

A BIDIRECTIONAL TEXT TRANSCRIPTION OF BRAILLE FOR ODIA, HINDI, TELUGU AND ENGLISH VIA IMAGE PROCESSING ON FPGA

Bijet Maynoher Samal¹, K.Parvathi², Jitendra Kumar Das³

¹PG scholar, School of Electronics Engineering, KIIT University, Odisha, India

²Professor, School of Electronics Engineering, KIIT University, Odisha, India

³Associate Professor, School of Electronics Engineering, KIIT University, Odisha, India

Abstract

Communication gap establishes an aura of unwillingness of understanding the factor behind. Basically, this is one of the common agenda for the visually challenged people. In order to bridge this gap, a proper platform of learning for both the mass and the visually challenged for any native language is emphasized in this paper through Braille pattern. Braille, a code, well known to visually challenged as their mode of communication is now converted into normal text in Odia, Hindi, Telugu and English languages using Image segmentation as the base criteria with MATLAB as its simulation field so that every person can be able to easily decode the information being conveyed by these people. The algorithm makes the best use of segmentation, histogram analysis, pattern recognition, letter arrays, data base generation with testing in software and dumping in using Spartan 3e FPGA kit. This paper also elaborates on the reverse conversion of native languages and English to Braille making the paper to be more compatible. The processing speed with efficiency and accuracy defines the effective features of this paper as a successful approach in both software and hardware.

Keywords: visually challenge, image segmentation, MATLAB, pattern recognition, Spartan 3e FPGA kit, compatible.

1. INTRODUCTION

Visually challenge is considered as the major challenge among all the differently able people who face tremendous problems in accessing the information from documents[14]. Therefore, Frenchman Louis Braille in 1821[8,18] invented the world's first binary encoding scheme for representing the writing scheme characters called as Braille. Braille is not a language but a code which is composed of letters, numbers and symbols depicted as dots, embossed on paper or any other material[8]. The result of the embossment on the sheet makes the dot to be raised as compared to the surrounding areas[21]. This enables the visually challenged people to sense the dots using the tip of their fingers as the area of the dot brings out the same area of the fingertip[20] and analyse the content being printed [21]. Each letter in Braille script comprises of six dots arranged in three rows and two columns. This arrangement is termed as a cell. Each Braille cell can be defined based on the language being used for their depiction[19]. A dot can be embossed in any of the six positions of the Braille cell individually or collectively, to form sixty four combinations of characters. Possible

pattern= $\sum_k^6 C \binom{6}{k}$ where k varies from 0 to 6[1,6,8].

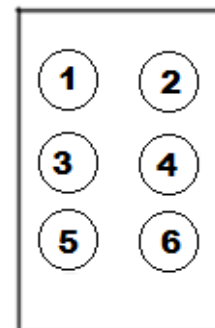


Fig -1.1: The Braille Cell [8,20]

Braille can be categorised into Grade 1 and Grade 2 [20]. Grade 1 Braille shows a one-to-one translation i.e. each Braille cell represents a single letter or a number or a punctuation mark ; Grade 2 Braille involves a single cell representing a word(in any language)[1]. But in English, Braille are of three grades, Grade 1 as a letter by letter transcription, Grade 2 as an addition of abbreviations and contractions and Grade 3 having over 300 abbreviations and contractions that reduces the Braille codes for written texts[19].

Jie Li et al. (2010) defined the Haar feature extraction and Support Vector Machine(SVM) that makes the best use in Braille character recognition[6]. The algorithm takes the scanned Braille document into full color image which is then gray scaled by the pre-processor and performs the geometric correction. The sliding window crops each image

into sub image whose Haar feature is being detected and passed on to the SVM to detect the dots thus converting the gray scale to binary image. The simple search technique is applied to the binary image to get the English characters from the Braille sheet. The Spider SVM, a MATLAB implementation followed with Radial Basis Function makes the paper more compatible. Thus, the processing time is easily reduced here by using C implementation of SVM.

Jyotsnarani Tripathy (2010) developed an efficient Optical Character Recognition(OCR) systems for ORIYA[7] for the reconstruction of the basic characters in Oriya text with different font sizes and types by using Hu's seven moments and Zernike moments. The stages involve binary conversion of the image following up with geometric moment, a pure statistical measure of pixel distribution around centre of gravity. The Zernike moment values gave both the real and imaginary parts comprising as its magnitude and thus, used in reconstruction purposes.

Munish Kumar et al. (2011) identified that various writing styles of people makes a difficult task for its recognition[12] thus creating a frontier research for an offline handwritten character recognition. The system focuses on the skeleton of the character to extract the feature and present an offline handwritten Gurmukhi character recognition efficiently by using diagonal and transition features(point distribution on the bitmap image of the character) by k-NN classifier. This paper has collected samples around offline 100 different handwritten Gurmukhi characters for partitioning and testing from a library of 3500 images being created.

Ramesh Sankara Subbu,& Pawan Gnanaraj (2012) emphasised on enabling visually challenged people for reading messages by electronic media(cell phones to email) [17] that uses a very low voltage with Bluetooth connection device makes the alphabets to vibrations in a definite format. Vibrations created by a small DC vibrator motors are placed in a 3x2 format so as to give an easy sensation to the fingertips of the challenged people. The device mentioned in this paper has been developed by open source technology(Arduino), Braille codes, vibration motors, Bluetooth Bee HC06, wireless SD shield which is a Unit-celled refreshable Braille display and converts digital text into Braille output.

Tapas Kumar Patra et al. (2012) designed simple matrix operations with sound recordings in[23] which are sampled and separated into their constituent phonetics. The separated syllables are concatenated for word reconstruction. This paper gives the idea of text to speech conversion transforming linguistic information saved as data or text into speech and is used in audio reading devices. The basic steps involved in the paper gives the recording simplification with basic editing of speech waveforms and spectrogram with time waveform analysis. The various amplitude peaks in the spectrogram defines the region of phonemes presence. Further, the phonemes are imported to the MATLAB workspace as sampled versions in column vectors. The approaches possible for the same are articulatory synthesis,

synthesis by rule and synthesis by concatenation where the paper best serves in the concatenation field. Therefore, the result finds a closer approach of the original sound of a word with the concatenated parts.

S. Padmavathi et al. (2013) made an approach in English, Hindi and Tamil mapping for Braille documents[19] pre-processed for their dot enhancement and noise reduction that gives a platform for multi language conversion of Braille. The dots from each cell by grid formation defines a number sequence that are mapped with the alphabets of the language and are spoken also by speech synthesizer. The mechanism also plays with the input by keyboard via number pad developed. But, the paper limited for the braille document that has to be aligned with the edges of the scanner. Also, the hardware section is not described here. Again, the data base created is only for three languages with unidirectional. It also gave a poor pronunciation for the Hindi and Tamil languages.

