International Journal of Engineering Research And Advanced Technology (IJERAT)
ISSN: 2454-6135 [Volume. 03 Issue.1, January– 2017]

BRAILLE CHARACTER RECOGNITION USING ASSOCIATIVE MEMORIES

Sarah Hikmat Khaled[1], Assist. Prof. Safana H. Abbas[2]

College of Education in Computer Science

AL Mustansiriya University

Baghdad, IRAQ

ABSTRACT

In this paper, Braille character recognition (BCR) module is proposed that translates a single side Braille documents into English text and voice signal to help blind peoples saving their documents and hear it whenever they need the module is consisting mainly of two stages: the preprocessing stage and the recognition stage. In the 1st stage different thresholds and mask sizes were tested, and then the preprocessed image will be used in the recognition stage. Modify Multi-Connect Architecture (MMCA) and Modify Bidirectional Associative Memory (MBAM) algorithms were used to get the English document. A comparison is made between the two algorithms to get the best results. The implemented (MMCA) algorithm achieved average accuracy for correct letters was 98.26%, average accuracy for correct words was 95.11% and average processing time around 11.5 sec per page, The implemented (MBAM) algorithm achieved average accuracy for correct letters was 91.87%, average accuracy for correct words was 51.26% and average processing time around 3.4 sec per page.

Key words : *Braille, Optical character recognition, Associative Memory.*

1.1 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. A Braille Cell is composed of 6 dots arranged in three rows where each row consists of 2 columns. 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[11], as shown in Figure1. The six positions of dots are coordinated to give no more 64 various braille characters. These punctuate are written using a specialized machine [2].Braille Character Recognition (BCR) is a technique to determine and recognize Braille document stored in an image, such as a jpeg, jpg, tiff or a gif image, and transform the text into a coded machine form such as text file [4].

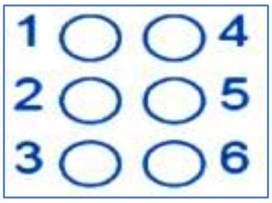
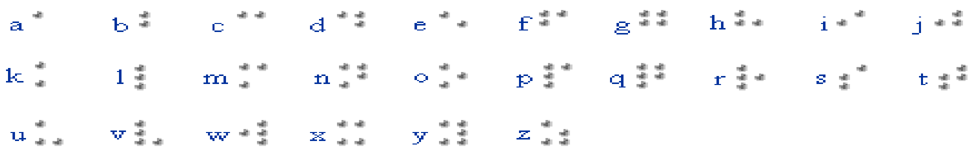


Figure 1.A Braille cell with 6 dots.

Each arrangement of dots is known as a cell and will consist of least one raised dot and a maximum of six, on a Braille sheet, as shown in Figure2.

Figure 2. Braille sheet.

The alphabet Braille is divided into two types grade1 Braille and grade2 Braille,Grade1 Braille is the most basic representations of letters, numbers and punctuation, [3, 4]. Grade 1 Braille is the most basic representations of letters, numbers and punctuation and is represented by the following dot combinations:

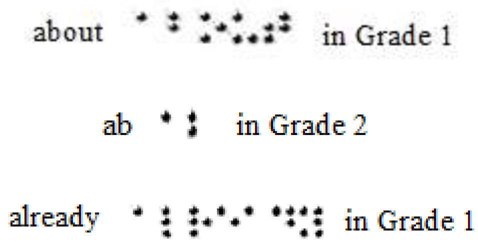


By itself a braille letter is assumed to be in lower case. To show an uppercase letter, the capital sign is put in front of the braille letter.

The number sign is used in the same way and put in front of the braille first letters of the alphabet.

Grade 2 Braille, now because Braille books are so much larger than print ones, numerous contractions have been introduced to make them take up less space and faster to read. Contracted Braille is known as Grade 2, and is by far the most widely used. It makes use of approximately more than 300 contractions (in addition to the representations mentioned above.) A contraction is used to shorten the length of a word. Contractions for example, the word about representing twoletters in Grade2. The word already represent only three letters in grade2 make it much faster and easier for everyone to enjoy a Braille book, While the Grade1 of the number

of letters in Braille word be equal to the number of letters of any language to the same word Here are some examples [3, 4, 5 and 6].



1.2 Related works

This section supply a survey of the literature related toBraille character recognition that were developed to get betterBraille character recognition.

S.Padmavathi, Manojna K.S.S, Sphoorthy Reddy .S and Meenakshy .D in 2013 [13] this paper suggests a process to transform a scanned Braille document to text which can be read out to many through the computer. The Braille documents are preprocessed to enhance the dots and reduce the noise. The Braille cells are segmented and the dots from each cell is extracted and transformed in to a number sequence.These are mapped to the appropriate alphabets of the language. The converted text is spoken out through a speech synthesizer. The paper as wellsupplies a mechanism to type the Braille characters through the number pad of the keyboard. The typed Braille character is mapped to the alphabet and spoken out. The Braille cell has a standard representation but the mapping varies for each language. In this paper mapping of English, Hindi and Tamil are assumed Accuracy of text conversion 97.8% -100%.

