International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 3662-3669

A fuzzy Classification and Recognition System for Arabic Braille Segmented Characters

Amer Al Nassiri
IT College, Ajman University, Fujairah Campus, Fujairah, UAE.

Shubair Abdulla
College of Education, Instructional and Learning Technology Department, Sultan Qaboos University, Muscat, Oman.

Abstract

Braille documents play undeniable crucial role in the daily lives of low vision and visually-impaired people in current information and Internet era. Braille documents and its characters need to be understandable by vision people as well to motivate bridging the communication gap between visually-impaired people and vision people in different professions and work areas. Optical Braille recognition systems aim at automatically converting printed Braille characters into natural language characters. Arabic Braille character recognition systems have witnessed some development in recent years. However, there is a demand for more research contributions in this research area specifically in the character feature extraction and character recognition. This paper introduces a new fuzzy classification system for Arabic Braille characters. The system specifically tries to overcome the difficulties of Arabic Braille character feature extraction and recognition. The system has been evaluated using segmented Arabic Braille characters and the results have shown that the suggested system could achieve very good accuracy level.

Keywords: Arabic Braille, Feature Extraction, Fuzzy Classification, FCM, and kNN

INTRODUCTION

In the current Internet and communication society, the vision problem of many people across the world has been an obstacle to access huge amount of information. Globally, an estimated 253 million people who are visually-impaired and 30% of them are more likely to be unemployed according to the World Health Organization (WHO). Statistical surveys indicate that there are 35 million in the Arab countries who are blind or have low vision [1]. The visually-impaired people have active part and play a significant role in the society of Arab countries. So far, there are big efforts needed for those people to get them more involved in the current big data and information world. As a part of these efforts, different systems are created to reach what the word has in printed documents [2]. One of the most useful and valuable systems is Braille read and write system. Braille is a tactile writing system that enables visually-impaired people to read and write [3]. Braille documents are created by inscribing characters on the paper using a pattern raised dots. Therefore, it allows the visually-impaired people to read and write by the touch rather than the vision. Since its inception by Louis Braille 1829, it has become a famous way for low vision and visually-impaired people to participate in a literate culture [4].

A written form of information is undeniably crucial in human daily lives. For example, people perform exchange information, learn sciences, and deal with lows at their work using a written from of information. Braille system is the most widely adopted convention among visually-impaired people. Therefore, there have been massive Braille documents produced at different parts of the Arab world. Unless there is a smooth system for information flow between sighted people and visually-impaired people, a wide generation gap would be created between these two groups of people and the work of visually-impaired people would have remained buried. Although it is easy to produce Braille documents, there are difficulties to convert Braille documents into computer-readable forms. Optical Character Recognition (OCR) is the recognition of scanned documents that contain printed or written text characters. An extension to OCR, optical Braille recognition (OBR) offers conversion of Braille character images to natural language characters.

The development of OBR for Arabic Braille characters has been started in recent years. However, OBR for Arabic Braille is still an open research area that necessitates contribution of many research works. Two main problems facing any OBR for Arabic Braille character recognition, feature extraction and recognition [5]. Regarding to the feature extraction, there are some critical issues to be considered. The feature extraction should be resilient to image rotation and scaling, and also robust to noises. For the recognition issue, selection proper classifier that best fits the feature extracted has great potential to produce an accurate Arabic Braille recognition system. This paper proposes a new fuzzy classification and recognition system for Arabic Braille characters. To match the requirements of statistical nature of the fuzzy classifier, a new numerical features extraction algorithm is created as well. Hence, it could be say that this paper contributes in two fields of the Arabic Braille characters research area, feature extraction and recognition. This paper is structured as follows: Section 2 presents some background information and reviews some related works; the research methodology is explained in section 3. Section 4 presents the experimental results. The discussion of the results is presented in Section 5. Finally, the paper is concluded in Section 6.