Prachi Rajarapolu et al. (2013) implemented the Braille to text or speech on FPGA Spartan 3 kit [15] providing a base region for the conversion of Braille to English language in normal domain. The Braille keypad having different combinations of cells acts as the mediator between the input image and input to the FPGA. The output of the kit is based on the decoding logic of VHDL and the alphabet generated is carried in to speech through the algorithm. Also, display through LCD is possible. The entire system is implemented through the CLB, IOB, Block RAM, Multiplier blocks, DCM of the FPGA where the simulations are being carried on in Xilinx Impact software.

S.R.Rupanagudi et al. (2014) proposed a work by introducing the Kannada braille to be converted into Speech[20]. So, it focuses on an automated segmentation of the dots, so as to make the process universal under any lighting condition and also on an unique and high speed identification system of the Braille letters exclusive for an FPGA implementation. The proposed technique uses a digital color image of a Kannada Braille sheet procured at a distance of 5 centimetres. The RGB (Red, green and Blue) color space image then is further color converted to a single color space for faster processing. The process then goes for Thresholding and Segmentation. Identification of a letter can be accomplished based on finding out which dot is highlighted in the Braille letter under consideration. The same algorithm being implemented on an FPGA made the code to be synthesized into a priority encoder having series of several multiplexors. Finally, the algorithm is implemented using Verilog HDL language on a Xilinx Spartan 3e series of FPGA. But, the problem in this is the threshold range which varied based on different lighting conditions. Again, the data base developed is only for the Kannada language with one directional(Braille to text). The hardware section gave idea of text conversion but could not emphasise on speech translation as mentioned in its topic.

K. Parvathi et al. elaborated a data base for Odia to text[8] bringing out a solution for the conversion of Braille to text and vice versa for a regional language, Odia using the single data base. The algorithm has taken up the cell segmentation, histogram analysis, dot pattern recognition, letter pattern recognition and final summation of all the data in a library called Data Base. This paper gives one of the best approaches in a Grade 1 braille conversion where the accuracy is also a factor of effectiveness of the work.

2. EXPERIMENTAL APPROACH

As the Odia braille sheet is acquired[8], similarly, the Hindi, Telugu and English braille sheets are obtained as shown in Fig.2.1, Fig.2.2 and Fig.2.3 respectively. The main aim is to have a Braille to native language conversion and vice versa in form of text using image processing technique. A standard pattern of alphabets for Braille only differs by mapping from language to language where appropriate mapping for each language gives the respective alphabets. The data base of Odia language being created[8] undergoes certain modifications in the letter pattern being described in the next section which defines it to be applicable for the bidirectional conversion in any of the native languages whose Braille sheets are available. The data base developed is then dumped into Spartan3e FPGA to display the output of the dot pattern along with letters via MATLAB making hardware incorporation an effective one. Further, using the similar concept of the data base developed, conversion of Braille to the international language, English and the reverse is also explained in the next section.

अ	आ	इ	ई	उ	ऊ	ए	ऐ	ओ	औ
क	ख	ग	घ	ङ	च	छ	ज	झ	ञ
ट	ठ	ड	ढ	ण	त	थ	द	ध	न
प	फ	ब	भ	म	य	र	ळ	व	ल
श	ष	स	ह	क्ष	ज्ञ	्	ड	अः	य

Fig -2.1: Hindi Braille Sheet[26]

అ	ఆ	ఇ	ఈ	ఉ	ఊ	ఎ	ఐ	ఒ	ఔ
క	ఖ	గ	ఘ	ఙ	చ	ఛ	జ	ఝ	ఞ
ట	ఠ	డ	ఢ	ణ	త	థ	ద	ధ	న
ప	ఫ	బ	భ	మ	య	ర	ల	వ	ళ
శ	ష	స	హ	క్ష	జ్ఞ	్	డ	అః	య

Fig -2.2: Telugu Braille Sheet[26]

a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	x	y	z	w	,	;	:	.

Fig -2.3: English Braille Sheet[26]

3. PROPOSED ALGORITHM

The basic approach for the cell segmentation and dot pattern recognition[8] follows up the similar criteria for other native languages. The flow chart of the algorithm developed for a modified data base is shown in Fig. 3 and has been elaborated below. Here, the Braille sheet has been segmented by image processing and the result is shown in form of text in any regional language.

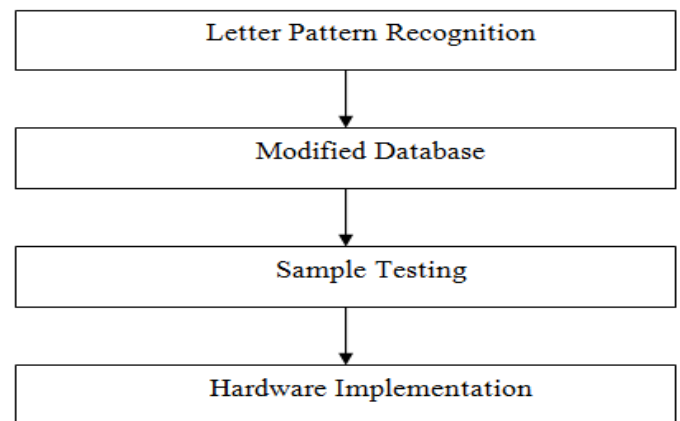


Fig -3: Steps involved in Braille to Native languages and English

3.1. Letter Pattern Recognition

3.1.1. Hindi Letter Pattern

The Hindi alphabetical letters are re-arranged in the arrays with respect to number of dots and dot patterns in sequence(Table 1).

Table -1: Hindi Letter Arrays

Number of dots	Hindi Letters									
	1	अ	इ	उ	ः					
2	ब	ख	ग	घ	ङ	च	छ	ज	झ	ञ
3	ट	ठ	ड	ढ	ण	त	थ	द	ध	न
4	प	फ	ब	भ	म	य	र	ळ	ल	व
5	श	ष	स	ह	क्ष					
6	ह									

3.1.2. Telugu Letter Pattern

Similarly, the Telugu alphabetical letters are re-arranged in the arrays with respect to number of dots and dot patterns in sequence as shown in the Table 2.

Table -2 : Telugu Letter Arrays

Number of dots	Telugu Letters															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	అ															
2	ఇ	ఈ	ఐ													
3	ఆ	ఊ	ఋ	ౠ	అ	ఇ	ఈ	ఐ	ఓ	ఔ						
4	క	ఖ	గ	ఘ	ఙ	చ	ఛ	ఞ	ట	ఠ	డ	ణ	త	థ	ద	ధ
5	ప	ఫ	బ	భ	మ	య	ర	ల	ళ							
6	న															

3.2. Modified Data Base Storage for Odia, Hindi, Telugu

The modified data base for Odia, Hindi and Telugu is shown in Table 3 which can be used to follow up for all the native languages. This data base acts a storage place or a library that can be used for retrieval of data for bidirectional conversions in the mentioned languages.