AbdulMalik Al-Salman, Yosef AlOhali, Mohammed AlKanhal, and Abdullah AlRajihin 2014 [1] the proposed system has used some new techniques to recognize Braille cells using a standard scanner. The system has been tested with a wide set of A4 scanned Braille documents, both single and double sided, written in the Arabic language andscanned with several scanners. Overall, on singlesided and double-sided documents 99% of the dots are rightly recognized.

Shreekanth.T, V.Udayashankara in 2013 [12] for highlight on the existing Optical Braille Recognition (OBR) solutions in this study with special emphasis on dot recognition of the Embossed Braille Image characters using matching algorithm to match each of the input decimal arrays corresponding to a word against the Braille word lookup table 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.

Aisha Musa, Hazem Hiary, Raja Alomari, and Loai Alnemer in 2013 [2] this system is very accurate, robust, and effective compared to the state-of-the-art systems, theysuggest a fully system to recognize characters for a single side braille document. This system improved each step begin from the image acquisition until the Braille cell recognition last stage. The recognition ability ranges between 94% and 99%.

Zainb. Authman, Zamen F. Jebr in 2013 [14] in this paper we have explained the development of an automatic system for recognizing printed Braille cells. It has been divided in different modules for each part of the image processing. For obtain this system, local and adaptive thresholding has been used, and shrinking mechanism is added to the system for make Braille cells shape's more regular to relent the next task of (OBR) system. This process has an activity and it take time only 17 sec. to recognize green Braille sheets and 21 sec. in recognize yellow Braille sheets.

1.3 Modify Multi-connect architecture associative memory [9, 10]

The MMCA associative memory comprise of a small architecture using the tinier size of the network, a steps that can be done by implementing the MMCA to a tiny number of nodes which makes the process of calculation for MMCA very fast and possible process in the real time. According to this principle, the network size is still fixed (but with two node) with a multiple connections between these two nodes. The number of connections between nodes consistent at two; this feature helps to deal with the smallest size of the network regardless of size of the patterns as explicated in Figure (3).

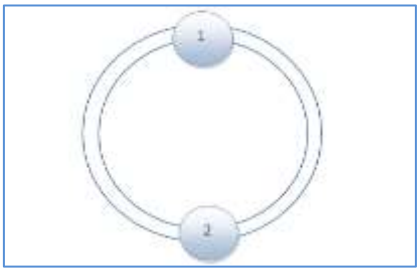


Figure 3.The architecture of MMCA associative memory.

Each neuron connected to the other neuron but not to itself as shown in Figure (3). These correlations between the neuron idealizes the corresponding weight for each vectors. The probable number of vectors is 2^2, which that need to the two only of the weights. This is because the bipolar representation for half of these vectors is orthogonal to the other half and these orthogonal vectors have the same weights.

1.4. Modify Bidirectional Associative Memory (MBAM) [7- 8]

The size of a Bidirectional associative memory (BAM) network (number of neurons in the network) is dependent on the pattern length utilized by the network (i.e., pattern with length ten requires a BAM network of size ten). Since MBAM is a modified BAM neural network, MBAM obeys the same size property. According to the first principle, the pattern will be segmented into a number of vectors with length two; this means that the size of the network will be fixed (i.e., two). Therefore, the network deals with parts of the pattern rather than of the entire pattern as one vector. This leads to the advantage of working with the smallest network size regardless of the pattern length, as well as multiple connections between the two neurons. **(7-8)**

The new architecture license the potential of avoiding learning the same vectors (which represents a certain part of the pattern) several times. This arrangement achieves the second principle described above. With a bipolar pattern representation, the elements will be either -1 or 1. The reason for choosing this length of vector is the shortest even length of any vector is two.

However, just as in the traditional BAM neural network, each node is connected to every other node but not to itself. These connections represent the corresponding weight of each vector in the pattern. Although the predicted number of vectors is, the number of connections will be just four, anfeature that helps deal with the teeniest network size regardless of the pattern length. Technically, as with traditional BAM nets, this adapted net has two phases (learning and convergence phases). In this research, these two processes will be modified.

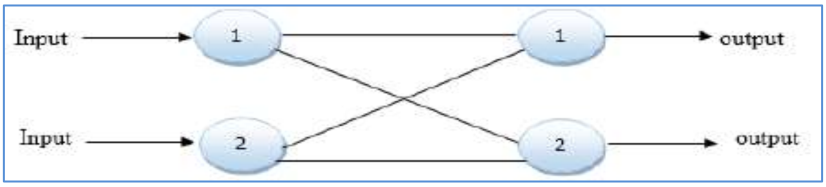


Figure (4): The Modified bidirectional associative memory (MBAM).

After presented the proposed new structure, it became important to provide algorithms that are approved this new structure for both learning and convergence phase.

Algorithm 1: of the learning phase for MBAM.

