the Blind and Physically Handicapped.
- [48] WorldBrailleUsage. UNESCO. National Library Service for the Blind and Physically Handicapped. Library of Congress. Washington, D.C., USA. 1990
- [49] M. Paul Lewis, Ed., Ethnologue: Languages of the World, Sixteenth Edition. SIL International, 2009.
- [50] WHO Available at: <http://www.who.int>