International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 10, No.1 (Jan-2021)

http://dx.doi.org/10.12785/ijcds/100113

English Characters OCR Pertinent for Mobile Devices

Saleh Alshehri

Department of Computer Science and Engineering, Jubail University College, Royal Commission at Jubail, Jubail Industrial City, Saudi Arabia

Received 5 Apr. 2020, Revised 25 Jul. 2020, Accepted 25 Aug. 2020, Published 1 Jan. 2021

Abstract: Optical character recognition concerns translating character images into character text. Four main stages constitute any optical recognition system. They are pre-processing, feature extraction, character recognition, and post-processing. The two most important stages are the feature extraction method and the character recognition algorithm. The optical character recognition system for mobile application must fulfill two requirements, small system size and high speed. In this research, a lightweight footprint feature dataset was created based on the center and centroid of the character image together with other simple image statistics. Fast character recognition algorithm based on weighted Euclidean distance was adapted. A mobile phone application prototype was developed. An accuracy of over 99% was achieved. The execution time for recognizing one character was in millisecond scale when a mobile phone model was used.

Keywords: OCR, Mobile Devices, Euclidian Distance

1. INTRODUCTION

Optical character recognition (OCR) has been the subject of research for decades. It has reached a mature state in most parts of it. The OCR algorithms vary where each has its unique features. However, the goal is the same, which is to convert the character image pixels into text characters. The focus of these algorithms can be on one or more of the three features; accuracy, speed, and storage capacity [1]. For example, in automatic mail forwarding applications, the accuracy is the main target since a mistake in one character may generate the wrong forwarding address. However, in large database document retrieval systems, the focus is on speed. In contrast, speed and storage capacity are the targets in mobile device OCR apps [2]. This is obvious because of the limited capacity and lower execution power that mobile devices have.

Any OCR system is built on finding suitable features of the character images that can be used to classify the image, and then developing the classification algorithm. There are many surveys that explain several feature extraction methods and classification and recognition algorithms [3-7]. The character document image's physical and logical structure, and how its different parts relate, lead to the proper feature extraction method [8]. These features are either structural, statistical, or a mixture of both [9,10]. Some possible feature extraction methods are histograms, invariant moments, zoning, x and y projection, n-tuples, crossings, and distances [3]. Based on these features, the algorithms are built.

The number of OCR applications is huge [3]. It ranges from simple applications such as invoice reading, postal address reader, and vehicle plate recognition, to vast document retrieval [9,11].

There are generally four stages in a complete OCR process. They are pre-processing, segmentation, feature extraction, and classification and post-processing [12]. The feature extraction and classification stages are the focus on this research. Simple pre-processing method including binarization can be found in the literature [13]. Other pre-processing and post-processing steps, which were built for mobile phone applications, were developed [14-16].

OCR can be built for printed and handwritten characters. In addition, it can be built for the English language or any other language. The nature of each language dictates the feature extraction method and the recognition algorithm. For example, Arabic writing starts from right to left and the characters are in cursive script [17-19].

Even though, there are many OCR applications for mobile phones, the need for fast and lightweight footprint system still exists [10]. The lightweight footprint system is import since it occupies less space. For example, Tesseract OCR which is one of the most prominent open

E-mail: shehri@ucj.edu.sa

source OCR software has a size of almost 13 MB [9]. Whereas the proposed system is almost 10 times smaller in size than Tesseract. The proposed technique also presents a method to qualify the importance of OCR recognition system parameters which leads faster recognition. This research focuses on building a lightweight footprint features dataset together with a fast algorithm that fits for mobile phone applications.

The preprocessing and post-processing stages are not considered in this research since developed methods were available in the literature.

2. OCR Feature Extraction and Recognition

The proposed OCR algorithm starts by building characters feature dataset. This feature dataset is used later to classify any character based on the similarity between this character extracted feature and the feature dataset. The focus was on building a small footprint dataset. Small size dataset consumes small memory space and needs less computation power to process. These two attributes encourage using it in the mobile devices. In addition, it can be used in any other larger device utilizing these two features.

