ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645
Indian Journal of Science and Technology, *Vol 10(33), DOI: 10.17485/ijst/2017/v10i33/112335, September 2017*

Text Translation of Scanned Hindi Document to Braille via Image Processing

Urooj Beg*, K. Parvathi and Vinod Jha

School of Electronics Engineering, KIIT University, Bhubaneswar – 751024, Odisha, India; begurooj@gmail.com, kparvati16@gmail.com, v.jha85@gmail.com

Abstract

Objective: Braille - a model to reduce the illiteracy rate among the visually challenged people. So, this paper focuses on text translation/ conversion of scanned Hindi document to Braille, a single mode of communication for visually challenged people via Image processing techniques. **Method/Analysis:** The translation of scanned Hindi text into Braille code involves the following four processes: (i) Hindi database generation of consonants and matras using image segmentation. (ii) Segmentation of test images into lines followed by words and finally into letters. This step involves three types of segmentation: (a) line segmentation (b) word segmentation (c) letter segmentation. (iii) Letter matching of segmented letters with the generated Hindi database using Principal Component Analysis (PCA). (iv) Conversion of matched letter into corresponding Braille code. The algorithm takes test documents as input and reads the letter wise and maps the letters in the corresponding Braille code. **Findings:** Three Hindi documents containing single, multiple lines are taken as test documents. These test documents are successfully converted into their corresponding Braille code using the proposed algorithm. A lot of research work has been conducted which emphasizes the conversion of scanned Braille code into the text, but the conversion of scanned text to Braille has not been attempted so far. These results are a step further in the field of Braille communication, adding a new way through which Hindi literature can easily be made available to visually impaired people for improving their knowledge and easy access to documents and books as needed. **Improvement/Application:** The results obtained will provide easy access of Hindi literature to visually challenged people as there is limited availability of Braille textbooks in the market. The memory consumption is very less as the size of each image in Hindi database is 187x128 pixels. This work can be modified for real time applications, like instant conversion of text to Braille using handheld devices.

Keywords: Hindi Braille, Hindi Data Base, Image Processing, Image Segmentation, Letter Matching, Principal Component Analysis, Text Translation

1. Introduction

India shows the largest statistic in global blindness, about 3.5 million across the country. Visually challenged people experience immense difficulties in accessing data from documents[1]. Thus, in order to bridge the communication gap between the blinds and the rest of the world, Braille system was developed in 1824[1,2]. Over time it has now become a well-known mode of communication for visually impaired people. Braille is a code, not a language[3] and these Braille codes are used for representing Braille character in any language, English, Hindi, Odia, Bengali, Kannada, Urdu. The Braille symbols are formed within a predetermined space called Braille cells[3]. Each Braille cell consists of 6 dots arranged in a matrix of dimension 3 x 2 as shown in the Figure 1.

*Author for correspondence

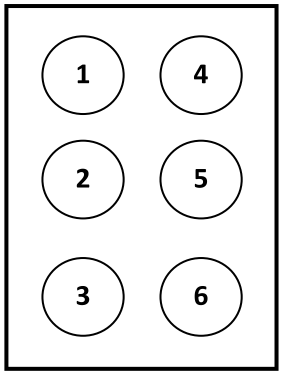

Figure 1. Braille cell.

These dots are embossed on a paper or any other material and are read by sensing the embossment by fingers. Even though the thickness of Braille sheet is also standardized, it differs from place to place. The dimensions of a Braille page are 11 inches by 11.5 inches and have a maximum of 40 to 43 Braille cells per line[2]. All the dimensions in Figure 2 are measured in inches[4].

