International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-3 Issue-1, June 2013

Development of a Bangla Character Recognition (BCR) System for Generation of Bengali Text from Braille Notation

Santanu Halder, Abul Hasnat, Amina Khatun, Debotosh Bhattacharjee, Mita Nasipuri

***Abstract*— This paper presents a novel Bangla Character Recognition (BCR) system which converts a Braille Document into Bengali text which is not attempted in research work so far. The system is capable of doing the extraction of Braille Characters from a Braille document followed by decoding them into Bengali characters and then the decoded Bengali characters are normalized to Bengali text which is in human-understandable form. This system can be very useful for the blind communities and the associated persons who want to know the Braille system through Bengali language. The proposed methodology has been tested on the Braille documents collected from the Jhunka Pratibondhi Aloke Niketan, West Bengal.**

***Index Terms*—Braille Cell, Braille Notation and Bengali Character, Decoding, Normalization, Bengali Text.**

I. INTRODUCTION

The Braille system, derived in 1821 by Frenchman Louis Braille, is a method that is widely used by visually impaired people. Braille refers to an approach in which text is printed on a thick sheet of paper using special symbols representing the letters of the alphabet **[1]**. A Braille Cell is composed of 6 dots arranged in three rows where each row consists of 2 columns **[1][2]**. However in the early days, Braille used 8 dots to represent a character. Later it was reduced to 6 because a person could read only 6 dots comfortably in one touch. **Fig. 1** shows such a Braille cell with 6 dots.

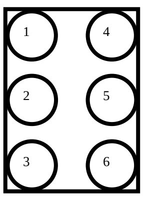


Figure 1. A Braille cell with 6 dots

Using 6 dots, 64 different characters of a language could be represented using Braille. Braille characters are embossed on a sheet and are read by sensing the embossment by fingers. The thickness of the sheet is used for embossing the Braille is directly is proportional to the life of the document.
Less the thickness, less the number of times the document could be read. Even though every language has its own Braille notation, very little research work has been observed in the literature survey **[3]**. The available optical Braille character recognition is focused only for English Braille, not for any other language **[3]** and hence most of the visually impaired people in the developing country are deprived of the modern technology that could be helpful for the advancement of their educational infrastructure and consequent joining the mainstream life through employment opportunities. Moreover people who work with the visually impaired person and are unable to read the Braille character require conversion of Braille documents into a normal language representation. Bengali is such a language for which no Braille Character recognition system has been developed so far. To give a solution in this direction this paper aims to design a Braille Character Recognition system (BCR) from Braille notation which will be useful for the blind community who wants to see the world using Bengali language. The BCR system comprises of three steps: (i) Braille Character Extraction from a Braille Document (ii) Decoding of extracted Braille Characters for Bengali Character Recognition (iii) Normalization of the decoded Bengali characters for getting the Bengali text.
This paper is organized as follows: The section II presents the brief description of the related work. Section III presents dimension of a Braille cell. Section IV depicts the Braille notation and Bengali character. Section V describes The Bangla Character Recognition (BCR) System. Section VI depicts the Extraction phase. The decoding phase is shown in Section VII. Section VIII describes the Normalization phase. The experimental result is shown in Section IX and finally section X concludes and remarks about some of the aspects analyzed in this paper.

II. RELATED WORK

In 1988, Dobus and his team designed an algorithm called Lectobraille which translates relief Braille into an equivalent printed version on paper **[3]**. In 1993 & 1994, Mennens and his team designed an optical recognition system which recognized Braille writing. It used scanner for digitizing the Braille document **[5-6]**. In 1999 Ng and his team approached the problem using boundary detection techniques to translate Braille into English or Chinese **[7]**.

Manuscript published on 30 June 2013.
***Correspondence Author(s)**
Dr. Santanu Halder, Mr. Abul Hasnat, Assistant Professor, Dept. of Computer Science and Engineering, Government College of Engineering Textile Technology, Berhampore, West Bengal, India.
Ms. Amina Khatun, Student, M. Tech, Jadavpur University, West Bengal, India.
Dr. Debotosh Bhattacharjee, Associate Professor, Dept. of Computer Science and Engineering, Jadavpur University, Kolkata, India.
Dr. M. Nasipuri, Professor, Dept. of Computer Science and Engineering, Jadavpur University, Kolkata, India.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

The paper presented an automatic system to recognize the Braille pages and convert the Braille documents into English/Chinese text for editing. They separated embossing on two sides of a sheet into two standard templates. The system took the advantage of regular spacing between Braille dots within a cell, and the regular spacing between cells.