A. Character strcture features

The centers of images with similar sizes are identical. However, the centroid location may not necessary be consistent. It depends on the shape of the character. It would be expected that the centroids of characters with different fonts be close to each other's. The line connecting the center and the centroid produces an angel that has some consistency among the same character in various fonts. A large database was used in this research where each character has 1016 different fonts [20]. It consists of 62 sets. Each set contains 1016 images corresponding to the character fonts. The image size is 128 x 128 pixels. Figure 1 shows examples of characters in the database.

Figure 1. Samples of the characters in [20]

Let the centroid point be at (c_x,c_y) and the center of the character image is at (x_c,y_c). Let (h) and (w) be the character image height and width respectively. The straight-line equation is given by Eq (1):

$$y = \frac{c_y - y_c}{c_x - x_c}(x - x_c) + y_c \qquad (1)$$

The length of the straight line becomes:

$$r = \sqrt{(c_y - y_c)^2 + (c_x - x_c)^2} \qquad (2)$$

The angel is given by Eq (3):

$$\theta = \tan^{-1}\left(\frac{c_y - y_c}{c_x - x_c}\right) \qquad (3)$$

This process is done for both the character image and its negative image. Figure 2 shows the line connecting centroid and center and the angel of one font of the character 'b'.

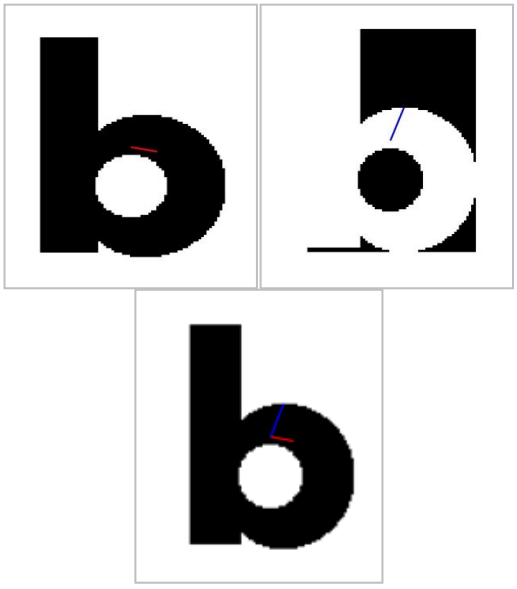

Figure 2. Straight lines connecting centeroid and ceter of 'b' image and its negative image.

Since the characters images vary in sizes, the line length may vary for the same character and the same font. To extract this feature with consistency, the normalized x-projection and y-projection of the line are calculated with respect to the x and y lengths. Eq (4) and Eq (5) show how to get these two values. However, the angles values are kept as they are since they are not affected by the image size.

$$x_{proj}^{norm} = \frac{r * \cos\theta}{|w/2|} \qquad (4)$$

$$y_{proj}^{norm} = \frac{r * \sin\theta}{|h/2|} \qquad (5)$$

The three previous features (θ, x_{proj}^{norm}, y_{proj}^{norm}) are calculated for the character image. It is also calculated for the negative image (θn, xn_{proj}^{norm}, yn_{proj}^{norm}). This will

constitute six features. Since each character normally has different dimensions, then it would be an advantage to calculate the height to with ratio (*HW*) and the ratio of the number of pixels to the image size (*PR*) as in Eq (6) and Eq (7) respectively.

$$HW = \frac{h}{w} \qquad (6)$$

$$PR = \frac{no.of\ pixels}{h*w} \qquad (7)$$

The resultant feature vector is [θ, x_{proj}^{norm}, y_{proj}^{norm}, θn, xn_{proj}^{norm}, yn_{proj}^{norm}, *HW*, *PR*]. These features were calculated for all 62992 characters images in the dataset producing a data matrix of size 62992 x 8.

B. Character recognition

Once the feature dataset is created, the recognition algorithm can be applied. There are many algorithms to choose from. Each of which has its focused applications. Some are good for fast recognition and others have excellent recognition accuracy. In this research, the two main features are the execution speed and the storage size requirement.