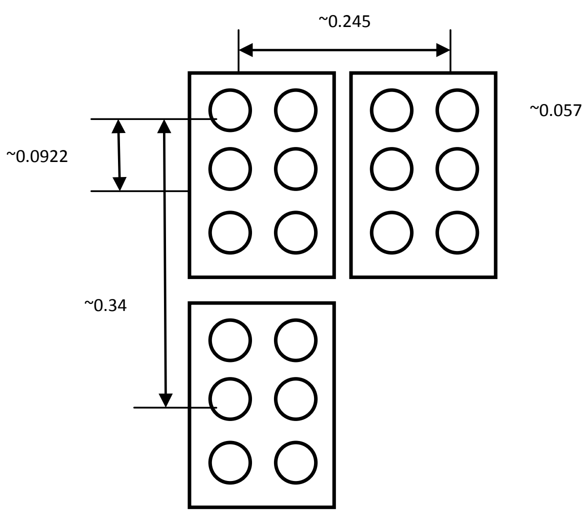

Figure 2. Dimensions of a Braille cell.

The lifetime of Braille sheet is directly proportional to the thickness of the material used for embossing[4]. Different combinations of dots may or may not be raised which corresponds to different characters of any language, so these dots combine to give 64 different combinations including the combination where no dot is raised[4,5].

$$\text{Possible pattern} = \sum_{t=0}^{6} C\binom{6}{t}$$

Although Braille cells are used throughout the world by blind people, the representation of Braille code varies with language. Braille can be classified into two types Grade 1 and Grade 2[6]. Grade 1 Braille is defined as one to one mapping where one cell represents one symbol, letter or a number and Grade 2 Braille is defined as the representation of words using a single cell. A Braille dot's dimension is specified according to the tactile resolution of person's fingertip. It, therefore, allows visually challenged people to read and write using their sense of touch rather than vision[6]. Earlier, this representation was only applicable in the English language, but now it is being used for different regional languages. In[6] the conversion of scanned Braille documents to text in English, Hindi and Tamil languages is emphasized. The mechanism involves pre-processing of the Braille cell for dot enhancement and noise reduction. The presence of raised dots is recognized and mapped to the corresponding language alphabet. It also puts forward a Number keypad for typing Braille alphabet.

Similarly, research work has been done in other regional languages. In[7] the translation of Braille code to Odia, a widely spoken language in East India. The proposed work involves cell segmentation, pattern recognition and generation of the single Odia database.

In[8] an electronic device for visually impaired people, allowing them to read messages by smart phones that work on low voltage supply is developed. This paper proposes the conversion of digital English text to Braille output in the specific vibration pattern, using 6 DC vibrators as 6 dots of a Braille cell. These vibrations are sensed by the blind people using their fingertips. This device can be connected to the smart phone via Bluetooth just like a regular add-on device.

In[9] Braille to text and speech is implemented using Field Programmable gate array Spartan 3 kit. A Braille keypad acts as an input device which consists of the different combination of cells. The given input is translated to English text in the normal domain through a program in VHDL language and further converted to speech using an algorithm. The output is displayed on an LCD connected to FPGA Spartan 3 kit.

In[10] an electronic device is developed. This device has push buttons used for entering the Braille code as input. The microcontroller reads and converts them into characters which are displayed on the LCD monitor. The microcontroller also sets the MP3 player to play the corresponding character MP3 document. This device act as a medium for learning for visually impaired people to recognize Braille code and its correct pronunciation.

A Bengali character recognition system was put forward which focuses on the translation of Braille into one of the regional languages of India, Bengali[11]. The proposed system is proficient in the extraction of Braille symbols from the document followed by its conversion to Bengali text and lastly into normalized Bengali text which is in human understandable format.

In[12] a bidirectional conversion of text to Braille for multilingual language on FPGA is proposed. This paper lacks a strong database and full-text conversion. A lot of research work done has been done in the field of conversion of Braille to English, Hindi, Odia, Kannada, Bengali, Gujarati[13] and vice versa but the conversion of scanned text into Braille is not yet attempted. This proposed scheme will improve the knowledge of visually challenged people by allowing them to access more number of Braille books. The normal textbooks can be scanned and easily converted to Braille code and it can be used in schools, libraries, institutions.

2. Proposed Algorithm

The proposed algorithm is elaborated in this section. The algorithm is divided into 4 parts: Hindi database generation, segmentation of scanned Hindi document, letter matching to the database using PCA and the letter to Braille code conversion. The flowchart of the proposed algorithm is shown in Figure 3.