BACKGROUND AND LITERATURE REVIEW

Typically, Braille characters are prepared using a special standard Braille paper of size 11 × 11.5 inches. A variety of line lengths are used depending on the material to be showed. A standard Braille sheet contains 25 lines with 40 embossed cells in line. Each cell accommodates one character represented by

2 columns × 3 rows of embossed dots. The height a dot is approximately 0.5 mm and the vertical and horizontal spacing between dot centers is approximately 2.5 mm [6]. Figure 1 shows an example of Braille character.

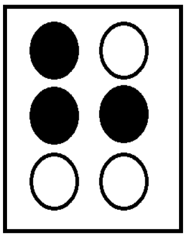

Figure 1. Braille character (2 columns × 3 rows of dots)

Arabic OBR is the process of converting Braille scanned images of machine printed or written text into a computer processable format. The basic steps in such systems can be generally divided into five stages [7]: Image Acquisition, Preprocessing, Segmentation, Feature extraction, and Classification. The image acquisition stage captures a Braille page and converts it into digital image files i.e. bmp or jpg. The preprocessing stage contains a collection of operations that apply some kinds of transformations on the image files to simplify the rest of stages. Examples of transformations include: distortion, noise reducing, and skew removing. The segmentation stage takes in an image and separates the different logical parts, i.e. Braille characters of a Braille word. The feature extraction stage, the most important stage, analyzes a Braille character and selects a set of features that can be used to uniquely identify the Braille character. The final stage, classification stage is a decision making stage of Arabic OBR system. It uses the features extracted to recognize the Braille character.

Different methods have been published to automate the transition of Arabic Braille characters into Arabic language characters and vice-versa. Abdul Malik et al. [6] attempted to develop algorithms to recognize image of Arabic Braille printed characters. The researchers located dots in cells using the soft shadows that are cast by Braille dots. Beside their method produced 99% classification accuracy rate, the need of a translation engine to convert the Braille cells to could be considered as an advantage in the research field. The system proposed in [8] performs a set of typical OBR system stages, document digitization, image preprocessing, image enhancement, image segmentation, image alignment and image recognition. The feature extraction depends entirely on computing the centroids of dots, while the recognition is done by grouping the dots based on location information relaying on the standard measurements of Braille documents. The system's recognition ability ranged between 94% and 99%.

Recent efforts involve translating Muslims' Holy Quran into Braille documents. Quran Braille translator system is introduced to translate Quran verses text into Braille code [9]. The system uses Extended Finite State Machine technique for detecting Quran reciting rules from the Quran text and Markov Algorithm (MA) for translating the detected rules and text into the matched Braille code. Similarly, Dayang Damit et. al. [10] attempted to mediate a new pathway to interlink and translates inputs from keyboard into the most used Braille character automatically. Their proposed system can be switched to Arabic Braille character to introduce them with Quran language. Its output was able to represent each alphabet, number and Arabic letter of the Braille symbol by mechanically raising and lowering pins to create the tactile dots of Braille using solenoid.

The attempts were not limited to translations and recognition of Arabic Braille characters, Zaghloul et al. [7] introduced a system of Arabic OBR which designed for recognizing a scanned Arabic Braille documents. The system performs two stages of conversion, scanned documents to computerized textual forms and computerized textual forms to sound files. They worked with a large database of Arabic Braille documents that include multi size and resolution digitized documents. The conversion stage depends on utilizing three stages of preprocessing, cell detection, interpretation. Similarly, Al-Shamma and Fathi [11] presented a system Arabic OBR with voice and text conversion. Their system was based on a comparison of Braille dot position extraction with database generated for each cell. It also involved a unique decimal code generation for each Braille cell which is used for word reconstruction with corresponding voice and text conversion database.

An overview of the research published in this field reveals that there is a need for sophisticated Arabic Braille recognition systems that satisfy basic conditions such as robust to image scanning noise, feature extraction efficiency, feature matching efficiency, and minimal test error classification.

RESEARCH METHODOLOGY

The process of recognizing Braille characters is naturally different from that of recognizing the printed characters. However, there are interesting parallels that can be drawn specifically the general stages of the process which pass through many stages starting from Braille image digitization stage to character recognition stage. This section is concerned with designing Braille image recognition and attempts to translate Arabic Braille characters to their equivalent Arabic characters. The overall architecture of Arabic Braille recognition system is shown in Figure 2.