For object recognition, object distance algorithm is known with its various approaches [21]. For its simple arithmetic calculation, Euclidian distance is used in this research. Euclidian distance between a given vector (v) and all vectors in the dataset (v_i) is obtained using Eq (8). The minimum of these distances indicates the recognized character as in Eq (9).

$$FD_i = \sqrt{\sum_{i=1}^{n}(v - v_i)^2} \qquad (8)$$

$$RC = \text{argmin}(FD_i) \qquad (9)$$

where n is the size of the dataset

It is found that some features contribute to describing the character more than others. Hence, they should be given more weight during classification stage. Figure 3 shows each normalized feature in the dataset where x axis is the normalized dataset size.

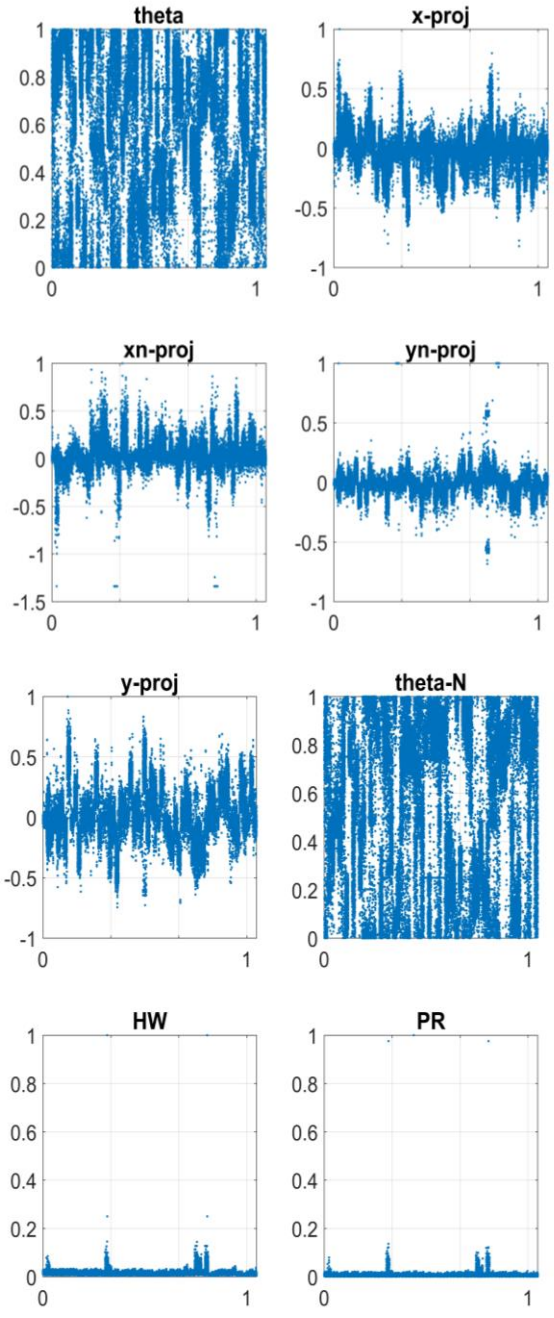

Figure 3. Normalized features of the characters in the dataset

To find out the expected contribution of each feature, the ratio of the standard deviation to the mean of the feature was calculated as in Eq (10). This is because larger value of standard deviation with respect to the mean value indicates that this feature span on larger sample space.

$$W_i = \left|\frac{std(f_i)}{mean(f_i)}\right| \qquad (10)$$

where *i* is the number of features

W_i is the feature's contribution absolute value and (std) is the standard deviation. The result W_i = (0.50, 8.08, 29.10, 0.57, 6.92, 20.40, 0.68, 1.51) are weights emphasis on Euclidean distance. Eq (8) became as in Eq (11).

$$FD_i = \sqrt{\sum_{i=1}^{n}(1/W_i^2)(v - v_i)^2} \qquad (11)$$

The character would be the minimum argument of FD_i. The formula was applied on the dataset where each character vector is compared with all vectors in the dataset. The character fonts are 1016 for each character in the dataset. This number is very large and some of the fonts are very rarely used. Part of the fonts; about 20, 200 and 500 fonts of each character were also investigated. The results of applying various distance measures algorithms based on Matlab Euclidean distance function are shown in Table 1. It can be recognized from the table that Euclidean with weights algorithm slightly overcome all other algorithms in terms of accuracy. However, it takes little more execution time that simple Euclidean algorithm would take.