The paper did not discuss anything about grid deformation. The capturing device used in this experiment was a digital camera, which was placed directly above the Braille page. In 2001, Murray and Dais designed a handheld device which handles the scanning as well as the translation **[8-9]**. Since the user is in control of the scanning orientation, and only a small segment is scanned at each instance, grid deformation is not a major concern and a simpler algorithm was used to yield efficient, real-time translation of Braille characters. In 2002, Morgavi and Morando published a paper where they described the use of a hybrid system using a neural network to solve the recognition problem **[10]**. The paper also provides a means of measuring accuracy in Braille recognition, and the result show the system can handle a large degree of image degradation compared with the algorithms that use more conventional and rigid image processing techniques. However no mention was made of the accuracy of the formatting conservation. Lisa Wong and team in the year 2004 presented an algorithm in which, Braille characters of 3 rows and 2 columns are divided into 2 half characters **[11]**. The two columns of a character are processed separately. In this paper, the columns were referred to as "half-characters". The proposed system consisted of three components: Half- character detection, half-character recognition, and text file transcription. The half character detections algorithm determined the whereabouts of the characters by detecting the possible dot positions. The half character recognition determines the half character that the dots represent by using a probabilistic neural network. After the half characters were recognized the grid would be determined with the text file transcription algorithm to produce a Braille text file where the formatting was preserved as much as possible. In 2005, Nestro Falcon and his team developed further more efficient techniques for Braille writing recognition using Image processing Techniques **[12]**. The paper presented the development of BrailLector, a system able to speak from Braille writing. By means of dynamic threshold, adaptive Braille grid, recovery dots techniques and TTS software (Text-To-Speech). An effort to recognize Arabic Braille recognition was found in the paper presented by Abdul Malik and his team in 2007 **[13]**. The algorithm was developed to recognize an image of embossed Arabic Braille and then convert it to text. It aimed at building fully functional Optical Arabic Braille Recognition system. The conversion of Braille to text was complicated because two or more cells would represent a single symbol, and sometimes a single cell would represent one text symbol or two or more symbols. The algorithm also tested for variations of Braille documents; skewed reversed or worn-out. Detecting the Dots of the Braille character has been developed by Amany Al and his team in 2008 **[14]**. The core of the proposed method was the use of stability of threshold with Beta distribution to initiate the process of thresholds estimation. Segmented Braille image is then used to form a grid that contains recto dots and another one that contains verso dots. Using the segmented image, Braille dots composing character on both single sided and double sided documents are automatically identified from those grids with good accuracy. Shanjun Zhang and team designed a new system that recognizes Braille characters from a photo taken by a mobile phone with embedded camera in 2010 **[15]**. Saad D Al-Shamma and Sami Fathi **[16]** presented Image processing technique in 2010 to convert Arabic Braille into equivalent Arabic and also to voice.

To conclude, no research work has been carried out to convert Braille documents into its equivalent Bengali language. The main reason behind this is the complexity involved in writing of this language which adopts syllable writing, and hence becomes a complex task to convert into equivalent form **[3]**. The BCR system is designed to give a solution in this direction.

III. DIMENSION OF A BRAILLE CELL

The dimensions of inter-dot, inter-cell and line spacing for a Braille Sheet are specified by the library of Congress. The diameter of a Braille dot is approximately 1.5 mm. The horizontal and vertical distance between the centers of two Braille dots within a character cell is approximately 2.3mm and 2.5mm respectively. The inter cell distance is approximated to 3.75mm horizontally and 5mm vertically. The standard Braille sheet is of size 11 inches wide and 11.5 inches in height. A Braille sheet contains 25 lines horizontally and 40-42 Braille cells in each line. **Fig. 2** shows the size of inter-dot, inter-cell and line spacing for a Braille sheet.