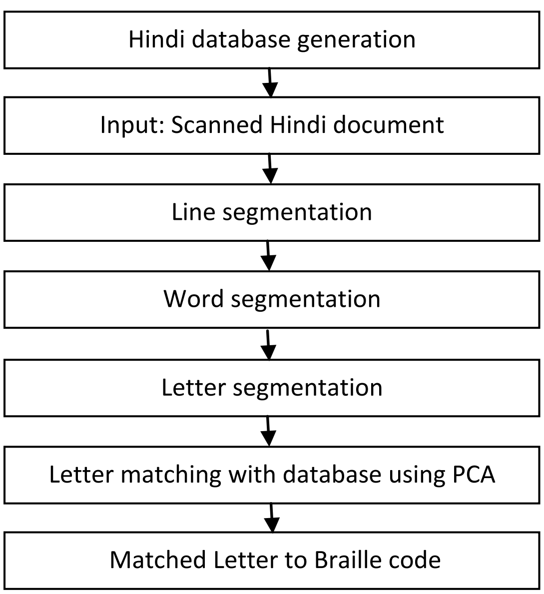

Figure 3. Flowchart of proposed algorithm.

2.1 Hindi Database Generation

A separate database of Hindi consonants and matras is generated. A scanned image containing all the Hindi alphabets along with matras is segmented to obtain individual letters. These individual Hindi letters of size 187 x 128 are stored in MATLAB to generate a Hindi database.

2.2 Segmentation

The segmentation of the scanned Hindi document is divided into 3parts: line segmentation, word segmentation, and letter segmentation. The input image undergoes above three segmentations. Firstly, the scanned Hindi document undergoes line segmentation to obtain all the Hindi text lines present in it. After obtaining the individual lines, the lines are further segmented into individual words. These segmented words further undergo segmentation to obtain individual letters. But before applying letter segmentation, the upper header line of each word is removed for proper segmentation of letters. This removal of upper header line helps to get both matras and consonants. The final output of the segmentation process gives the letter of each word which is stored in MATLAB in the same sequence as that of the input Hindi document.

2.3 Letter Matching using PCA

Principal Component Analysis (PCA) is a mathematical technique that has numerous applications in the fields of Image processing, namely pattern recognition, face recognition, computer vision and image compression, and is most used technique for pattern recognition because of its simplicity and its mathematical approach to extract relevant information from confusing data of high dimension[14,15]. The goal of PCA is to lower a multidimensional correlated data into a lower dimensional set of data consisting of uncorrelated variables while preserving the important information.

2.3.1 Mathematical Approach of Principal Component Analysis

Two-dimensional images transform into a one- dimensional vector for each input image.

Step 1: Suppose we have N input vectors having a set of image and suppose r_M represent the pixel value of images. Let Y_i is a n x 1.

$$Y_i = [r_1, r_2, r_3, \ldots\ldots\ldots r_M]^T, \ I = 1, \ldots\ldots, N \qquad (1)$$

Step 2: Calculate mean vector of each Image. This vector is named A which is defined below:

$$A = 1/N \sum_{k=1}^{m} Y_i \qquad (2)$$

The size of the vector A is nx1.

Step 3: Calculate vector B of size n x m placing in each of its columns, one of the 'm' vectors subtracted from the mean vector A.

$$B = [Y_1\text{-}A\ Y_2\text{-}A \ldots\ldots\ldots\ldots\ldots Y_N\text{-}A] \quad (3)$$

Step 4: The covariance matrix is obtained using the formula given below:

$$G=BB^T \quad (4)$$

Step 5: The covariance matrix has size n x n where n is the size of each vector Y of the input image. Since the size of the covariance matrix is large, usually a matrix H is considered.