Input: training patterns p with code c.
Output: lookup table for all n corresponding stored patterns.
Step1: Repeat steps 1.1 and 1.2 to the end of training pattern p with code c:
Step 1.1: Divide the training pattern p to n vectors v with length two.
Step 1.2: For each vector v, repeat steps 1.2.1, 1.2.2 and 1.2.3:
Step 1.2.1: Assign the weight for each n vectors v, weights matrix sw as follows:

$$svw_i = v_i * c$$

Where: c is code
Step 1.2.2: Assign the stored vector's weight svw as follow:

$$svw = f(Dcode(v)) \begin{cases} 0 & \{means w0 \\ 1 & \{means w1 \\ 2 & \{means w2 \\ 3 & \{means w3 \end{cases}$$

Where: Decode is a function to convert the binary number to decimal number.
Step 1.2.3: Save svw for this vector in the lookup table.
Step 2: End.

Algorithm 2 (a): The convergence code algorithm for MBAM.

Input: n of unknown patterns p.
Output: Convergence code c.
Step 1: Repeat steps 1.1 and 1.2 until the unknown pattern p is ended:
Step 1.1: Divide the unknown pattern p to n vectors v with length two.
Step 1.2: Sum up the energy function for all n vectors v in the unknown pattern each with its corresponding vector in the stored patterns:

$$ep = -0.5 \sum_{i=1}^{n} v_i * svw_i * y_i$$

Where y_i is :

$$y_i = v_i * svw_i$$

Step 2: Determine the stored code number $minc$ with the minimum energy function to converge the unknown pattern towards it:

$$minc = \min(ep)$$

Where the min function is to determine the minimum energy function in ep array.
Step 3: End.

Algorithm 2 (b): The convergence pattern algorithm for MBAM.

Input: n of unknown codes c.
Output: Convergence pattern CP.

Step 1: Repeat steps 1.1 until the lookup table is ended:
Step 1.1: Find energy function ep:

$$ep = -0.5 * \sum x_i . c$$

Where x_i :

$$x_i = y_i * svw_i$$

Where y_i :

$$y_i = c * svw_i$$

Where c: is unknown code.
svw: is matrix weights stored in lookup table.

Step 2: Determine the stored pattern number $minp$ with the minimum energy function to converge the unknown code towards it:

$$minp = \min(ep)$$

Where the min function is to determine the minimum energy function in ep array.
Step 3: Build the final converge pattern cp:

$$cp = svw_{minp} * c$$

Step 4: End.

1.5The proposed BCR Module

The general flowchart of*BCR*module as shown in Figure 5.Where the captured by a scanner is then processed by the system and the assessment results are storing in the text file. The details of each phase discussed in next subsection.

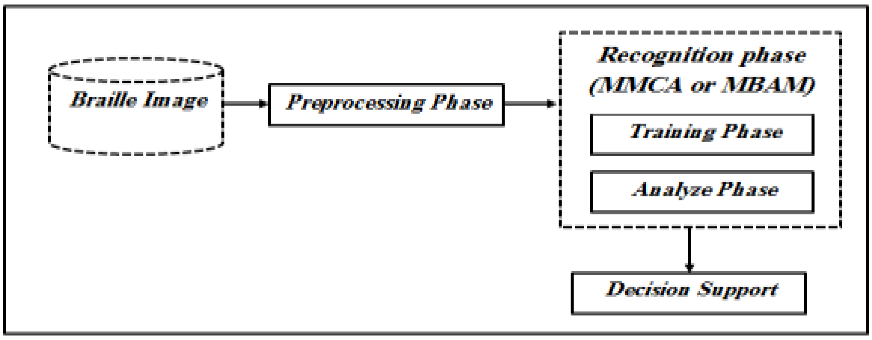


Figure 5.Flowchart of*BCR*module.

1.5.1 Image Acquisition

The role of scanner is just to scan the Braille sheet. Therefore,the Braille sheet scanned by using Brather Scan MFC-J470DW scanner with horizontal and vertical resolution 200 dpi, Image data are transferred from scanner to computer and stored in memory of the computer with JPG image format.

1.5.2 Preprocessing

The pre-processing phase consists in a set of operations that make the scanned image more suitable for the further phases:

1. The first operation performed to the image is the conversion to gray scale;
2. then the image converted into black and white format using the thresholding method
3. Next using dilation to expand the dots in braille image and to be clear, as shown in Figure 6.

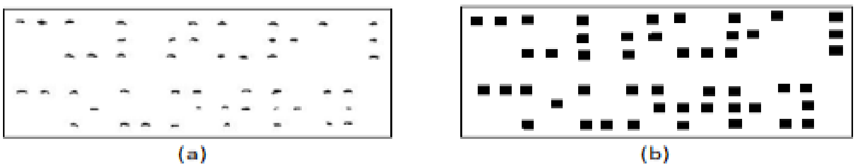


Figure 6. (a) complemented image, (b) An image after dilation

1.5.2.1 Character Area Allocation

In this step Braille sheet is projected horizontally and vertically to located area for each character. Each character comprises four positions (X_{up}, X_{low}, Y_{up}, Y_{low}) which are lower and upper bounds of the character as shown in Figure 7.