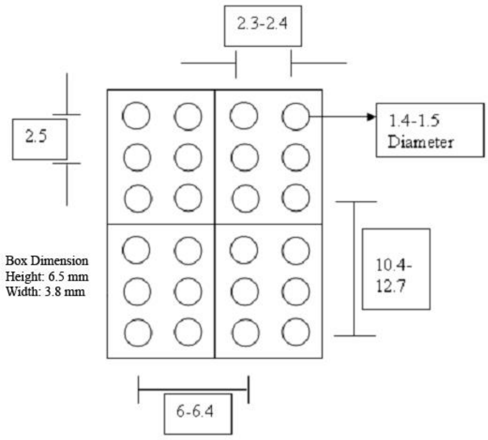


Figure 2. Dimension of inter-dot, inter-cell and line spacing of a Braille document

IV. BRAILLE NOTATION AND BENGALI CHARACTER

Text in bangla is a combination of the letters, numbers and punctuation marks like the other languages. There are some vowels and many consonants in Bengali language. They are shown in **Fig. 3**.

Vowels

অ	আ	ই	ঈ	উ	ঊ	ঋ	এ	ঐ	ও	ঔ

Consonants

ক	খ	গ	ঘ	ঙ	চ	ছ	জ	ঝ	ঞ
ট	ঠ	ড	ঢ	ণ	ত	থ	দ	ধ	ন
প	ফ	ব	ভ	ম	য	র	ল	ব	শ
ষ	স	হ	ড়	ঢ়	য়	ত্	ং	ঃ	ঁ

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Vowel Extensions

া	ি	ী	ু	ূ	ৃ	ে	ৈ	ো	ৌ

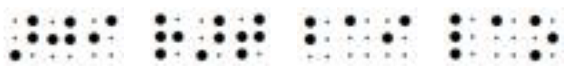
Figure 3. Alphabets in Bengali language

The Braille notations and equivalent English letters for all the Bengali alphabets are shown in **Table I**. Some Bengali letters like ঋ, ঢ় etc. are written using two Braille characters.

TABLE I: BRAILLE NOTATIONS OF BENGALI ALPHABET

Braille Notation	Bengali Letters	Braille Dots	Code	English Equivalent
	অ	1	100000	o
	আ, া	3 4 5	001110	a
	ই, ি	2 4	010100	i
	ঈ, ী	3 5	001010	I
	উ, ু	1 3 6	101001	u
	ঊ, ূ	1 2 5 6	110011	U
	ঋ, ৃ	(5) (1 2 3 5)	(000010) (111010)	rri
	এ, ে	1 5	100010	e
	ঐ, ৈ	3 4	001100	OI
	ও, ো	1 3 5	101010	O
	ঔ, ৌ	2 4 6	010101	OU
	ক	1 3	101000	k
	খ	4 6	000101	kh
	গ	1 2 4 5	110110	g
	ঘ	1 2 6	110001	gh
	ঙ	3 4 6	001101	Ng
	চ	1 3	100100	c
	ছ	1 6	100001	ch
	জ	2 4 5	010110	j
	ঝ	3 5 6	001011	Jh
	ঞ	2 4	010010	NG
	ট	2 3 4 5 6	011111	T
	ঠ	2 4 5 6	010111	Th
	ড	1 2 4 6	110101	D
	ঢ	1 2 3 4 5 6	111111	Dh
	ণ	3 4 5 6	001111	N
	ত	2 3 4 5	011110	t
	থ	1 4 5 6	100111	th
	দ	1 4 5	100110	d
	ধ	2 3 4 6	011101	dh
	ন	1 3 4 5	101110	n
	প	1 2 3 4	111100	p
	ফ	2 3 5	011010	ph
	ব	1 2	110000	b
	ভ	4 5	000110	bh
	ম	1 3 4	101100	m
	য	1 3 4 5 6	101111	z
	র	1 2 3 5	111010	r
	ল	1 2 3	111000	l
	ব	1 2	110000	b
	শ	1 4 6	100101	sh
	ষ	1 2 3 4 6	111101	Sh
	স	2 3 4	011100	s
	হ	1 2 5	110010	h
	ড়	1 2 4 5 6	110111	R
	ঢ়	(5) (1 2 4 5 6)	(000010) (110111)	Rh
	য়	2 6	010001	y
	ৎ	(5) (2 3 4 5)	(000100) (011110)	t`
	ং	5 6	000011	ng
	ঃ	6	000001	:
	ঁ	3	001000	^

There are no different Braille codes for vowel extensions in Bengali Braille. They follow the notations of vowels. If any vowel comes after any consonant in the Bengali text then the vowel is treated as vowel extension. In case of writing a vowel after any consonant, the vowel is to be followed by অ (o). **Fig. 4** depicts such a situation.