Table -3: Data Base Table for Odia, Hindi and Telugu

Braille Cells	Number of dots	Dot pattern	Odia letters	Hindi letters	Telugu letters
	1	100000	ଅ	अ	ఆ
		010000	ఇ	इ	ఐ
		000001	ఈ	ः	ఊ

	2	011000	ఖ	ख	య
		000110	ఁ	ख	ఁ
		100100	ఠ	प	ల
		010010	ఠ	प	ల
		100010	ఠ	क	అ
		010001	ఠ	ख	అ
		110000	ఠ	व	ఁ
		100001	ఠ	उ	ఁ
		001100	ఠ	व	अ
		101000	ఠ	ब	अ
		010100	ఠ	म	अ
		001001	ఠ	य	ఁ
	3	010110	అ	आ	ఊ
		100011	ఠ	उ	ఊ
		100110	ఠ	ओ	ఊ
		011001	ఠ	ओ	ఊ
		101001	ఠ	व	ఁ
		010011	ఠ	उ	ఁ
		011100	ఠ	ज	ఁ
		000111	ఠ	झ	ఁ
		110100	ఠ	द	ఁ
		001110	ఠ	फ	ఁ
		110010	ఠ	म	ఁ
		101010	ఠ	उ	ఁ
	010101	ఠ	उ	ఁ	
	110001	ఠ	श	ఁ	
	011010	ఠ	स	ఁ	
	101100	ఠ	ल	ఁ	
	100101	ఠ	ख	ఁ	
	4	101101	ఠ	उ	ఁ
		111100	ఠ	ग	ఁ
		011101	ఠ	उ	ఁ
		111001	ఠ	उ	ఁ
		010111	ఠ	ग	ఁ

	4	011110	ଉ	ତ	ଓ
		110101	ଥ	ଧ	ଝ
		011011	ଢ	ଧ	ଝ
		110110	ଞ	ନ	ଠ
		111010	ପ	ଫ	ଓ
		101110	ର	ସ	ଠ
		101011	ଞ	ଵ	ଓ
	5	011111	ଠ	ଡ	ଓ
		111101	ଢ	ଢ	ଓ
		110111	ସ	ଫ	ଓ
		111011	ସ	ଫ	ଓ
		111110	ସ	ଫ	ଓ
	6	111111	ଢ	ଢ	ଓ

But, for the English language, the developed data base can be applied with some modifications as it has got 26 alphabets only, with some special characters(Fig. 2.3). So, the extension of the data base to a new table(Table 4) for English is discussed below.

3.2.1. English Letter Pattern Recognition

The basic logic used in the previous data base for the native languages remains unaltered for the English braille conversion. Only the letter rearrangement according to the number of dots is the main criteria for the modification of the data base. Table 4 gives the brief idea for the same.

Table -4: English Letter Arrays

Number of dots	English letters									
1	a	,								
2	b	c	e	i	k	;				
3	d	f	h	j	l	m	o	s	u	.
4	g	n	p	r	t	v	w	x	z	
5	q	y								
6	Space									

3.2.2. Data Base Storage for English

The dot pattern, the corresponding letters and the cells are now stored in a table(Table 5) and this table can be used to retrieve the data for braille to English and vice versa.

Table 5: Data Base Table for English

Braille Cells	Number of Dots	Dot Pattern	English Letters	
	1	1 0 0 0 0 0	a	
		0 0 1 0 0 0	,	
	2	1 0 1 0 0 0	b	
		1 1 0 0 0 0	c	
		1 0 0 1 0 0	e	
		0 1 1 0 0 0	i	
		1 0 0 0 1 0	k	
		0 0 1 0 1 0	;	
		3	1 1 0 1 0 0	d
	1 1 1 0 0 0		f	
	1 0 1 1 0 0		h	
	0 1 1 1 0 0		j	
	1 0 1 0 1 0		l	
	1 1 0 0 1 0		m	
	1 0 0 1 1 0		o	
	0 1 1 0 1 0		s	
	1 0 0 0 1 1		u	
	0 0 1 1 0 1		.	
	4	1 1 1 1 0 0	g	
		1 1 0 1 1 0	n	
		1 1 1 0 1 0	p	
		1 0 1 1 1 0	r	
		0 1 1 1 1 0	t	
		1 0 1 0 1 1	v	
		0 1 1 1 0 1	w	
		1 1 0 0 1 1	x	
		5	1 0 0 1 1 1	z
			1 1 1 1 1 0	q
	1 1 0 1 1 1		y	
	6	0 0 0 0 0 0	space	

3.3. Sample Testing and Hardware Incorporation

The library created is being verified by testing different samples of braille for their conversion. The data base created is now dumped into FPGA so as to verify the performance of the simulation in software and hardware and develop an interface platform to read the braille documents in either of the above mentioned languages.

4. RESULTS AND DISCUSSIONS

The number of dots and dot pattern serves as the reason for the mapping in Hindi, Telugu and English Braille sheet making the data base for dot pattern[8] to be align with that of Table 3 and Table 5(with some modifications).

4.1. For Braille to Odia/Hindi/Telugu Letter/Word

The user can give the choice of input language out of Odia, Hindi or Telugu along with the either of the random, single number or a string of numbers from 1 to 50 in the workspace as input as given in Fig. 4.1 for Hindi and Fig. 4.3 for Telugu. The length of the string determined makes the string to read element-wise and extract the respective Braille cell given as input. Applying the algorithm process[1], the data about the unknown dot patterns is used to verify with the data of the particular row of the Developed Data Base(Table 3) and based on the pattern matching, the respective Odia/Hindi/Telugu letter/word is being displayed. The same is followed up for all the inputs in string.