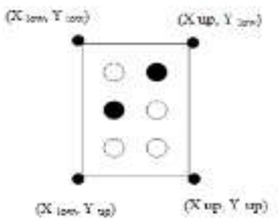


Figure 7: Charter zone represented by the quadruple (X_{up}, X_{low}, Y_{up}, Y_{low}).

1.5.3 Character Recognition using MMCA and MBAM

At this stage, the output of the preprocessing stage will be implemented to a recognition algorithm in order to be converted to a text file. Two algorithms are used, the MMCA and MBAM to recognize the image, which are used in two phases training phase and analysis phase as shown in next subsection.

1.5.3.1Training Phase

For MMCA, this phase will be applied on set of character as a training image during learn phase is implement for each character and save it in a lookup table in MMCA associative memory to be remembered during recognition process. For MBAM, this phase will be applied on set of character as a training image and set of code as code character, during learn phase is implement for each character and code then save it in a lookup table in MBAM associative memory to be remembered during recognition process.

1.5.3.2 Analyzing Phase

The analysis phase applied to analyze the BCR; this phase implemented for each braille image, to detect each character for it, the convergence phase of MMCA and MBAM using the lookup table that is built during the training Phase.Figure (8) illustrates the training phase and analysis phase steps to implement the MMCA and MBAM method respectively.

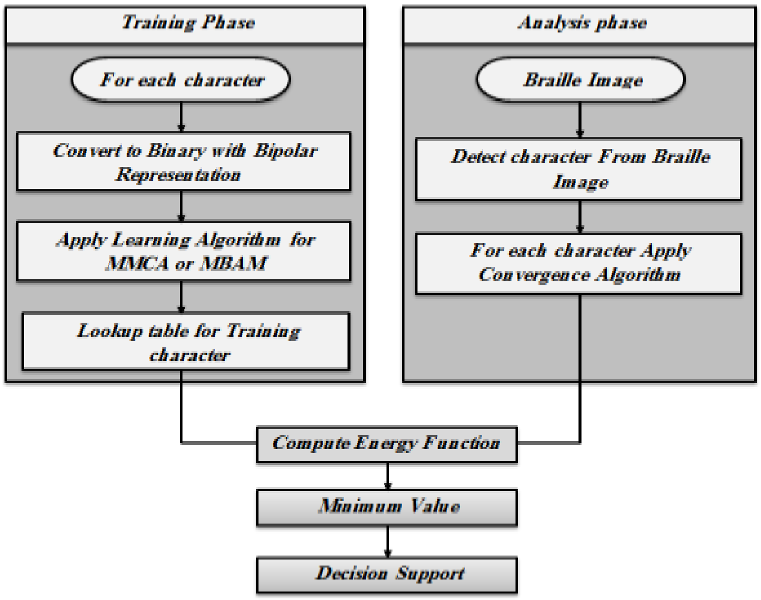


Figure 8: the training and recognition phase steps to implement of MMCA and MBAM method.

1.6 Results for MMCA and MCA Associative Memory

To evaluate the performance and stability of the proposed BCR module, the Braille image scanned with Brather Scan MFC-J470DW scanner. The Braille image transferred from scanner to computer and stored with JPG format. Image dataset contained 46 samples, each sample image have (35-114) words and 896 letters (the space expiration as letter) as shown in Table (1.4). The images have taken with dimensions (1700 * 2338 pixels) and resolution 200 dpi (dot per inch).

In this experiment, accuracy result and process time was measured using MMCA and MBAM. The dataset is used in this research composed from 46 Braille documents. The average accuracy for correct letters was

98.26% for MMCA and 91.87% for MBAM, as shown in Figure (9)..

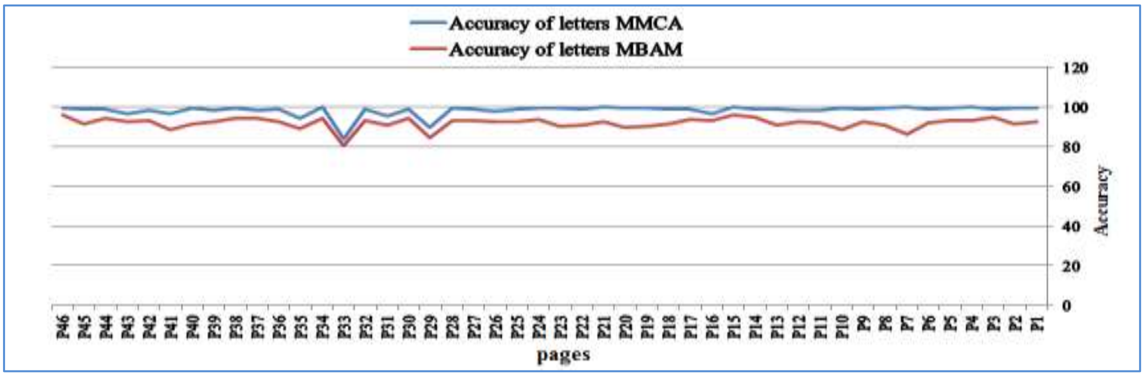


Figure (9): Accuracy of letters of each page by using MMCA and MBAM algorithms.