আ জ ই   র আ ত   ব অ ই   ল অ ও

আজি (today) রাত (night) বই (book) লও (take)

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Figure 4. Some examples of Bengali Braille words.

V. THE BCR SYSTEM

The block diagram of the BCR system is shown in **Fig. 5.** The proposed system first takes the input in Braille notation and then feeds the data into Extraction phase to extort each line, word and characters from the Braille document. The Decoding phase decodes the Braille characters into binary code followed by converting the code into Bengali character according to **Table I**. **So far no work has been observed in the literature that decodes a Braille documents into Bengali character.** The Normalization phase differentiates between vowels and vowel extensions. Finally the system outputs the Bengali text into normalized form.

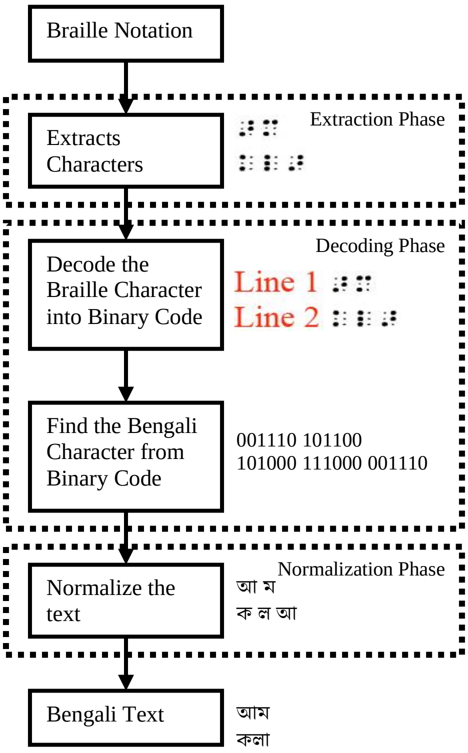


Figure 5. Block diagram of the BCR system

VI. EXTRACTION PHASES

This phase extracts the Braille character from a Braille cell using a fixed size window. The use of fixed size window does not produce an anomaly because it is already verified that the distance of inter-dot, inter-cell and line spacing are fixed as discussed in **Section III**. In spite of this fact, the spacing may vary with the image size of the document during the digitization process (using Scanner, Camera etc.) of Braille hardcopy. To solve this problem, the present system has an option to change the size of the window for a particular document. In this paper, three types of window have been considered: 1) Cell Window 2) Inter-Cell Window and (3) Line Window. Cell Window is used to detect the Braille character and its size is determined by the cell dimension of the current document. Inter-Cell Window is used to recognize the spacing between two Braille cell and its size also depends on the dimension of Inter-cell of the input Braille document. Line window senses the line spacing and the window size is calculated from the number of spacing between two lines. The space between two Braille words is recognized by a Braille cell with no dots. **Fig. 6** depicts the Cell Window, Inter-Cell Window and Line Window on a Braille document marked with Red, Blue and Green respectively.

Figure 6. A Braille document indicating Cell Window, Inter-Cell Window and Line Window

Another approach for extracting the Braille characters has been attempted by sensing the blank rows and blank columns between Braille cells. In this approach, first the document is segmented into lines (L_i) by sensing the difference between two lines. The algorithm for line extraction is shown in **Algorithm 1.**

Algorithm 1
Algorithm Extract Line (BD, T_L)
// Purpose: This algorithm extracts the lines.
// Input: BD is the Braille document. T_L is the threshold value for sensing the new line.
// Output: This algorithm outputs the lines L_i.
Begin
  Find the binary image BD_{Binary} of BD.
  Remove all objects from BD_{Binary} that have fewer than T_P pixels to remove the noise and get a new image BD_{New}.
  For each row R_i of BD_{New}
    Find the number of consecutive blank rows C_{row}.
    If $C_{row} \geq T_L$
      Mark it as new line L_i.
    End If
  End For
End of Algorithm