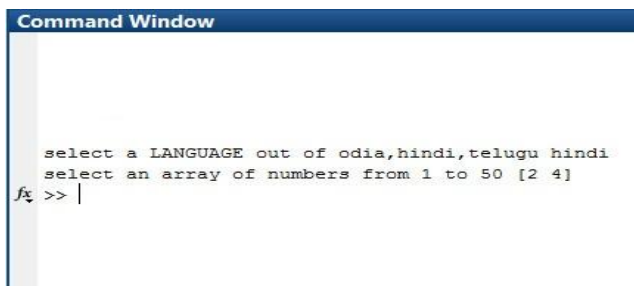


Fig -4.1: Hindi Braille as Input in workspace

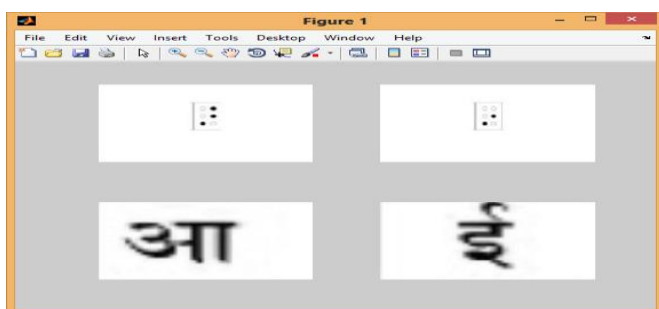


Fig -4.2: Output of Braille to Hindi pattern Conversion

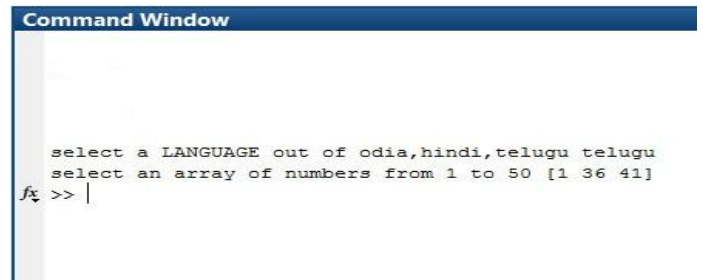


Fig -4.3: Telugu Braille as Input in workspace

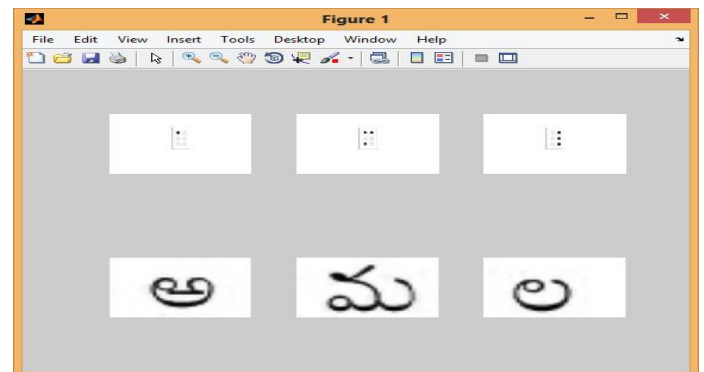


Fig -4.4: Output of Braille to Telugu pattern Conversion

4.2. For Odia/Hindi/Telugu Letter/Word to Braille

The user can give the choice of input language out of Odia, Hindi or Telugu along with the either of the random, single number or a string of numbers from 1 to 50 in the workspace as input as given in Fig. 4.5 for Hindi and Fig. 4.7 for Telugu. The length of the string determined makes the string to read element-wise and extract the respective letter pattern given as input. Applying the algorithm process[8], the Developed Data Base makes a one-to-one matching of the given letter as input with the Braille cell as output. The same is followed up for all the inputs in string.

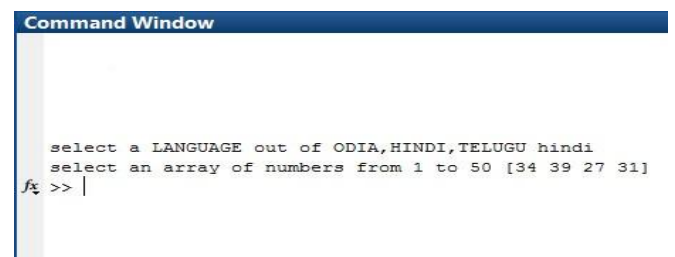


Fig -4.5: Hindi pattern as Input in workspace

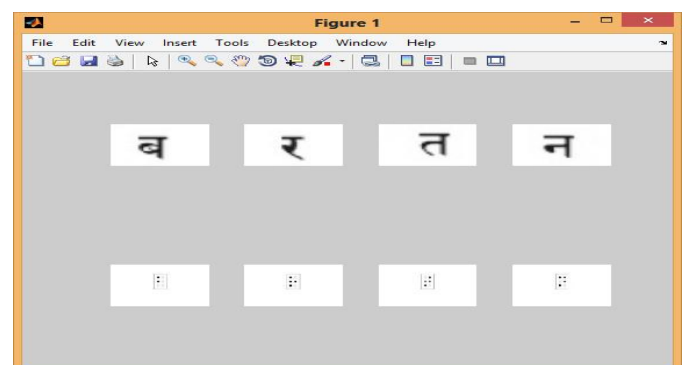


Fig -4.6: Output of Hindi pattern to Braille Conversion

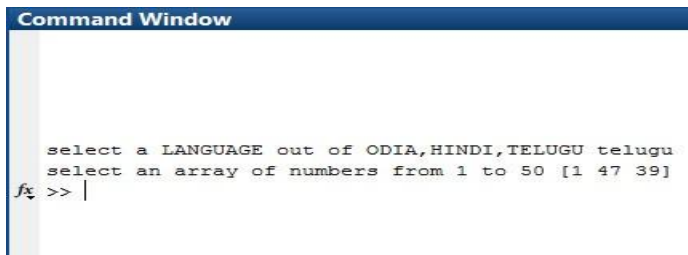


Fig -4.7: Telugu pattern as Input in workspace



Fig -4.11: Output of Braille to English pattern Conversion

The output of the conversion(Fig. 4.11) is as such because the coding is developed in such a fashion that the cells taken as input is extracted in the MATLAB figure window but its corresponding English conversion will be displayed in the MATLAB workspace. This makes the data base to be more user friendly and impactful.

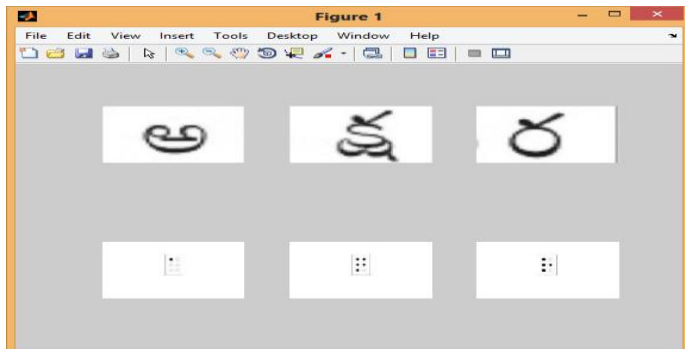


Fig -4.8: Output of Telugu pattern to Braille Conversion

4.4. For English Letter/Word to Braille

4.3. For Braille to English Letter/Word

Either of the random, single number or a string of numbers from 1 to 30 in the workspace as input as given in Fig. 4.9. The length of the string determined makes the string to read element-wise and extract the respective Braille cell given as input(Fig. 4.10). Applying the algorithm process[8], the data about the unknown dot patterns is used to verify with the data of the particular row of the Developed Data Base(Table 5) and based on the pattern matching, the respective letter/word is being displayed in workspace again(Fig. 4.11). The same is followed up for all the inputs in string.

Either of the random, single character or a string of characters or a sentence is given as the input in the workspace directly from the keyboard (Fig. 4.12). The length of the string determined then allows for reading the string element-wise. Applying the algorithm process[8], the Developed Data Base(Table 5) gets a one-to-one matching of the given character as input with the Braille cell as output(Fig. 4.13). The same is followed up for all the inputs in the string.