The average accuracy for correct words was 95.11% for MMCA and 51.26 for MBAM, the accuracy words for each Braille image show in Figure (10).

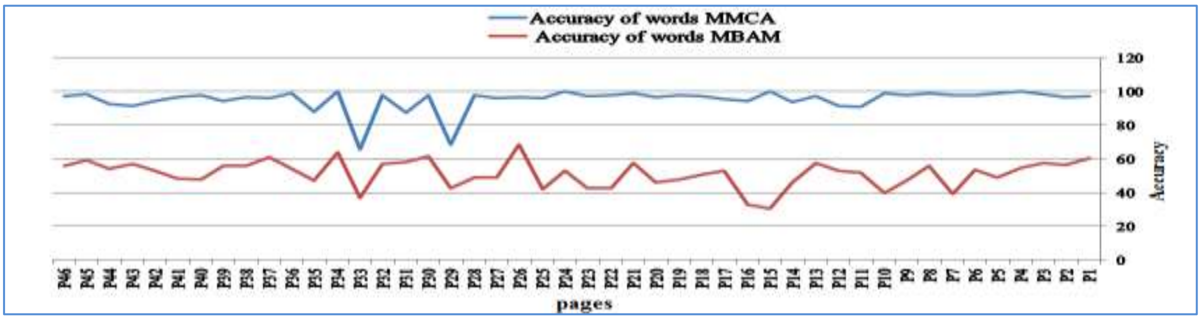


Figure (10): Accuracy of words of each page by using (MMCA) and (MBAM) algorithm.

The average processing time around 11.5 sec for MMCA and 3.4 sec for MBAM per page, the time process for each braille image illustrate in Figure (11).

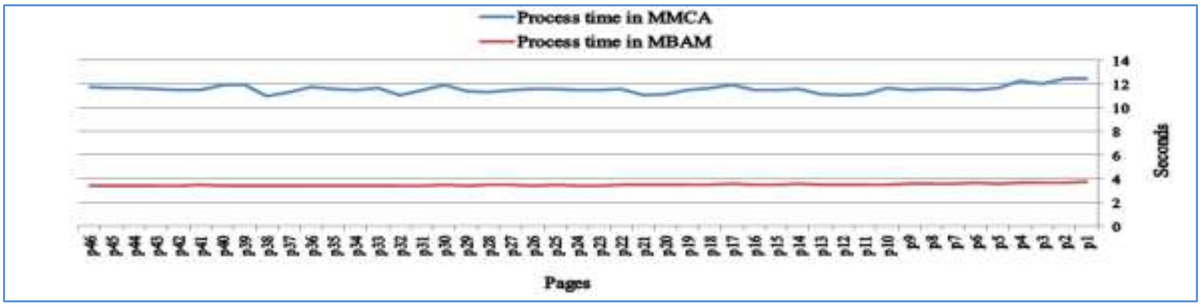
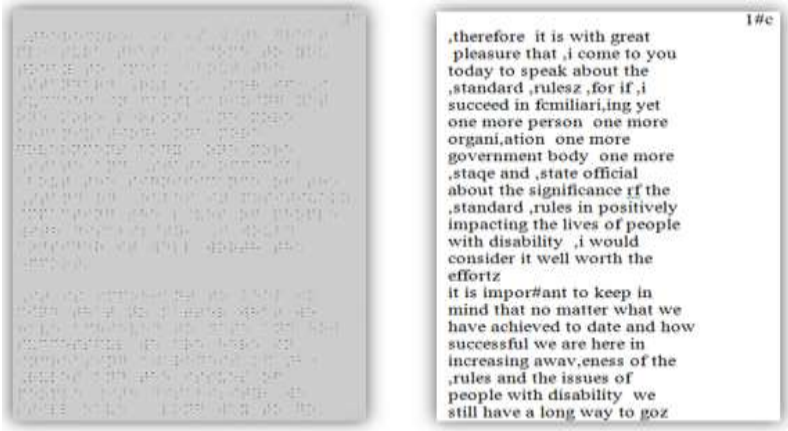