$$H=B^TB \quad (5)$$

The size of matrix H is m x m, where m is the number of input vectors. Let v_i and e_i are eigen vectors and eigen values of H, respectively. So

$$H\, v_i = e_i v_i$$
$$(B^TB)\, v_i = e_i v_i$$
$$B(B^TB)\, v_i = B\, e_i v_i$$
$$(B.B^T)B\, v_i = e_i B v_i$$
$$GBv_i = e_i B v_i \quad (6)$$

The above equation (6) indicates that Bv_i is the eigen-vector of matrix $G=B.B^T$. Therefore, let C be the matrix which has m eigenvectors of H in each of its columns. The matrix Z contains the eigenvectors of G in its column, can be calculated as given below:

$$Z= BC \quad (7)$$

The matrix Z has size n x m because B has size n x m and matrix C has size m x m.

1). p eigenvectors of covariance matrix G are selected to obtain the principal component matrix, where p≤m. The selection is done by arranging the eigenvectors in descending order with respect to their Eigen values. The eigenvector having the highest Eigen values shows the high variance in an image.

$$Z = [\ Z_1 Z_2 Z_{3\ldots\ldots} Z_p] \quad (8)$$

The size of matrix Z is n x p.

2). Each input vector Y is projected onto the eigenvectors space.

$$S= Z^T(Y_i\text{- } B) \quad (9)$$

The vector S has size px1. The most important step of the PCA algorithm is equation number (5) because it reduces the data set of higher dimension into a lower dimension. It's basically a compression step that reduces the size of the input vector Y of n x 1to p x 1.

The K_{test} is the test image vector of size n x 1 submitted for matching task. To match the test image with the Hindi database, the following steps are followed:

1). Subtract mean vector A from $K_{test.}$ The obtained vector is projected into eigenvector space Z is done in order to obtain a new vector D.

$$D = Z^T(K_{test} - A) \quad (10)$$

2). In order to find the Matching letter, minimum Euclidean distance is calculated.

$$M= \min||\ D\text{-}S_i\,|| \quad (11)$$

2.4 Text to Braille Conversion

The Braille cell for Hindi text can be generated with the help of mapping table[4,12] where a Binary equivalent dot pattern for the corresponding Hindi text is shown. The raised dots are read as '1' and the non-raised dots are read as '0'.The rectangular Braille cell is divided into 6 equal squares and according to the standard dimensions of a Braille cell, it is calculated that each square occupies a space of approximately a 9x9 pixels. The Braille cell shown in Figure 1 is read row-wise in the order of the marked sequence. Therefore, a Braille cell consists of 27 x 18 pixels and a gap of approximately 6 pixels exists between two adjacent cells. One raised dot can be represented as a 9x9 binary image and the non-raised dots can be represented as a 9x9 binary image with all pixel value equal to 1 as shown in Figure 4 and Figure 5 respectively[4]. A Braille cell representation for dot pattern '001100' is shown in Figure 6.

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

Figure 4. Matrix representation of raised dot.

1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Figure 5. Matrix representation of non-raised dot.

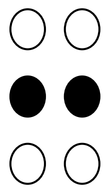

Figure 6. A Braille cell for dot pattern '001100'.

3. Results and Discussion

All the simulations of the used algorithm are done in MATLAB R2015 (b). The results of scanned Hindi image to Braille conversion are shown in a stepwise fashion.

3.1 Scanned Hindi Image-1

A sentence of font size 14 is considered as input as shown in Figure 7.This image is segmented to obtain the individual words followed by individual letters of each word.

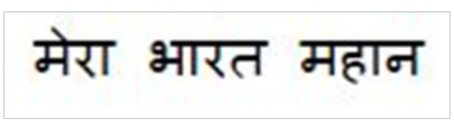

Figure 7. Scanned input image-1.

3.2 Segmentation-1

Segmentation of scanned Hindi image undergoes three processes: line segmentation, word segmentation, letter segmentation. The results of segmentation are shown in Figure 8 and Figure 9 respectively. As there is only one line in the scanned input image there is no need of Line segmentation. So, the process starts from Word segmentation which is implemented by extracting the connected components of the scanned input image shown in Figure 7. The segmented words are further processed to obtain individual letters by extracting the connected components of each word after removing the upper header line.