TABLE I. PERCENTAGE ERROR AND EXECUTION TIME USING VARIOUS VERSIONS OF EUCLIDEAN DISTANCE MEASURE

Algorithm	20		200		500		All	
	% Error	*Execution time*	*% Error*	*Execution time*	*% Error*	*Execution time*	*% Error*	*Execution time*
Euclidean	0.0806	9.151x10-5	0.0807	21.120x10-5	0.207	1.001x10-3	0.281	2.098x10-3
Euclidean with weights	0.0756	11.656x10-5	0.0757	25.211x10-5	0.201	1.108x10-3	0.277	2.270x10-3
square deuclidean	0.0806	9.530x10-5	0.0807	20.616x10-5	0.207	0.966x10-3	0.281	2.093x10-3
cityblock	0.0806	14.260x10-5	0.0807	20.549x10-5	0.207	0.965x10-3	0.281	2.070x10-3
minkowski	0.0806	9.506x10-5	0.0807	21.549x10-5	0.207	1.001x10-3	0.281	2.120x10-3
chebychev	0.0806	12.971x10-5	0.0807	28.865 x10-5	0.207	1.196x10-3	0.281	2.678x10-3
cosine	0.0806	23.800x10-5	0.0807	85.271x10-5	0.207	4.752x10-3	0.281	10.545x10-3
correlation	0.0806	28.069x10-5	0.0807	97.742x10-5	0.207	5.621x10-3	0.281	11.928x10-3
hamming	0.0806	11.709x10-5	0.08071	33.152 x10-5	0.207	1.507x10-3	0.281	2.859x10-3

C. Storage capacity reduction

The dataset size that can be used ranges from 1240 x 8 to 62992 x 8. Depending on the number of fonts needed to be covered in the OCR system. In all cases, the data is mostly real numbers. To store real number in 32-bit system, 4 bytes are needed. In 64-bit system it is 8 bytes. Less number of bytes can be used to store integer numbers if needed. For example, 2 bytes are enough to store an integer number if the maximum is 216. This can be specified in the programing language being used. It is also possible to use less number of bits by dedicated coding during storing stage. Then during the OCR recognition, the dataset is expanded and standard integer format.

The dataset was analyzed to find the minimum and maximum of each feature and process it to scale the real numbers to integers. Figure 4 and figure 5 show the values of each feature before and after value scaling. It can be recognized that the values which are less than "1" are very rare in each feature. This makes it convenient since coding the feature value (*FV*) using integers will not cover those values which are less than "1".

By observing the maximum values of each feature, the number of bits needed to code these FVs are determined. The scaling was done as follows:

$$FV_i = FV_i - \min(FV)$$

The FVs then scaled by S = (1, 128, 128, 1, 128, 128, 8, 32).

$$FV_i = FV_i * S$$

The maximum values became (359.99, 200.31, 113.71, 359.98, 223.59, 211.74, 189.75, 1399.1). The number of bits to code each bits are (9, 8, 7, 9, 8, 8, 8, 11) by taking log2 of the maximum values and round it up. This gives a total of 68 bits for each entry in the dataset. The total reduction in the dataset size is %73.4 if 32-bit system is used and 86.7% if 64-bit system is used. In case all fonts for all character are used, the dataset size becomes 535 kB instead of 2.02 MB or 4.03 MB if 32-bit or 64-bit systems are used respectively.