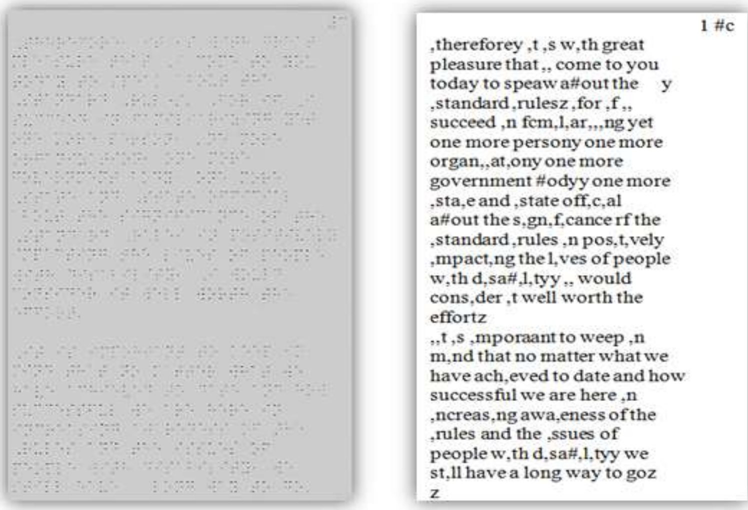
Figure (11): Process time of each page by using MMCA and MBAM algorithms.

The system is working for Grade 1 Braille English for purpose, in Figure (12) and (13) show some of the documents printed in Braille for the translation by the system, which translation from Braille to English text by using MMCA and MBAM algorithms respectively. Figures (12) and (13) show the original Braille image before and after using proposed BCR module. We can see in Figure (12) (b) that the translation was much clear and readable than Figure (13) (b).



(a) (b)

Figure (12): translation of Braille to English text by using (MMCA).
(a) Original Braille Image.(b) Translate using MMCA.



(a) (b)

Figure (13): translation of Braille to English text by using (MBAM).
(a) Original Braille Image. (b)Translate using MBAM.

The total number of words was (3940) and letters were (41216). The tested depend on Grade 1 Braille English. The experiments show that there are just 213 words and 813 letters for MMCA and 1992 words and 3347 letters for MBAM were unrecognized from the total number of words and letters (i.e. the 3940 words and 41216 letters respectively). Accordingly, the average accuracy was 95.11% words and 98.18% letters for MMCA and the average accuracy was 51.26% words and 91.87% letters for MBAM. The comparison between MMCA and MBAM illustrated in Table (1) and Figure (14).

Table (1): shows the accuracy results for MMCA and MBAM algorithms.

The algorithm type	number of pages	average accuracy for correct letters	average accuracy for correct words	average processing time
MMCA	46	98.26%	95.11%	11.5 sec.
MBAM	46	91.87%	51.26%	3.4 sec.

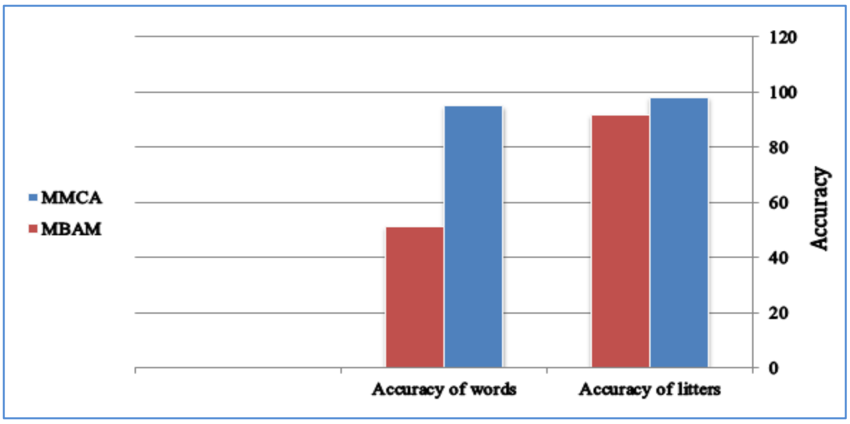


Figure (14): the accuracy results for MMCA and MBAM algorithms.

It is obvious from the above figure that the Accuracy of correct letters and words by using MMCA algorithm is higher than MBAM but MBAM has less time execution than MMCA.

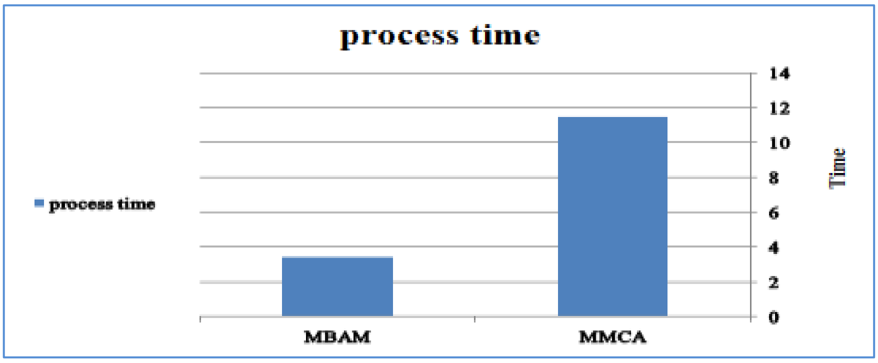


Figure (15): Process time by using MMCA and MBAM.

The reasons to why the accuracy is not 100% are some pale dots in the input images and the rotation of the character location caused by the scanner. From the above figure the process time for MBAM is faster than MMCA because the processing for MBAM is less than MMCA.