Feature Extraction

Feature extraction process is a representational mechanism of Braille images [12]. It aims at extracting features of Braille characters from the digitized image. In the sequence of an Arabic OBR stages, the stage of feature extraction comes after Braille image acquisition, preprocessing, and segmentation stages, which all aim to make the Braille image be acceptable for the feature extraction algorithm, i.e. removing noise from image and identifying the regions of Braille characters in the image. Extraction of good features plays a very influential role in Arabic Braille recognition system because it leads to correctly recognizing Braille code characters [13]. It must contain small sets of discriminating information that can efficiently distinguish between cells, and must be robust in

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 3662-3669

order to avoid generating same feature codes for cells in different classes.

This paper presents a new numerical features extraction algorithm to match the requirements of statistical nature of the classifier adopted in this paper. The algorithm is categorized as blob detection method [14]. It assumes that the 6 dots of Braille's defined cell are very influential in recognizing the character. In each cell, there are 6 dots, 3 rows × 2 columns. Figure 3 shows the Braille representation of Arabic characters. Two main tasks are performed in the preparation step. Firstly, the scanned Braille characters are converted to BMP monochrome images. Secondly, the BMP monochrome images are represented into two-dimensional array of binary numbers, 0 indicates white color whereas 1 represents black color in corresponding pixel position.

The algorithm tries to detect and isolate dots in each cell. Typically, the dots differ in color property, dark and light color in most Braille images. The detection and isolation processes are done by scanning in a successive manner the cell in four movements. As shown in the Braille sample depicted in Figure 4, the first movement, called start movement, is a diagonal movement that begins from the upper left cell (0,0) down to the first cell of the left column at the point (103,74).