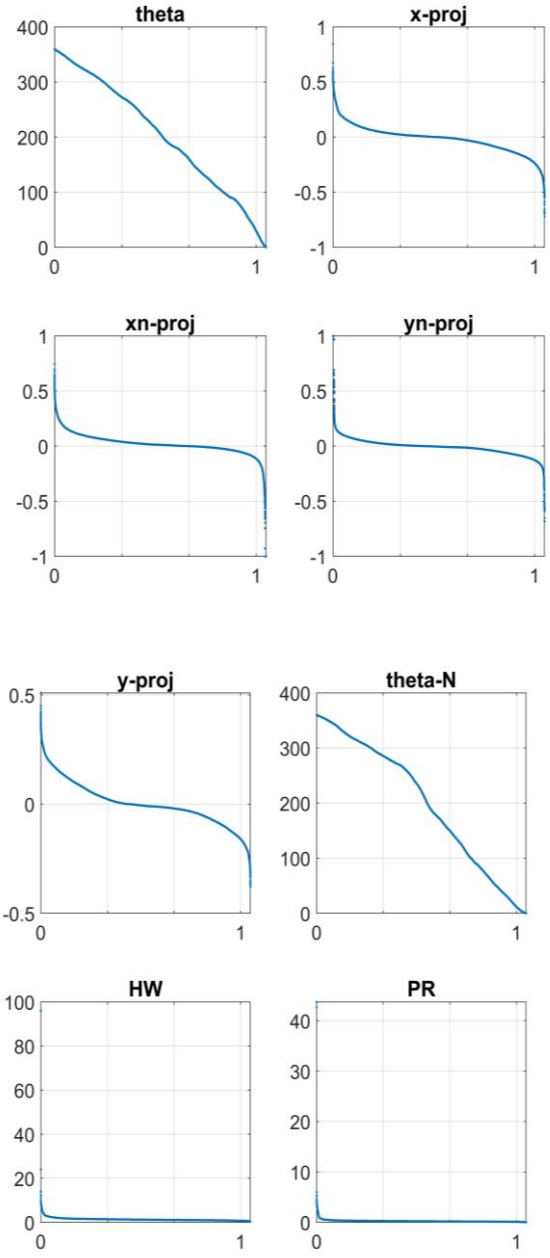

Figure 4. Features sorted values

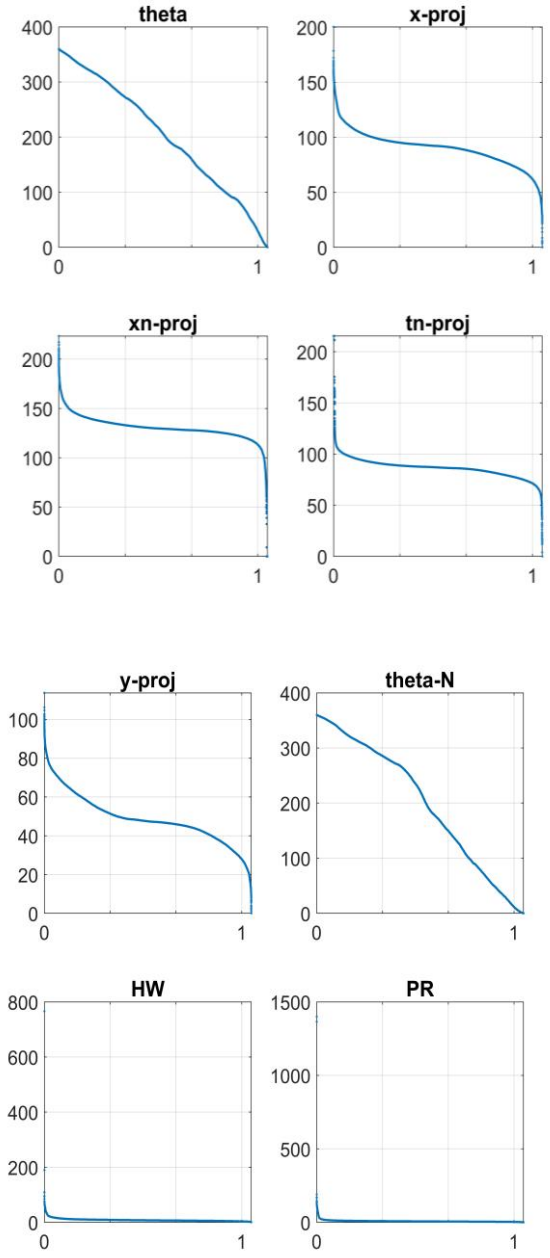

Figure 5. Features sorted values after scaling

3. PROPOSED ALGORITHM EXECUTION ON MOBILE DEVICES

The proposed OCR algorithm was developed for iPhone 7 mobile phones. The test focused on the speed and storage capacity. First, the dataset was migrated onto the iPhone 7. The dataset size in the device was shown to be 4336931 bytes. iPhone 7 "Apple A10 Fusion" has a 64-bit processor which is a 64-bit ARM-based system on a chip with four cores. It has 2 GB of RAM and uses 32 GB, 128 GB, or 256 GB of flash storage. Its frequency is between 1.64 and 2.34 GHz [22].

Two versions of the dataset were tested. The first one was the dataset with real numbers while the second one was scaling and interceded one. GUI was built to accept the character image data as a vector. Then this vector was processed against the dataset and the resulting character was displayed. Figure 6 shows the mobile phone GUI. The execution time was found to be 0.46446 second for the first dataset while it was 0.46442 for the second dataset. This means, two characters can be recognized in a second. Therefore, a standard page with 80 characters by 25 lines would take about 16 minutes to be completed. This indicates that the use of the integer-coded dataset was comparable to real number representation in terms of speed. However, the integer represented dataset size was 535432 bytes, which gives size reduction of about 87%.

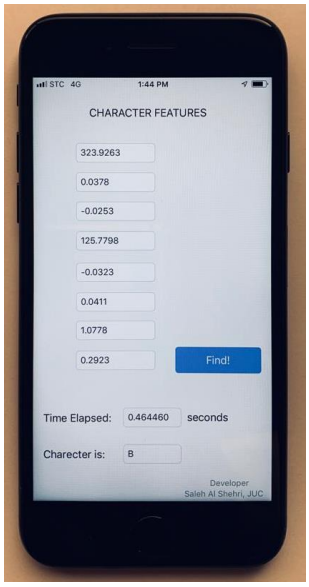

Figure 6. iPhone 7 running the application

4. CONCLUSIONS