Now the extracted lines L_i are processed to take out the words from L_i. The steps are shown in **Algorithm 2.**

Algorithm 2
Algorithm Extract Word (L_i, T_W)
// Purpose: This algorithm extracts the words from input line.
// Input: L_i is the input line which is in binary form. T_W is the threshold value for sensing the next word.
// Output: This algorithm outputs the extracted words W_j.
Begin
  For each column C_k of L_i
    Find the number of consecutive blank columns C_{column}.
    If $C_{column} \geq T_W$
      Mark it as new word W_j.
    End If
  End For
End of Algorithm

Next the Braille characters $Char_m$ are extracted from words W_j. The extraction process is shown in **Algorithm 3.**

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Algorithm 3
Algorithm Extract Letter (W_j, T_C)
// Purpose: This algorithm extracts the Braille characters from input word.
// Input: W_j is the input word which is in binary form. T_C is the threshold value for sensing the next character.
// Output: This algorithm outputs the extracted characters $Char_m$.
Begin
    For each column C_k of W_j
        Find the number of consecutive blank columns C_{column}.
        If $C_{column} \geq T_C$
            Mark it as new character $Char_m$.
        End If
    End For
End of Algorithm

The extraction phase is implemented using Matlab 6.0. A snapshot for extraction phase is shown in **Fig. 7.**

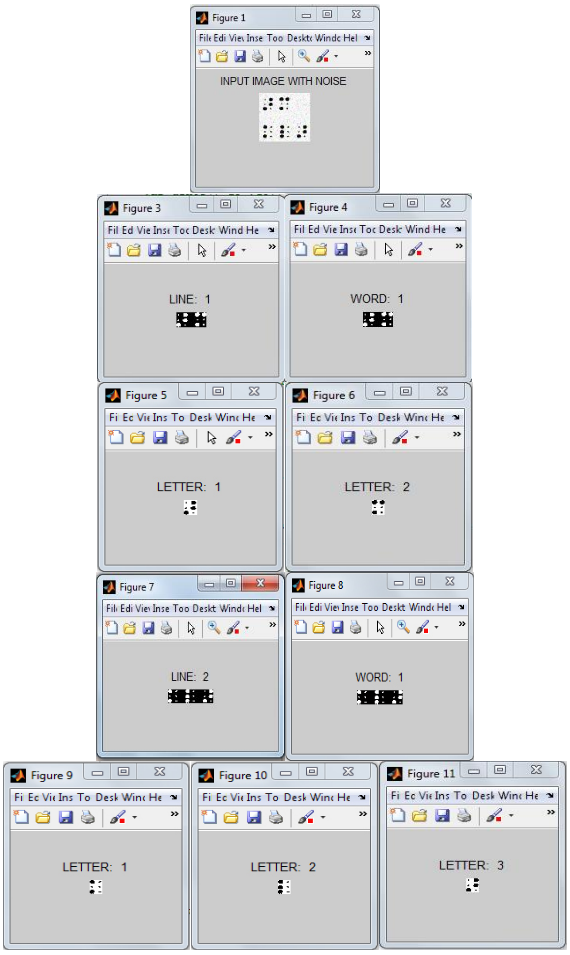


Figure 7. Snapshot of Extraction Phase

VII. DECODING PHASE

This phase is responsible for finding the Bengali character corresponding to Braille notation. It works in two steps:

1. Find the Binary code of the extracted Braille notation.
2. Get the Bengali character from the Binary code according to **Table I**.

The steps for Decoding phase is shown in **Algorithm 4**.

Algorithm 4
Algorithm Find Letter (BN_m)
// Purpose: This algorithm finds the Binary code of Braille notation followed by finding the Bengali character from the Binary code.
// Input: BN_m is the input Braille character in Binary image with size $M \times N$.
// Output: This algorithm outputs the Bengali letter corresponding to input Braille character.
Begin
    For i=1 to 6
        C[i]=0; //Initialize code C[] to 0.
    End For
    X=M/3; Y=N/2; // Divide the image BN_m into 3 rows and 2 columns.
    Find the connected components in binary image of BN_m.
    For each connected components CC_i of BN_m
        Find the center point (X_{center}, Y_{center}) of CC_i.
        A=floor(X_{center}/X)+1;
        B=floor(Y_{center}/Y);
        Index=A+B×3;
        C[Index]=1;
    End For
    Extract the Bengali character corresponding to Binary code C[] from **Table I**.
End of Algorithm