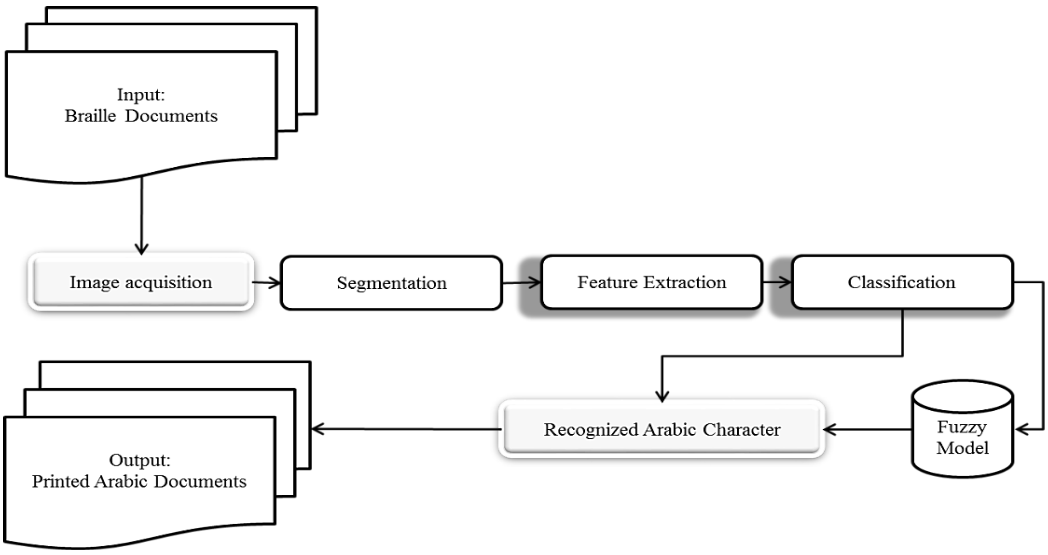

Figure 2 Architecture of Arabic Braille classification and recognition system

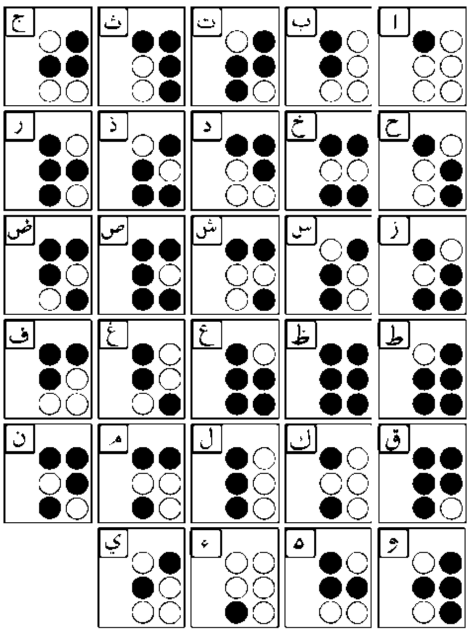

Figure 3 Braille codes for Arabic characters

The second movement is vertical straight down scan. It begins from the end point at (103,74) of the previous movement and seeks to detect and isolate the left column in the cell, it will end at point (103,292) as in the sample shown in Figure 4. The third movement starts from the end point of the first movement at (103,74). It scans the region to the right and stops at the first dot of the right column and stops at (329,74). The final movement resembles the second movement. It aims to detect and isolate the right column in the cell starting from (329,74) to (329,292).

Figure 4 shows the scanning movements. Upon finishing the scans, each column in the region is represented by a single binary number of three digits. The 1's and 0's reflect occupied (black) dots and empty (white) dots respectively. After compiling the 1's and 0's that total 6 digits, the algorithm converts them into octal number of two digits to represent character code. The Most significant digit in character octal code represents the left column, while the right column is represented by the least significant octal character code. Table 1 shows the codes (octal numbers) for the Arabic characters generated by the feature extraction algorithm.

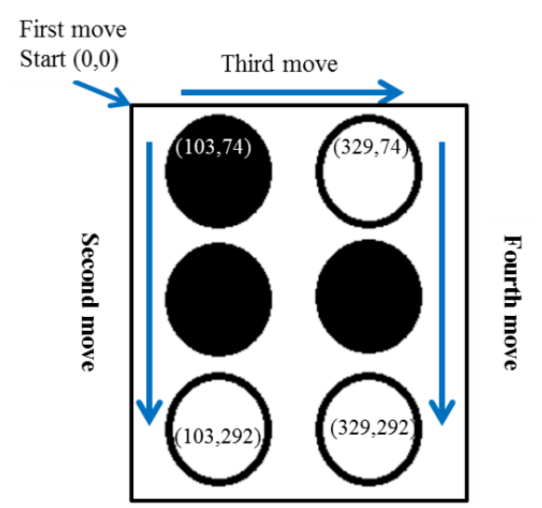

Figure 4 Scan movements of feature extraction algorithm

Classification

A new fuzzy classification algorithm is introduced in this section. The algorithm is inspired by the fuzzy c-mean (FCM) [15] and the fuzzy kNN [16] classification algorithms, which are among the most popular fuzzy clustering techniques. The new algorithm builds the classification model by clustering the training objects in a way somewhat similar to that in the FCM. However, instead of calculating the center of the cluster, it uses the training objects' attributes as cluster centers initially. The classification of the unseen objects is based on the Euclidean distance as in kNN. There is, however, a difference in the determination of the class. While the unseen object is classified in kNN by a majority vote of its neighbors in the new algorithm, it is simply classified using the nearest cluster center

Table 1 Arabic character codes

Char	Code	Char	Code	Char	Code
ا	10	ز	56	ق	73
ب	20	س	61	ك	50
ت	63	ش	15	ل	70
ث	17	ص	75	م	51
ج	23	ض	35	ن	53
ح	14	ط	67	و	27
خ	33	ظ	77	ه	32
د	13	ع	76	ي	60
ذ	65	غ	34	ء	21
ر	72	ف	31		