Figure 8. Word segmentation.

Figure 9. Letter segmentation-1.

3.3 Letter Matching with Hindi Database-1

Letter matching with Hindi database is implemented using PCA. The mathematical approach of PCA is previously discussed in section II. The segmented Hindi letters shown in Figure 9 are stored in MATLAB and these letters are the test images for the PCA algorithm. According to the algorithm, the Hindi database, as well as the test, is converted into vectors. These vectors are projected on an eigenvector space to obtain new vectors for both Hindi databases as well as a test image. Euclidean distance is calculated between the newly obtained vector of test images and Hindi database for the purpose of matching. Minimum Euclidean distance corresponds to the matched letter in the Hindi database. The position of the matched letter of the Hindi database with the test image is displayed in the command window of MATLAB as shown in Figure 10 which indicates the position of letters in the Hindi database to which the test image gets matched. For example ‘29’ represents the position of ‘म ’, ‘2’ represents the matra ‘े ’.

3.4 Hindi to Braille Conversion-1

This is the final step in the conversion of scanned Hindi images into Braille. For every position to which the segmented letter gets matched with the letter in the Hindi database as shown in Figure 10, a binary equivalent code is assigned to it. The binary equivalent code is the representation of Hindi Braille code, where '1' stands for a raised dot and '0' stands for a non-raised dot. This process of assigning a binary equivalent code is done for all the segmented letters to generate a corresponding Braille code for the scanned input image-1 as shown in Figure 11.

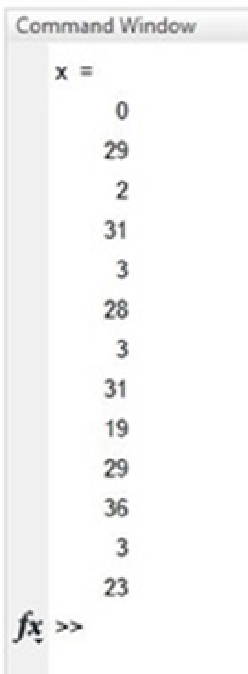

Figure 10. The position of matched letter in Hindi database for scanned image-1.

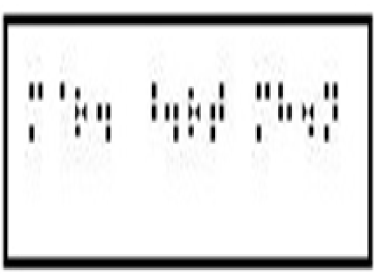

Figure 11. Braille code for scanned input image-1.

3.5 Scanned Hindi Image-2

A scanned input image consisting of more than one line undergoes the process of segmentation, letter matching and Hindi text to Braille code in order to generate the corresponding Braille code for the scanned input image shown in Figure 12. The font size of Hindi text is 14.

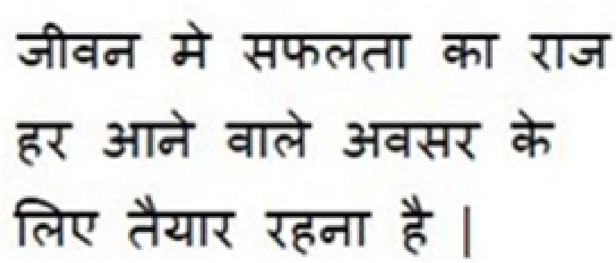

Figure 12. Scanned input image-2.

3.6 Segmentation-2

The segmentation of scanned input image in Figure 12 undergoes three types of segmentation as discussed in section 2. The result of line segmentation is shown in Figure 13. It is done by finding the white pixel rows in the input image and extracting the Hindi text lines between these rows. This is followed by word segmentation by extracting the connected components in the scanned image. The result of word segmentation is shown in Figure 14. The segmented words further undergo letter segmentation after removing the upper header line from each word. The upper header line is removed by finding the row with a maximum number of black pixels. Letter segmentation shown in Figure 15 is done by extracting all the connected components in each word. The extraction of letters becomes a lot easier after removing the upper header line.