1.7Testing With Different Threshold Binary

In this experiment, using five braille images with different binary thresholds were applied with MMCA and MBAM associative memory. The results are shown in ***Table (2)*** and ***Table (3)***.

Table (2): The results With Different Thresholds using (MMCA).

By using (MMCA) Algorithm with different threshold								
Pages	***0.5***		***0.6***		***0.7***		***0.8***	
	L	***W***	***L***	***W***	***L***	***W***	***L***	***W***
P1	58.70%	1.83%	95.31%	87.89%	99.33%	97.24%	0%	0%
P2	55.80%	5.26%	94.30%	75.43%	99.33%	96.49%	0%	0%
P3	76.00%	8.82%	96.09%	70.58%	98.88%	98.48%	0%	0%
P4	66.40%	6.09%	94.64%	71.95%	99.77%	100%	0%	0%
P5	61.16%	4.87%	95.53%	73.17%	99.66%	98.78%	0%	0%
Average	63.61	5.37	95.17	75.80	99.39	98.19	0	0%

Where:

P1, p2, p3, p4, p5: page1, page2, ….

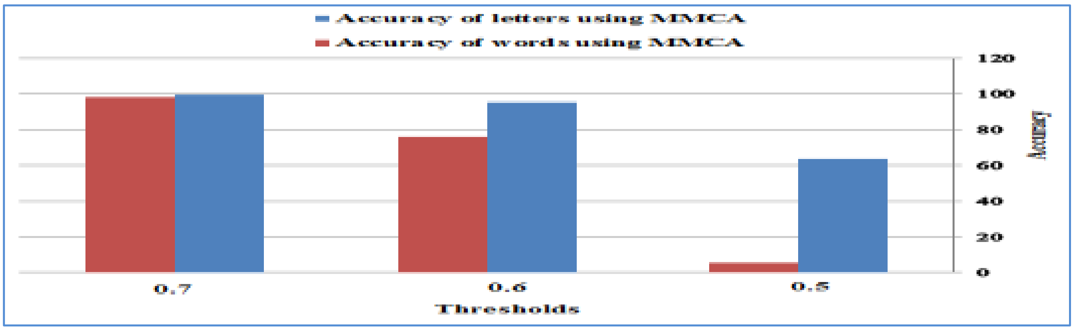
L: accuracy for correct letters.

W: accuracy for correct words.

Table (3): The results With Different Thresholds using (MBAM).

By using (MBAM) Algorithm with different threshold								
Pages	***0.5***		***0.6***		***0.7***		***0.8***	
	L	***W***	***L***	***W***	***L***	***W***	***L***	***W***
P1	54.46%	019. %	92.07%	51.37%	92.63%	60.55%	0%	0%
P2	50.55%	1.75%	86.83%	46.49%	91.62%	56.14%	0%	0%
P3	74.77%	1.47%	92.85%	36.76%	94.97%	57.35%	0%	0%
P4	62.72%	1.21%	88.28%	30.48%	92.85%	54.76%	0%	0%
P5	58.81%	3.65%	89.73%	36.58%	92.96%	48.78%	0%	0%
Average	60.26	1.65	89.95	40.33	93	55.51	0	0

The threshold with 0.7 gives higher accuracy than other thresholds by using MMCA and MBAM associative memory as shown in Table (2) and Table (3).



Figures (16): Accuracy of correct letters and words by using MMCA with different threshold.

In experiment above use different threshold and made compression, the results with 0.7 thresholds as shown in Table (2) and Table (3) and Figure (16) give higher accuracy than other thresholds as shown in Figure (16) by using MMCA associative memory.

1.7.1 Testing with Different Mask size for dilation and without dilation for five pages:

In this experiment, using five braille images with Different Mask size for dilation, which applied with MMCA and MBAM associative memory. The results are shown in Table (4) and Table (5).

Table (4): The result With Different Mask size for dilation with (MMCA).

By using (MMCA) Algorithm						
Pages	**Mask 3*3**		**Mask 5*5**		**Mask 7*7**	
	L	W	L	W	L	W
P1	99.33%	97.24%	93.19%	66.97%	78.79%	22.93%
P2	99.33%	96.49%	96.87%	80.70%	88.83%	54.38%
P3	98.88%	98.48%	97.32%	85.29%	90.29%	41.17%
P4	99.77%	100%	97.54%	80.48%	86.27%	40.24%

P5	99.66%	98.78%	97.09%	80.48%	88.50%	40.24%
Average	99.39	98.19	96.40	78.78	86.53	39.79

Table (5): The result With Different Mask size for dilation with (MBAM).