To build a classification model, the algorithm partitions set of objects $o=\{o_1,o_2,\ldots,o_n\}$ in R^d dimensional space into c ($c=n$) fuzzy clusters with $Z=\{Z_1,Z_2,\ldots,Z_n\}$ cluster centers. A fuzzy matrix μ with $n\times n$ dimensions is used to describe the fuzzy clustering of an object. The element μ_{ij} in the ith row and jth column in u indicates the degree of membership function of the ith object with the jth cluster:

$$\mu_{ij} \geq 0 \;\forall i = 1, 2 \ldots n;\; \forall j = 1, 2 \ldots n \qquad \ldots (1)$$

Initially, the centers for the clusters are equal to the objects:

$$o=\{o_1,o_2,\ldots,o_n\},\; Z=\{Z_1,Z_2,\ldots,Z_n\}, O=Z \qquad \ldots(2)$$

For a new unseen object o_x, the degree of membership function is defined as the Euclidean distance between an object and the center of its cluster:

$$\mu_{xj} = \sum_j^n d_{xj} \qquad \ldots (3)$$

Where

$$d_{xj} = \|o_x - Z_j\| \qquad \ldots (4)$$

The d_{xj} indicates the proximity of the cluster centers. The object o_x is simply assigned to the class of that nearest cluster center. The fuzziness nature of our algorithm is an added feature to the classifier that could solve the problems of noise and incorrectly scanned documents. Extraction of features of noisy documents, which occurs very often, might result in 11 for class "ا" instead of 10. The fuzzy sets theory allows us to classify it correctly as its membership value strongly tends to class "ا". In contrast by using FCM and kNN, our algorithm is straightforward, and easy to be used since no further calculations of cluster centers or objective function. Moreover, the classification is done by performing single step of finding the nearest cluster center.

SYSTEM EVALUATION

Dataset

It is noticed that the research community lacks public dataset for Arabic Braille images that could be used for evaluating newly produced recognition systems. The dataset used throughout the experiments is obtained by scanning single side embossed Arabic characters Braille documents using a flat-bed scanner with horizontal and vertical resolution 300 dpi bit depth 24. About 5 documents each of 112-125 cells have scanned and stored in JPEG format files. Two main tasks have been performed in preparing the dataset. Firstly, all the files have been converted from JPEG format to BMP monochrome format. Secondly, segmentation process is done to determine the cells. The segmentation is done manually since the main goal of this is research Arabic Braille character classification and recognition. The cell frame size is experimentally determined and recorded as 100×95 pixels. All the cells are saved in BMP monochrome format files. The segmentation task produced 578 cells, and we selected 500 cells to perform the experiments. Table 2 shows the classes, number of classes, and number of samples in the dataset. To prepare input for the feature extraction engine, the Braille cells BMP monochrome images are represented in two-dimensional array of binary numbers, 0 indicates white color and 1 represents black color in corresponding pixel position.

Results

The input to the feature extraction and classification stages are all the segmented Braille character cells. As the dataset do not have an exactly equal number of instances in each class which could be considered as imbalance, the following performance measures that can provide more insight into the accuracy of any multi-classification algorithm are calculated in the experiments:

- Specificity: A measure of a classifier's true negative rate.
- Sensitivity: A measure of a classifier's true positive rate.
- Accuracy: A classifier's accuracy rate.
- Error: A classifier's error rate.

We used class-based method to evaluate our multi-class algorithm. Initially, the four metrics are computed for each class as it done for a binary classification as shown in Table 3:

$$specificity = \frac{tn}{tn+fp} \quad \text{... (5)}$$

$$sensitivity = \frac{tp}{tp+fn} \quad \text{... (6)}$$

$$accuracy = \frac{tp+tn}{tp+fn+tn+fp} \quad \text{... (7)}$$

$$error = \frac{fp+fn}{tp+fn+tn+fp} \quad \text{... (8)}$$

After that, these class-wise metrics are aggregated to produce measurements for the algorithm. Eq. 9 has been used to obtain the class-based classification metrics [17]:

$$\bar{B} = \frac{1}{n}\sum_{j=1}^{n} B(tp_j, fp_j, tn_j, fn_j) \quad \text{... (9)}$$

Where

$\bar{B}$: any metric e.g. specificity for the algorithm,
B: any metric e.g. specificity for *j*th class, and
n: number of classes (29 classes)

For each class c_x, the *tp*, *fp*, *tn*, and *fn* variables are calculated as follows:

- *tp* of c_x**:** is all c_x instances that are classified as c_x.
- *fp* of c_x**:** is all non-c_x instances that are classified as c_x.
- *tn* of c_x**:** is all non-c_x instances that are not classified as c_x.
- *fn* of c_x**:** is all c_x instances that are not classified as c_x.