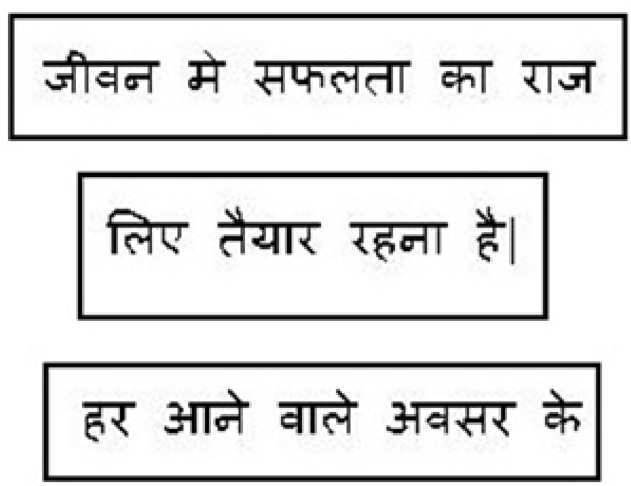

Figure 13. Line segmentation-2.

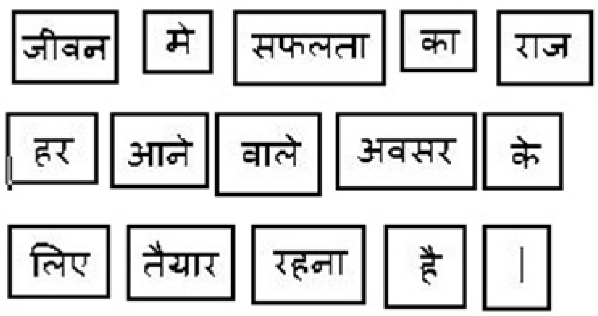

Figure 14. Word segmentation-2.

Figure 15. Letter segmentation-2.

3.7 Letter Matching with Hindi Database-2

The Segmentation of scanned Hindi image provides all the letters of the input image. These letters are used as a test image in the PCA algorithm. The Hindi database and the test letters are represented as vectors. The Projection of a test image and Hindi database vectors into eigenvector space is implemented to obtain new vectors. Euclidean distance is calculated between the newly generated vector of test images and Hindi database images for the purpose of letter matching. The minimum Euclidian distance corresponds to the matched letter in Hindi database. The position displayed below represents the position of the matched letter in Hindi database. For ex: '12', '33' is the position of the letter ' ज 'and ' व ' respectively as shown in Figure 16.

Figure 16. The position of matched letter in Hindi database for scanned image-2.

3.8 Hindi to Braille Conversion-2

The conversion of scanned Hindi images into Braille code is done with the help of Hindi mapping table[4,12]. The segmented letters are matched with the Hindi database using PCA and the position of every matched letter is displayed in the command window. For every matched letter position an equivalent Braille code is assigned. The equivalent Braille code is the binary representation of Braille code for the corresponding Hindi letter, where '1' means a raised dot and '0' means a non-raised dot. For example, अ has a binary equivalent of '100000' and its corresponding Braille code is shown in Figure 17. Similarly, all other matched letters are converted into corresponding Braille code as shown in Figure 18.

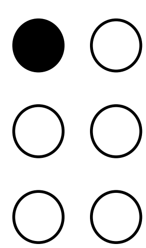

Figure 17. A Braille cell for अ.

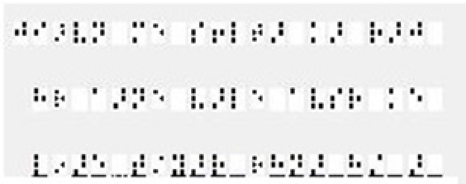

Figure 18. Hindi to Braille code conversion for scanned Hindi image-2.

4. Conclusion

This paper successfully presented the conversion of scanned Hindi text into the corresponding Braille using PCA and image segmentation technique. The first part of the paper is a generation of Hindi database where the size of every image is 187x128 pixels which require very less memory of the system.