When developing OCR, the system may inforce certain requirements. Mobile devices need lightweight footprints and fast systems. In this research, these two goals were achieved. A lightweight feature dataset was built based on the straight line connecting the centroid and the center of the character image. The angle generated by this line and its length projections to the x and y axes were used as feature data points. A weighted Euclidean distance was used as the recognition algorithm. Various versions of the Euclidean distance algorithm were investigated to find the best of them. The feature dataset was reduced using integer representation in binary form instead of real numbers. The dataset size reduced was 87%. The execution time was reasonable for mobile devices.

REFERENCES

[1] Eugene Borovkov, "A survey of modern optical character recognition techniques", 2014.

[2] Sravan Ch, Shivanku Mahna and Nirbhay Kashyap, "Optical Character Recognition on Handheld Devices", International Journal of Computer Applications, Vol. 115, No. 22, pp. 10-13, 2015.

[3] Amarjot Singh, Ketan Bacchuwar, and Akshay Bhasin, "A Survey of OCR Applications", International Journal of Machine Learning and Computing, Vol. 2, No. 3, pp. 314- 318, 2012.

[4] Noman Islam, Zeeshan Islam,Nazia Noor, "A Survey on Optical Character Recognition System", Journal of Information & Communication Technology-JICT Vol. 10 Issue. 2, pp. 1-4, 2016.

[5] Rohit Verma and Jahid Ali, "A-Survey of Feature Extraction and Classification Techniques in OCR Systems", International Journal of Computer Applications & Information Technology, Vol. I, Issue III, pp. 1-3, 2012.