Regarding the algorithm-wise experimental results, Figure 4 summarizes the results by showing the averages of *sn*, *sp*, *acc*, and *err* metrics. In Figure 5, we computed separately the summations for *tp*, *tn*, *fp*, and *fn* metrics over all 29 classes. The high value for these metrics is justified to the calculation of summations.

Table 2. The dataset

Class	Samples	Class	Samples
ا	58	ط	14
ب	25	ظ	9
ت	14	ع	23
ث	5	غ	13
ج	13	ف	18
ح	20	ق	13
خ	7	ك	14
د	15	ل	42
ذ	8	م	22
ر	20	ن	15
ز	7	و	15
س	16	ه	28
ش	15	ي	18
ص	15	ء	8
ض	10		
Classes: 29		**Samples: 500**	

Table 3 Class-wise experimental results

S	Class	tp	tn	fp	fn	sn	sp	acc	err
1	ا	41	343	99	17	71%	78%	77%	23%
2	ب	20	398	77	5	80%	84%	84%	16%
3	ت	14	436	50	0	100%	90%	90%	10%
4	ث	4	396	99	1	80%	80%	80%	20%
5	ج	13	413	74	0	100%	85%	85%	15%
6	ح	18	373	107	2	90%	78%	78%	22%
7	خ	5	428	65	2	71%	87%	87%	13%
8	د	15	397	88	0	100%	82%	82%	18%
9	ذ	7	401	91	1	88%	82%	82%	18%
10	ر	19	396	84	1	95%	83%	83%	17%
11	ز	6	421	72	1	86%	85%	85%	15%
12	س	14	388	96	2	88%	80%	80%	20%
13	ش	13	402	83	2	87%	83%	83%	17%
14	ص	14	401	84	1	93%	83%	83%	17%
15	ض	10	410	80	0	100%	84%	84%	16%
16	ط	13	384	102	1	93%	79%	79%	21%
17	ظ	7	390	101	2	78%	79%	79%	21%
18	ع	20	400	77	3	87%	84%	84%	16%
19	غ	13	396	91	0	100%	81%	82%	18%
20	ف	15	397	85	3	83%	82%	82%	18%
21	ق	9	414	73	4	69%	85%	85%	15%
22	ك	12	403	83	2	86%	83%	83%	17%
23	ل	35	369	89	7	83%	81%	81%	19%
24	م	21	379	99	1	95%	79%	80%	20%
25	ن	14	407	78	1	93%	84%	84%	16%
26	و	13	414	71	2	87%	85%	85%	15%
27	ه	23	380	92	5	82%	81%	81%	19%
28	ي	14	398	84	4	78%	83%	82%	18%
29	ء	6	407	85	2	75%	83%	83%	17%

DISCUSSION

As each class has resulted in different values of *tn* and *fp*, it is rational to see the total values for these two metrics appeared in thousands. Obviously, there is a margin for developing the algorithm performance in terms of classification accuracy (*acc*=83%). However, the accuracy of the algorithm declines with increase in the noise level of Braille documents. Hence, it is better to incorporate advanced noise removal technique in any Braille character recognition system. This is one of the challenges facing the algorithm presented in this paper.

The variation of the Braille dot size greatly affects the scan movements of feature extraction engine. In some Braille cells rows and columns were scanned incorrectly. This is what the specificity metrics has revealed, for example classes of "ب", "ت", "ث", "ج" all have adjacent dots on a row and/or column which caused high rate of misclassification (*sp*=90% for "ت"). The higher values of *err*>=19% have been calculated for classes "ا", "ط", "ظ" and "و" for example. Each class of them either has three empty dots or has three painted dots in the second column. This reflects the need for some improvement in performing scanning for the second column, the fourth scanning move in the feature extraction algorithm.