Another advantage presented is no storage of letter to the Braille mapping table. Thus, it not only saves memory but also consumes less programming time as it uses the position of letters in the Hindi database for letter matching. The paper has attempted transcription of the scanned Hindi text to Braille, which hasn't been attempted so far. Even though the paper proposes the conversion of Matras into the Braille, but came across certain challenges like differentiating औ, ओ. ई and special letters श,ण,ए which were difficult to segment and matching with the database. So, the future scope of work is the creation of resultant Braille sheet which will further help in hardware implementation for real-time application of instant conversion of scanned Hindi text to Braille.

5. References

1. Chitte PP, Pimpalkar YA, Nair PB, Thombe SA. Braille to text and speech for cecity persons. International Journal of Research in Engineering and Technology. 2015 Jan; 4(1):263–8. Crossref.
2. Al-Shamma SD, Fathi S. Arabic braille recognition and transcription into text and voice. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 5th Cairo International Biomedical Engineering Conference, Egypt; 2010 Dec 16–18. p. 227–31. Crossref.
3. Wajid M, Abdullah MW, Farooq O. Imprinted braille-character pattern recognition using image processing techniques. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conference on Image Information processing, Shimla, India; 2011 Nov 3–5. p. 1–5. Crossref.
4. Proceedings of International Conference on Advanced Computing and Intelligent Engineering. 2015, 2.
5. Isayed S, Tahboub R. A review of optical braille recognition. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) 2nd World Symposium on Web Applications and Networking (WSWAN), Tunisia; 2015 Mar 21–23. p. 1–6. Crossref.
6. Padmavathi S, Manojna KSS, Reddy SS, Meenakshy D. Conversion of braille to text in English, Hindi and Tamil languages. International Journal of Computer Science and Applications. 2013 Jun; 3(3):19–32. Crossref.
7. Parvathi K, Samal BM, Das JK. Odia braille: text transcription via image processing. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) International Conferene on Futuristic Trends on Computational Analysis and Knowledge Management, Noida, Inida; 2015 Feb 25–27. p. 138–43. Crossref.
8. Subbu RS, Gnanaraj P. Enabling visually impaired to read messages from modern gadgets. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) Sixth International Conference on Sensing Technology, Kolkata, India; 2012 Dec 18–21. p. 502–5. Crossref.
9. Rajarapollu P, Kodolikar S, Laghate D, Khavale A. FPGA based braille to text and speech for blind persons. International Journal of Scientific and Engineering Research. 2013 Apr; 4(4):348–53.
10. Hidayat, Prafanto A. Electronic voice for set of the syllables from braille code input based on microcontroller. Indian Journal of Science and Technology. 2016 Dec; 9(47):1–8. Crossref.
11. Halder S, Hasant A, Khatun A, Bhattacharjee D, Nasipuri M. Development of Bangla Character Recognition (BCR) system for generation of bengali text from braille notation. International Journal of Innovative Technology and Exploring Engineering. 2013 Jun; 3(1):5–10.
12. Samal BM, Parvathi K, Das JK. A bidirectional text transcription of braille for odia, hindi, telugu and english via image processing on FPGA. International Journal of Research in Engineering and Technology. 2015 Jul; 4(7):483–94. Crossref.
13. Jariwala NB, Patel B. Conversion of gujarati text into braille: a review. International Journal of Innovation and Advancement in Computer Science. 2015 Jan; 4(1):59–64.
14. Khan K, Ullah R, Khan NA, Naveed K.Urdu character recognition using principal component analysis. International Journal of Computer Applications. 2012 Dec; 60(11):1–4. Crossref.
15. Prasad GK, Khan I,Chanukotimath NR, Khan F. Online handwritten character recognition system for Kannada using principal component analysis approach: for handheld devices. In the Proceedings of the Institute of Electrical and Electronics Engineers (IEEE) World Congress on Information and Communication Technologies, Trivandrum, India; 2012 Oct 30 – Nov 2. p. 675–8.