By using (MBAM) algorithm						
Pages	**Mask 3*3**		**Mask 5*5**		**Mask 7*7**	
	L	W	L	W	L	W
P1	92.63%	60.55%	86.94%	40.36%	72.43%	11%
P2	91.62%	56.14%	89.06%	48.24%	82.92%	32.45%
P3	94.97%	57.35%	93.41%	54.41%	86.83%	20.58%
P4	92.85%	54.76%	90.84%	47.56%	82.14%	19.51%
P5	92.96%	48.78%	90.95%	46.34%	83.70%	21.95%
Average	99.39	55.51	90.24	47.38	81.60	21.09

The results with Mask 3*3 size for dilation give higher accuracy than with other Mask sizesby using MMCA and MBAM associative memory as shown in Table (4) and Table (5).

Finally, the proposed module work is convert braille document to English language text and audio by using MMCA and MBAM associative memory networks. The braille image in this module does not use filters because these networks accepting high rate of noise and give high results.

1.8 CONCLUSION

Results Discussion and Analysis show that the proposed BCR average accuracy (MMCA) was 98.26%, for letter and 95.11% for word and average accuracy (MBAM) was 91.87% for letter and 51.26% for word. Thus, the results obtained from using (MMCA) algorithmbetter than the results obtained from using (MBAM) algorithm. This work focused on the development of an BCR for recognition single side English document via using a new techniques (i.e. associative memory with modify multi-connect architecture and Modify Bidirectional Associative Memory) paving the way to the future works to develop more efficient BCR in speed and accuracy using associative memory (may be after modified it). This work includes two stages preprocessing stage and the recognition stage the output of the recognition stage is converted into an audio signal. The input forms used in the experiments were Braille sheets. The scanned copies are then used as input to the proposed BCR.

REFERENCES

1. AbdulMalik Al-Salman, Yosef AlOhali, Mohammed AlKanhal, and Abdullah AlRajih,"*An Arabic*

Optical Braille Recognition System", ICTA'07, April 12-14, Hammamet, Tunisia.

2. Aisha Mousa, Hazem Hiary, Raja Alomari, and Loai Alnemer,"*Smart Braille System Recognizer"* ,IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 6, No 1, November 2013, ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784**.**
3. Gordon E. Legge, Cindee Madison and J.Stephen Mansfield, *"Measuring Braille Reading Speed With The Mnread Test"*, University of Minnesota, 75 East River Road, Visual Impairment Research, 2000 (in press).
4. Hind Mwafaq Taha, *"Robust Braille Recognition System Using Image Processing And Feature Extraction Algorithms"*, University Tun Hussien Onn Malaysia, January, 2014.
5. Manzeet Singh and Parteek Bhatia, *"Enabling The Disabled With Translation of Source Text To Braille"*,Thapar University, Patlala-147004,July – 2010**.**
6. Mohammed Y. Hassan and Ahmed G. Mohammed, "Conversion of English Characters into Braille Using Neural network", IJCCCE, Vol. 11 No.2, 2011.
7. Nisreen Abd Alhadi Jabr, Emad I Abdul Kareem, *"Modify Bidirectional Associative Memory (MBAM)"*, International Journal of modern trends in Engineering and Research, ISSN (Online) 2349-9754, ISSN (print) : 2393-8161 ,2015.
8. Nisreen Abd Alhadi Jabr, Emad I Abdul Kareem,*"Novel Hetero-Associative Memory: A Modified Bidirectional Associative Memory"*, International Journal of Engineering Research and Advanced Technology (IJERAT), ISSN: 2454-6135 (Online), Volume. 02 Issue.02, 2016.
9. RusulHusseinHasan,EmadIAbdulKareem, *"MODIFYMULTI-CONNECT ARCHITECTURE(MMCA)ASSOCIATIVEMEMORY"*,IntegratedJournalofEngineeringResearchandTechnology,Vol2(4)ISSNNO.2348–6821, 2015.
10. RusulHusseinHasan,EmadIAbdulKareem, *"An Image Processing Oriented Optical Mark Reader Based on Modify Multi-Connect Architecture MMCA"*,International Journal of modern trends in Engineering and Research, ISSN (Online) 2349-9745, ISSN (print) : 2393-8161 ,2015.
11. Santanu Halder, Abul Hasnat, Amina Khatun, Debotosh Bhattacharjee, Mita Nasipuri, *"Development of a Bangla Character Recognition (BCR) System for Generation of Bengili Text from Braille Notation"*, International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-3, Issue-1, June 2013.
12. Shreekanth. T, V.Udayashankara, *"A Review on Software Algorithms for Optical Recognition of Embossed Braille Characters"*, International Journal of Computer Applications (0975-8887) Volume 81-No.3, November 2013.
13. S.Padmavathi, Manojna K.S.S, Sphoorthy Reddy .S and Meenakshy .D, *"Conversion of Braille to Text in English, Hindi and Tamil Languages"*, International Journal of Computer Science,

Engineering and Applications (IJCSEA) Vol.3, No.3, June 2013.

14. Zainab. Authman1, Zamen F. Jebr, *"Algorithm Prototype for Regular Feature Extraction of Arabic Braille Scripts"*, International Journal of Scientific Knowledge Computing and Information Technology, Vol. 2, No.1, ISSN 2305-1493, 2013.