Another observation is the sensitivity which measures the proportion of cells that are correctly classified (*tp*) out of all class cells (*tp*+*fn*). The aggregated value is 86% and in classes "د" and "ض" the rate hits 100%. These values reflect excellent performance in some cases. The standard deviation is calculated as in Figure 6 to tell how measurements *sn*, *sp*, *acc*, and *err* are spread out from the mean. The values of standard deviation show a significant approximation of the mean.

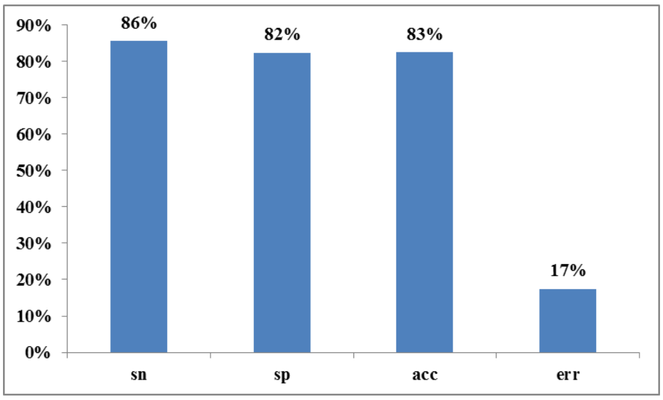

Figure 4 Algorithm-wise experimental results (Percentages of *sn*, *sp*, *acc*, and *err*)

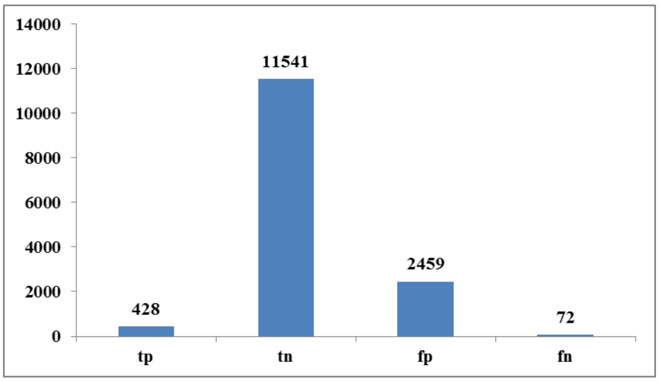

Figure 5 Algorithm-wise experimental results (Totals of *tp*, *tn*, *fp*, and *fn*)

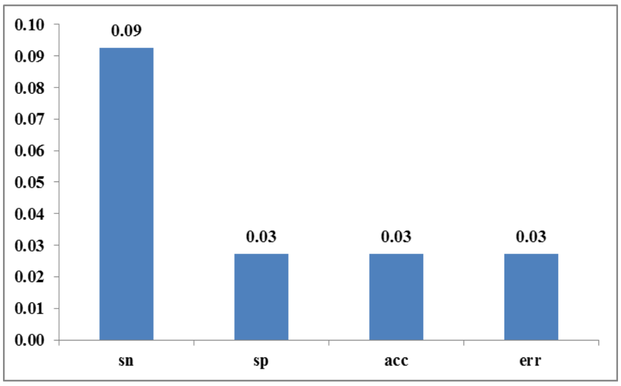

Figure 6 Algorithm-wise experimental results (Standard deviation of *sn*, *sp*, *acc*, and *err* percentages)

CONCLUSION

The Braille documents are tactile documents widely used by visually-impaired people to help them read and write. Development of automatic characters Braille recognition system can fill the gap of communication between vision people and visually impaired. A lot of research effort has been made to fill this gap. In this research an attempt has been made on feature extraction engine and fuzzy classification algorithm of Arabic Braille character recognition. After collecting and scanning 5 Arabic Braille documents, they are preprocessed and segmented into regions which represent an Arabic Braille character. The feature extraction engine presented in this paper is used to detect and isolate cells in each single region. It scans a cell in four scanning directions in order to produce a code of octal number for the character. The fuzzy classification algorithm is used to classify characters based on their octal number codes generated. It creates clusters based on the centers of the training points. For unseen objects, it computes the fuzzy membership degree of them to all of the clusters created during the training, and then simply assigns them to the class of that nearest cluster center. The performance of feature extraction engine and the fuzzy classification algorithm has been evaluated in terms of accuracy. The results have confirmed a promising attempt to automatically recognizing Arabic Braille characters. The majority of the misclassification errors could be attributed to the challenges faced by the noise removal and the variation of the Braille dot size. All of these challenges open a door for future research directions.