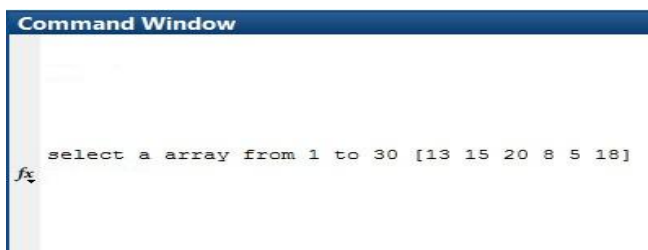


Fig -4.9: English Braille as Input in workspace

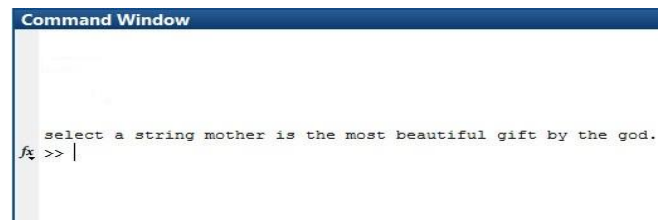


Fig -4.12: English Pattern in form of a Sentence as Input in workspace

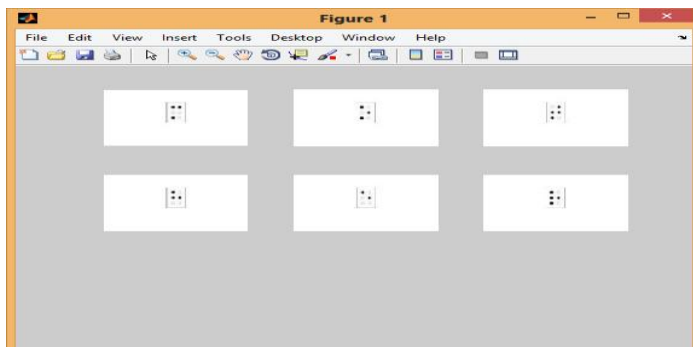


Fig -4.10: Extracted Braille cells taken as input

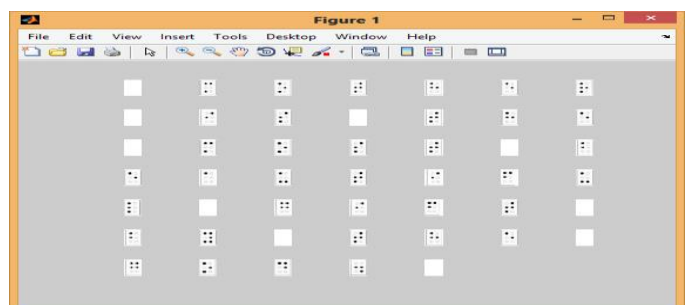


Fig -4.13: Output of English pattern to Braille Conversion

Besides giving the inputs using random numbers from 1 to 30 or 50, there is another form of giving the inputs, i.e. directly giving the unknown Braille cells as input together in form of a sentence which verifies the data base efficiency for practical implementation on unknown data.

4.5. Simulation Result of Braille to Odia /Hindi /Telugu /English Sentence

The input taken here is an unknown Braille sample images which undergoes the algorithmic flow discussed above to generate the corresponding output which gives a closest approach to the original meaning of the document.

When a particular language is given as input in workspace, then the respective image is taken as input. For Odia language, the input image is shown in Fig. 4.14.

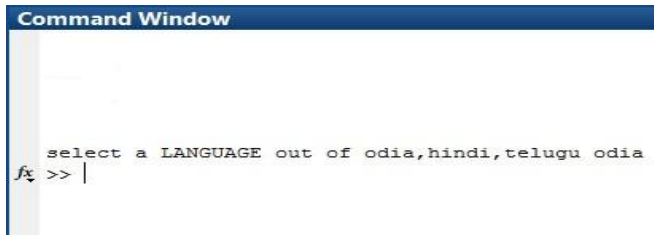


Fig -4.14: Selecting Odia language as Input in workspace

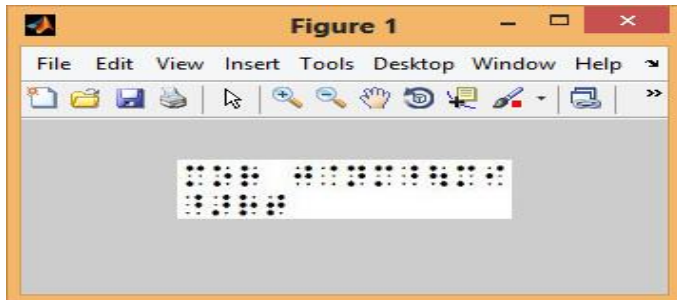


Fig -4.15: Reading the unknown Odia Braille image taken as input[25]

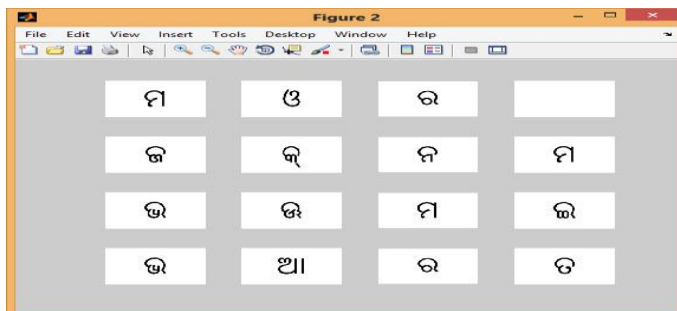


Fig -4.16: Interpretation of the Data Base for Odia Pattern as output

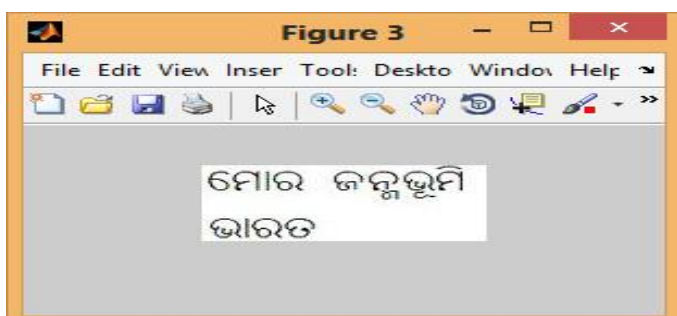


Fig -4.17: Original meaning of Figure 4.15[25]

When Hindi language is given as input in workspace(Fig. 4.18.), then the input image is shown in Fig. 4.19.