[6] Sukhpreet Singh, "Optical Character Recognition Techniques: A Survey", Journal of Emerging Trends in Computing and Information Sciences, Vol. 4, No. 6, pp. 545- 550, 2013.

[7] Masum Mohammad et al., "Automatic knowledge extraction from OCR documents using hierarchical document analysis", Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems,October 2018, pp. 189–194.

[8] Song Mao, Azriel Rosenfeld, and Tapas Kanungo "Document structure analysis algorithms: a literature survey", Proc. SPIE 5010, Document Recognition and Retrieval X, (13 January 2003).

[9] Stevan Cakic et al., "The Use of Tesseract OCR Number Recognition for Food Tracking and Tracing", 2020 24th International Conference on Information Technology, Zabljak, Montenegro, 18-22 Feb. 2020.

[10] Arindam Chaudhuri, Pratixa Badelia and Soumya K. Ghosh," Optical Character Recognition Systems for Different Language with Soft Computing", Springer, 2017.

[11] S.L. Chang, T. Taiwan, L.S. Chen, Y.C. Chung, S.W. Chen, " Automatic license plate recognition" in IEEE Transactions on Intelligent Transportation Systems, 2004, Vol: 5 , Issue: 1, p.p. 42 – 53.

[12] Nabil Aharranw, Kaim El Moutaouakil and Khalid Satori, "A comparison of supervised classification methods for a statistical set of features: Application: Amazigh OCR" in Intelligent Systems and Computer Vision (ISCV) 2015, IEEE, pp. 1-8, March 2015.

[13] Saleh Alshehri, "Document Image Binarization Method That Compromises Between Global and Local Thresholding Techniques and Automates the Free Parameter Selection," Global Conference on Computer Science, Software, Network & Engineering, November 06– 08, 2014 in Turkey.

[14] J.C. Burie et al., "ICDAR2015 Competition on Smartphone Document Capture and OCR (Smartdoc)", International Conference on Document Analysis and Recognition (lCDAR), 2015.

[15] Saleh Alshehri, "OCR for Mobile Phone App Based on Partial Projection of Letter Pixels," International Journal of Applied Engineering Research, pp. 9180-9184, 2016.

[16] Mithe, Ravina, Supriya Indalkar, and Nilam Divekar. "Optical character recognition." International Journal of Recent Technology and Engineering (IJRTE) 2.1, 2013.

[17] Abdullah I. AlShoshan, "Arabic OCR Based on Image Invariants", Proceedings of the Geometric Modeling and Imaging-New Trends (GMAI'06), 2006.

[18] Mohammad Tanvir Parvez and Sabri A. Mahmoud, "Offline Arabic Handwritten Text Recognition: A Survey", ACM Computing Surveys, 45(2), 23–23:35.

[19] Sohail Abdul Sattar, Salman Shah, Character recognition of Arabic script languages, in Proceedings of the International Conference on Computer and Information Technology (ICCIT'12), 2012.

[20] T. E. de Campos, B. R. Babu and M. Varma, "Character recognition in natural images.", In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), Lisbon, Portugal, February 2009.

[21] Jie Zhang, Jiazhen Pang, Jianfeng Yu and Pan Wang, "An efficient assembly retrieval method based on Hausdorff distance," Robotics and Computer-Integrated Manufacturing, 51, pp. 103-111, 2018.

[22] Cunningham, Andrew, "iPhone 7 and 7 Plus review: Great annual upgrades with one major catch". Ars Technica. 2016.

Saleh A. Alshehri received BSc, MSc and PhD degrees in computer engineering from King Fahd University of Petroleum and Minerals, King Saud University and University of Putra Malaysia respectively. His research interests include electronic cooling, image processing, Ultra-Wide Band (UWB) imaging and pattern recognition. He is an associate professor in computer science and engineering department at Jubail University College (JUC). Currently he is the director of Prince Saud Bin Thonyan research center at Royal Commission in Jubail.