REFERENCES

[1] R. R. Bourne, S. R. Flaxman, T. Braithwaite, M. V. Cicinelli, A. Das, J. B. Jonas, *et al.*, "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis," *The Lancet Global Health,* vol. 5, pp. e888-e897, 2017.

[2] S. A. Hassen and Y. Assabie, "Recognition of Double Sided Amharic Braille Documents," *International Journal of Image, Graphics and Signal Processing,* vol. 9, p. 1, 2017.

[3] J. Siqueira, F. A. A. de Melo Nunes, D. J. Ferreira, C. R. G. Silva, L. de Oliveira Berretta, C. B. R. Ferreira, *et al.*, "Braille Text Entry on Smartphones: A Systematic Review of the Literature," in *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40th Annual*, 2016, pp. 521-526.

[4] M. Zakariah, M. K. Khan, O. Tayan, and K. Salah, "Digital Quran Computing: Review, Classification, and Trend Analysis," *Arabian Journal for Science and Engineering,* pp. 1-26, 2017.

[5] S. Isayed and R. Tahboub, "A review of optical Braille recognition," in *Web Applications and Networking (WSWAN), 2015 2nd World Symposium on*, 2015, pp. 1-6.

[6] A. Al-Salman, Y. AlOhali, M. AlKanhal, and A. AlRajih, "An arabic optical braille recognition system," 2007.

[7] R. I. Zaghloul and T. J. Bani-Ata, "Braille Recognition System–With a Case Study Arabic Braille Documents," *European Journal of Scientific Research,* vol. 62, pp. 116-122, 2011.

[8] A. Mousa, H. Hiary, R. Alomari, and L. Alnemer, "Smart braille system recognizer," *IJCSI International*

Journal of Computer Science Issues, vol. 10, pp. 1694-0814, 2013.

[9] A. Abualkishik and K. Omar, "Framework for translating the Holy Quran and its reciting rules to Braille code," in *Research and Innovation in Information Systems (ICRIIS), 2013 International Conference on*, 2013, pp. 380-385.

[10] D. S. A. Damit, A. I. C. Ani, A. I. Muhamad, M. H. Abbas, and F. Z. Ali, "Dual braille code translator: Basic education tool for visually impaired children," in *Computer, Communications, and Control Technology (I4CT), 2014 International Conference on*, 2014, pp. 399-402.

[11] S. D. Al-Shamma and S. Fathi, "Arabic braille recognition and transcription into text and voice," in *Biomedical Engineering Conference (CIBEC), 2010 5th Cairo International*, 2010, pp. 227-231.

[12] E. Chekol, "Recognition of Amharic Braille documents," *Unpublished Master Thesis, Faculty of Informatics, Addis Ababa University, Addis Abeba, Ethiopia,* 2010.

[13] M. Gadag and D. V. Udayashankara, "Generation of English Text from Scanned Braille Document," *International Journal of Engineering Science,* vol. 4446, 2016.

[14] A. Sanin, C. Sanderson, and B. C. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods," *Pattern recognition,* vol. 45, pp. 1684-1695, 2012.

[15] H. Izakian and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem," *Expert Systems with Applications,* vol. 38, pp. 1835-1838, 2011.

[16] M. Arif and M. U. Akram, "Pruned fuzzy K-nearest neighbor classifier for beat classification," *Journal of Biomedical Science and Engineering,* vol. 3, p. 380, 2010.

[17] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering,* vol. 26, pp. 1819-1837, 2014.