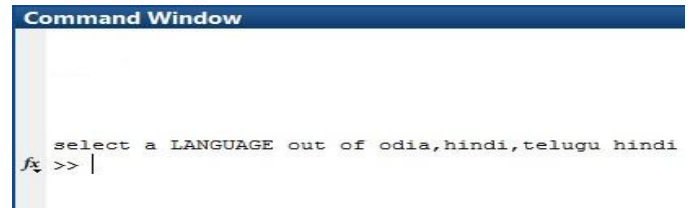


Fig -4.18: Selecting Hindi language as Input in workspace

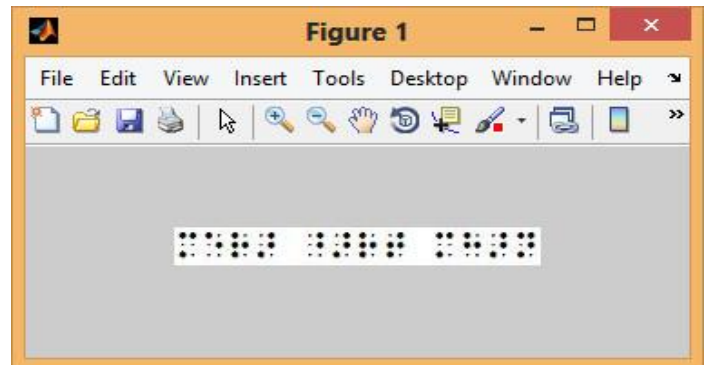


Fig -4.19: Reading the unknown Hindi Braille image taken as input[25]

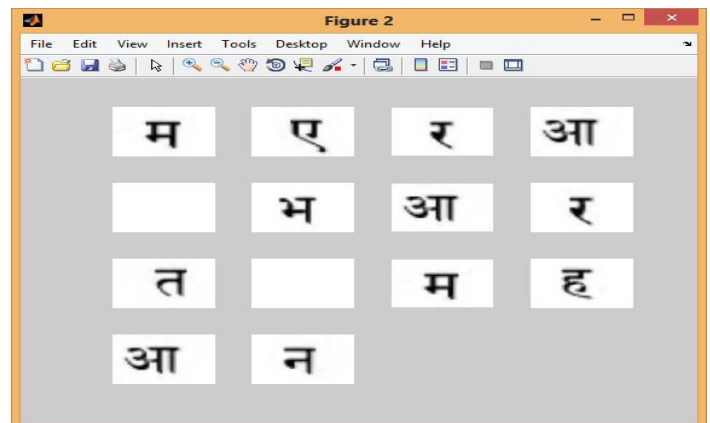


Fig -4.20: Interpretation of the Data Base for the Hindi Pattern as output

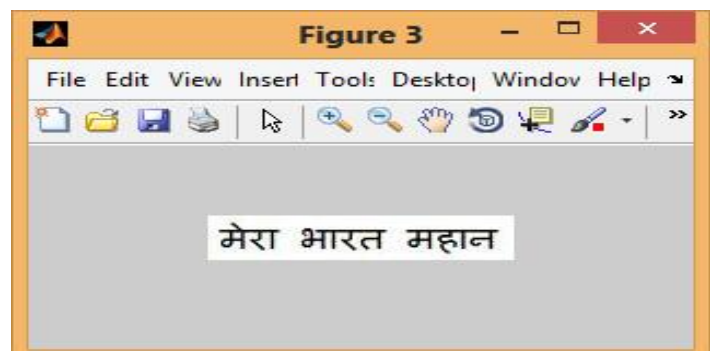


Fig -4.21: Original meaning of Figure 4.19[25]

When Telugu language is given as input in workspace(Fig. 4.22.), then the input image is shown in Fig. 4.23.

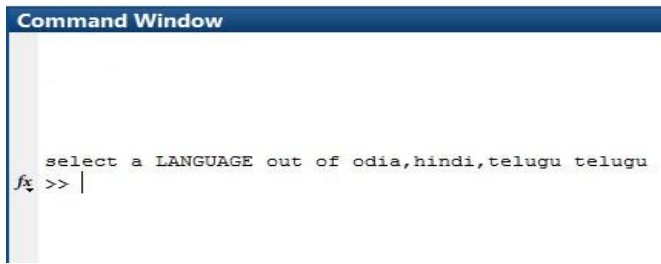


Fig -4.22: Selecting Telugu language as Input in workspace

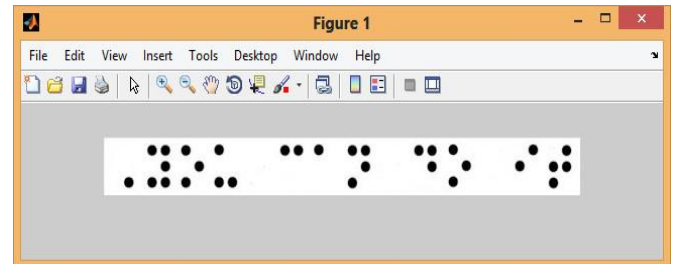


Fig -4.26: Reading the unknown English Braille taken as input[25]

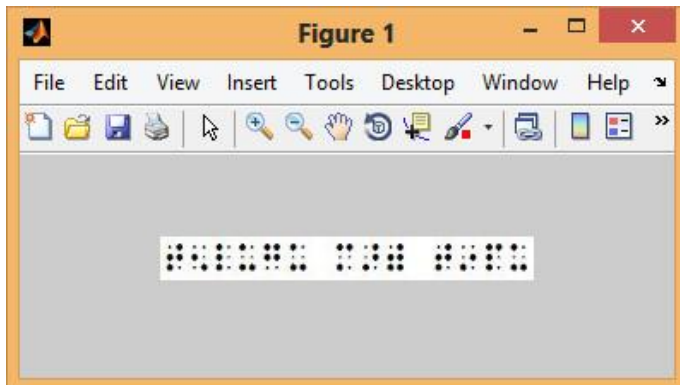


Fig -4.23: Reading the unknown Telugu Braille image taken as input[25]

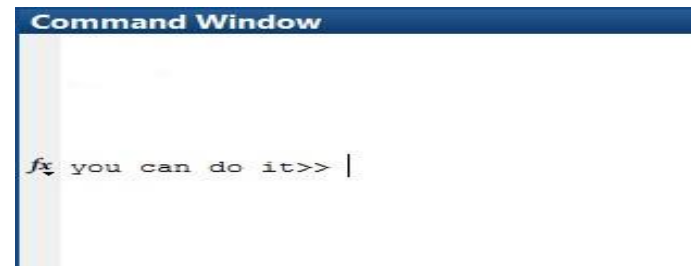


Fig -4.27: Interpretation of the Data Base for the English Pattern as output

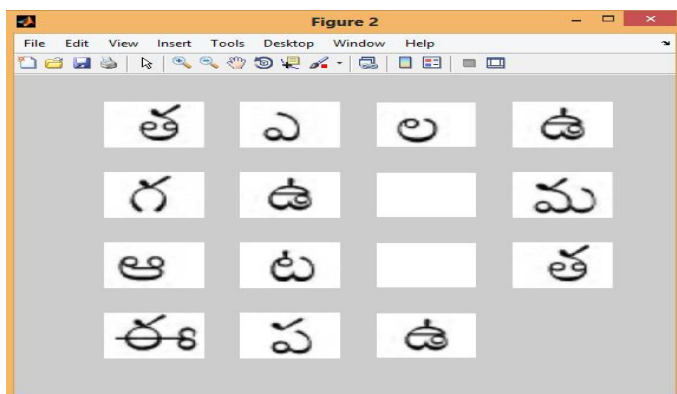


Fig -4.24: Interpretation of the Data Base for the Telugu Pattern as output

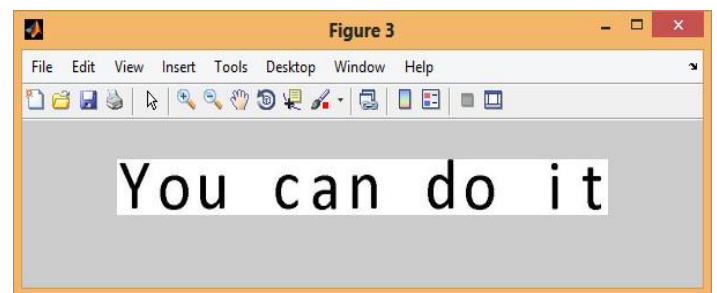


Fig -4.28: Original meaning of Figure 4.26[25]