The snapshot of Decoding phase for Braille document of **Fig. 7** is shown in **Fig. 8.**

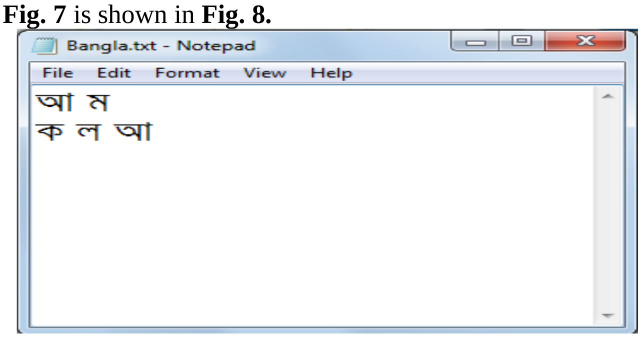


Figure 8. Snapshot of Decoding phase for Braille document of **Fig. 7**

VIII. NORMALIZATION PHASE

This phase takes care of normalization of the decoded Bengali text from the given Braille document. For example, the second line of the Braille document of **Fig. 8** should be interpreted as কলা (banana) instead of ক ল আ. More examples of such situation have been described in **Section IV**. The present system manages this phase manually. **Fig. 9** shows the normalized Bengali text of **Fig. 8.**

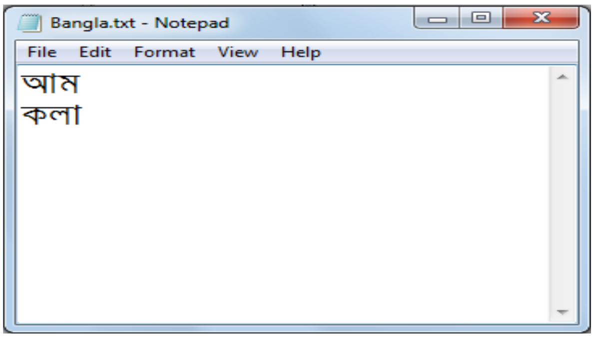


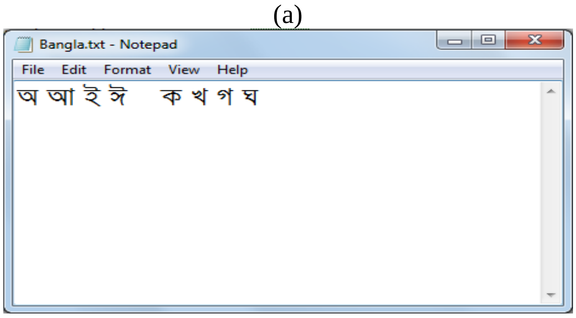
Figure 9. Snapshot of normalized Bengali text

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

IX. EXPERIMENTAL RESULTS

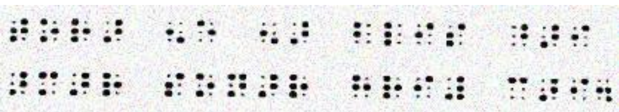
The BCR system was implemented using Matlab 6.0 on the Braille documents collected from the Jhunka Pratibondhi Aloke Niketan, West Bengal. The experimental results verify that the BCR system efficiently detects the Braille dots from digital Braille image. The Binary code generator for each Braille cell used for word reconstruction provides satisfactory result with a high accuracy and very low processing time. Some results are shown in **Fig. 10 to Fig. 11.**

(a)

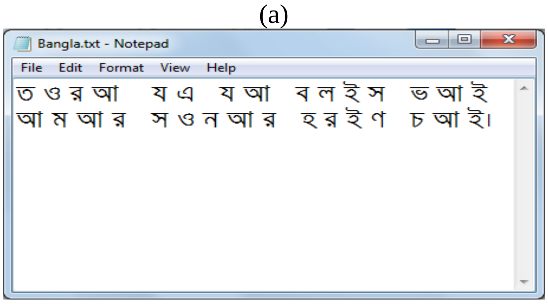


(b)