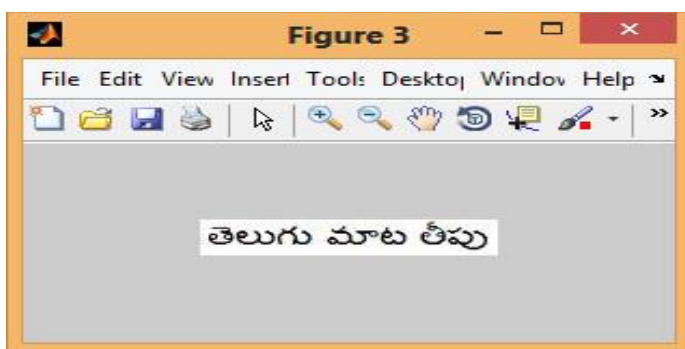


Fig -4.25: Original meaning of Figure 4.23[25]

As the data base for English is quite modified, so using that, the input is directly taken as the unknown Braille image displayed in MATLAB figure window(Fig. 4.26) while its interpretation as output is displayed in the workspace(Fig. 4.27).

The figures shown above for interpretation of the unknown images basically wants to convey the fact that the Interpreted images by the Developed Data Base is the real approach for the visually challenged people but is the closest approach for the normal mass as because when a vowel is followed by a consonant, then the consonants changes its shape in the native languages other than English, but, the visually challenged people go for reading the code cell wise for which they need to interpret the combinations of the letters through certain patterns.

5. HARDWARE IMPLEMENTATION

The developed data base is tested in hardware by dumping it into Spartan3e FPGA where the dot patterns of the corresponding patterns are being obtained using case structure. The following flow chart(Fig. 5.1) describes the same.

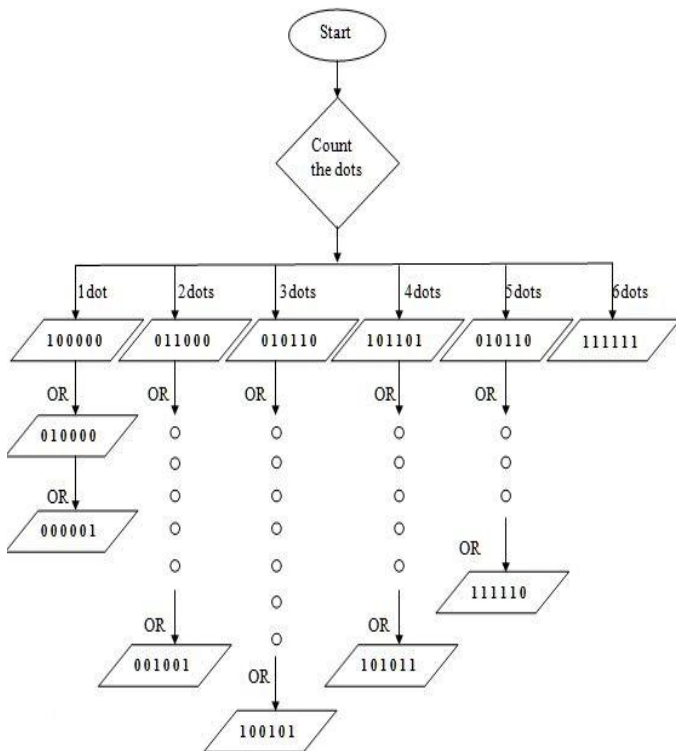


Fig -5.1: Flow chart of case structure for FPGA

5.1. Delay Calculation

The delay is the processing time taken by the system to convert the input into its respective output. Here, the data base after being dumped into FPGA generates the each dot pattern after certain delay, say Δt (where sampling frequency is 8kHz). The case statement having a delay of Δt adds up with the delay of each pattern recognition to generate the total delay of the output. Thus, the dot patterns after being recognized are stored in an address location in the ROM table (Table 6, created by the user) using which the data is retrieved. After getting the dot pattern, the corresponding letters are matched with the patterns.

Table -6: Dot Pattern Storage Table (ROM Table)

Number of dots	Dot pattern	ROM address location	Maximum delay
1	1 0 0 0 0 0	0	4 Δt
	0 1 0 0 0 0	1	
	0 0 0 0 0 1	2	
2	0 1 1 0 0 0	3	13 Δt
	0 0 0 1 1 0	4	
	1 0 0 1 0 0	5	
	0 1 0 0 1 0	6	
	1 0 0 0 1 0	7	
	0 1 0 0 0 1	8	
	1 1 0 0 0 0	9	
	1 0 0 0 0 1	10	
	0 0 1 1 0 0	11	
	1 0 1 0 0 0	12	
	0 1 0 1 0 0	13	
	0 0 1 0 0 1	14	
	0 1 0 1 1 0	15	

3	1 0 0 0 1 1	16	18 Δt
	1 0 0 1 1 0	17	
	0 1 1 0 0 1	18	
	1 0 1 0 0 1	19	
	0 1 0 0 1 1	20	
	0 1 1 1 0 0	21	
	0 0 0 1 1 1	22	
	1 1 0 1 0 0	23	
	0 0 1 1 1 0	24	
	1 1 0 0 1 0	25	
	1 0 1 0 1 0	26	
4	0 1 0 1 0 1	27	13 Δt
	1 1 0 0 0 1	28	
	0 1 1 0 1 0	29	
	1 0 1 1 0 0	30	
	1 0 0 1 0 1	31	
	1 0 1 1 0 1	32	
	1 1 1 1 0 0	33	
	0 1 1 1 0 1	34	
	1 1 1 0 0 1	35	
	0 1 0 1 1 1	36	
	0 1 1 1 1 0	37	
5	1 1 0 1 0 1	38	6 Δt
	0 1 1 0 1 1	39	
	1 1 0 1 1 0	40	
	1 1 1 0 1 0	41	
	1 0 1 1 1 0	42	
	1 0 1 0 1 1	43	
6	0 1 1 1 1 1	44	2 Δt
	1 1 1 1 0 1	45	
	1 1 0 1 1 1	46	
	1 1 1 0 1 1	47	
	1 1 1 1 1 0	48	
	1 1 1 1 1 1	49	

5.2. Sample Testing

The unknown Braille sample image is taken for testing whose dot pattern is being recognized via MATLAB and is dumped into FPGA. The dot pattern found is then verified with the dot pattern stored in the ROM table and if matching occurs, then the corresponding images are displayed in the monitor. Taking a sample of unknown Braille cell with count=3 and dot pattern= '1 0 1 1 0 0' generated the desired output of matching after a total delay of 17 Δt.

6. CONCLUSION AND FUTURE SCOPE

The proposed algorithm successfully resulted in the generation and verification of the Developed Braille Data Base using MATLAB. Also, it has elaborated the conversion of Braille to any native language (by some appropriate mapping of other languages) and vice versa within a fraction of few seconds for each letter. Another factor which makes the work more impactful is that the developed data base with small modifications gave the conversion of Braille to English and vice versa. Thus, the sample testing gave the desired direction (braille to text and vice versa) for the proper functioning of the designed Database. The data base is also tested for hardware section

by dumping into FPGA which defines the dot patterns for the letters and the delay occurrences for the output. The basic advantages of the paper are the availability and compatibility of the data base that can be implemented for any native language in modern gadgets, plenty of unknown Braille images can be stored and recognized in a parallel conversion in any language, the Data base created for English brings out the keyboard interfacing with the system programming accessibility, the flexibility of the data base which makes the possibility for two way conversions(braille to text and text to braille) which is not defined in any of the previous researches. The reverse conversions of the native language to braille creates a platform for the generation of Braille sheets from readily available texts starting from newspaper, novels to holy books. Also, the hardware section allows the MATLAB field to be dumped in FPGA that can be installed in mobile phones or other hand held devices via chips for two-way communication over text between the visually challenged and the mass which could be a futuristic approach. Moreover, this work can be carried out for a Grade 2 braille (English).

REFERENCES

- [1]. A.Malik et al.(2010). An Efficient Braille Cells Recognition. 6th International Conference on Wireless Communication Network and Mobile Computing (WiCOM),pp.223-223.
- [2]. Al-Salman et al.(2012). A Novel Approach for Braille Images Segmentation. *International Conference on Multimedia Computing and Systems (ICMCS)*,pp.190-195.
- [3]. C.Saravanan(2010). Color Image to Grayscale Image Conversion. Second International Conference on Computer Engineering and Applications (ICCEA), pp.96-199.
- [4]. Gerard Lachiver et al.(1984). An Optoelectronic Device to Read and Spell Braille Braillect. *IEEE Transactions on Biomedical Engineering*, Vol. BME-31, NO. 8.
- [5]. J.Yin et al. (2010). The Research on Paper Mediated Braille Automatic Recognition Method. *Fifth International Conference on Frontier of Computer Science and Technology*,pp.619-624.
- [6]. Jie Li et al.(2010). Optical Braille Recognition with Haar Wavelet Features and Support-VectorMachine. *International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*.
- [7]. Jyotsnarani Tripathy(2010). Reconstruction of Oriya Alphabets Using Zernike Moments. *International Journal of Computer Applications (0975 – 8887) Volume 8– No.8*.
- [8]. K.Parvathi et al.(2015). Odia Braille:Text Transcription via Image Processing. *International Conference On Futuristic Trends in Computational Analysis and Knowledge Management, IEEE*.
- [9]. Kazuyuki Ito et al.(2007). Development of Pen-based Note-Taking System for Blind People. *Second International Conference on Innovative Computing, Information and Control, ICICIC*.
- [10]. M.Wajid et al.(2011). Imprinted Braille Character Pattern Recognition using Image Processing techniques. *International Conference on Image Information Processing(ICIIP)*.
- [11]. Majid Yoosefi Babadi et al.(2009). Probabilistic Estimation of Braille Document Parameters. *16th IEEE Conference on Image Processing (ICIP)*.
- [12]. Munish Kumar et al.(2011). *k* -Nearest Neighbor Based Offline HandwrittenGurmukhi Character Recognition. *International Conference on Image Information Processing (ICIIP 2011)*.
- [13]. National Federation of the blind(2009). The Braille Literacy Crisis in America Facing the Truth, Reversing the Trend, Empowering the Blind. *A Report to the Nation*.
- [14]. P.P.Chitte et al.(2015). Braille to Text and Speech for Cecity Persons. *International Journal of Research in Engineering and Technology(IJRET)*, Volume:4, Issue:01.
- [15]. Prachi Rajarapollu et al.(2013). FPGA Based Braille to Text and Speech for Blind Persons. *International Journal of Scientific and Engineering Research*, Volume 4, Issue 4.
- [16]. Rafel C. Gonzalez,& Richard E.Woods(2008). *Digital Image Processing, third edition*. Pearson publisher.
- [17]. Ramesh Sankara Subbu,& Pawan Gnanaraj (2012). Enabling visually impaired to read messages from modern gadgets. *Sixth International Conference on Sensing Technology (ICST)*.
- [18]. S.D.Al-Shamma,& S.Fathi(2010). Arabic Braille Recognition and Transcription into Text and Voice. *5th Cairo International Biomedical Engineering Conference, Cairo, Egypt*,pp.16-18.
- [19]. S.Padmavathi et al.(2013). Conversion of Braille to text in English, Hindi and Tamil Languages. *International Journal of Computer Science, Engineering and Applications(IJCSEA) Vol.3, No.3*.
- [20]. S.R.Rupanagudi et al.(2014). Novel Methodology for Kannada Braille to Speech Translation using Image Processing on FPGA. *International Conference on Advances in Electrical Engineering(ICAEE)*.
- [21]. S.Shastry et al.(2013). 'i'- A novel algorithm for Optical Character Recognition(OCR). *International Multi Conference on Automation, Computing, Communication, Control and Compressed Sensing (iMac4s)*, pp.389-383,22-23.
- [22]. Shanjun ZHANNING,& Kazuyoshi YOSHINO (2007). A Braille Recognition System by the Mobile Phone with Embedded Camera. *Second International Conference on Innovative Computing, Information and Control, ICICIC*.
- [23]. Tapas Kumar Patra et al.(2012). Text to Speech Conversion with Phonematic Concatenation. *International Journal of Electronics Communication and Computer Technology (IJECCCT) Volume 2 Issue 5*.
- [24]. Veena Bansal,& R. M. K. Sinha(2002). Segmentation of Touching and Fused Devanagari Characters. *Elsevier Pattern Recognition*, Vol 35, Issue 4, pp.875-893.
- [25]. [www.google.co.in/#q=hindi+braille+sample+images\(ba raha braille\).com](http://www.google.co.in/#q=hindi+braille+sample+images(ba+raha+braille).com).
- [26]. [www.oriya, hindi, telugu, english braille-Wikipedia, the free encyclopedia](http://www.oriya,hindi,telugu,english+braille-Wikipedia,the+free+encyclopedia).

BIOGRAPHIES

Bijet Maynoher Samal received the BTech(etc) from Trident Academy of Technology, Odisha. MTech from KIIT University, Odisha.



K.Parvathi received the BTech(ece) from Osmania university. MTech & PhD from Andhra University, AP. Research interests are signal & Image Processing, Computational Intelligence.



Jitendra Kumar Das received BE from Utkal University. Mtech & PhD from NIT Rourkela, Odisha. Research interests are Signal Processing, VLSI & Embedded systems.