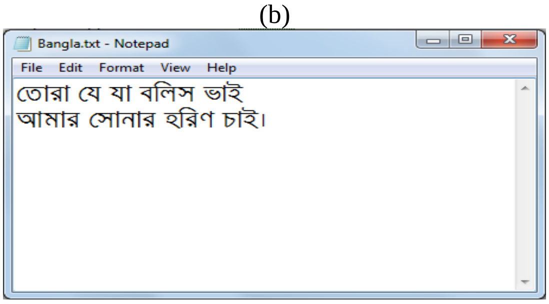
Figure 10. Example of conversion from Braille document to Bengali Characters (a) Braille Document (b) Decoded Bengali Characters



(a)



(b)



(c)

Figure 11. Example of conversion from Braille document to Bengali Text (a) Braille Document (b) Decoded Bengali Characters (c) Normalized Bengali Text.

X. CONCLUSION

This proposed BCR system is capable of doing the extraction of Braille characters from a Braille document followed by decoding them into Bengali characters and then normalization of the decoded Bengali characters into understandable Bengali text. The conversion from Braille documents into Bengali text is new one.

ACKNOWLEDGEMENT

Authors are thankful to the "Center for Microprocessor Application for Training Education and Research", "Project on Storage Retrieval and Understanding of Video for Multimedia" of Computer Science & Engineering Department, Jadavpur University, for providing infrastructural facilities during progress of the work. Two of the authors, Dr. Santanu Halder and Mr. Abul Hasnat, are thankful to Government College of Engineering and Textile Technology, Berhampore for kindly permitting him to carry on the research work.

REFERENCES

1. W.- David, A. Adler, “A Picture of Louis Braille”, New York, McGrawHill, 1999.
2. Durre, K.P., W. Tuttle, “A Universal Computer Braille Code for Literacy and Scientific Texts”, International Technology Conference, 1991.
3. 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.
4. 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 10th International Conference in Medicine and Biology Society, New Orleans, Nov. 1988, pp 1585-84.
5. Jan Mennens “*Optical recognition of Braille writing*”, IEEE 1993. pp 428-431.
6. Jan Mennens, LUC Va Tichelen, Guido Francois and Jan J Engelen., “*Optical Recognition of Braille writing using Standard Equipment*”, IEEE Transactions on Rehabilitation Engineering. Vol. 2, No.4, December 1994.
7. 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, pp 302—306, 1999.
8. I. Murray and T. Dias, “*A portable device for optically recognizing Braille - Part i: hardware development*”, Seventh Australian and New Zealand Intelligent Information Systems Conference 2001, pp 129–134, 2001.
9. I. Murray and T. Dias, “*A portable device for optically recognizing Braille - part ii: software development*”, Seventh Australian and New Zealand Intelligent Information Systems Conference 2001, pp 141–146, 2001.
10. G. Morgavi and M. Morando, “*A neural network hybrid model for an optical Braille recognitor*”, International Conference on Signal, Speech and Image Processing 2002 (ICOSSIP 2002), 2002.
11. Lisa Wong, Waleed Abdulla and Stephan Hussmann, “*A software Algorithm prototype for Optical Recognition of Embossed Braille*”, Proceeding of 17th International Conference on Pattern Recognition (ICPR 04).
12. Nestor Falcon, Carlos M. Travieso, Jestis B. Alonso and Miguel A Ferrer, “*Image Processing Techniques for Braille writing Recognition*”, EUROCAST 2005, LNCS 3643.
13. Abdul Malik Al-Salman, Yosef ALOHAI, Mohammed ALKanhal and Abdulla AIRajith, “An Arabic Optical Braille Recognition System”, ICTA 07 April 12-14 Hammamet, Tunisia.
14. Amany AL-Saleh, Ali EL-Zaart, and Abdul Malik AISalman, “*Dot Detection of Optical Braille Images for Braille Cells Recognition*”, ICCHP 2008 LNCS 5105.
15. Shanjun ZHANG and Kazuyoshi YOSHINO, “*A Braille Recognition System by the Mobile Phone with Embedded Camera*” 2010.
16. Saad D. Al-Shamma and Sami Fathi, “*Arabic Braille Recognition and Transcription into Text and Voice*”, 5th Cairo International Biomedical Engineering Conference, Cairo, Egypt, December 16-18, 2010.