Ain Shams Engineering Journal 14 (2023) 102178

Contents lists available at ScienceDirect

Ain Shams Engineering Journal

journal homepage: www.sciencedirect.com

# A low cost IoT-based Arabic license plate recognition model for smart parking systems

Mohammad M. Abdellatif [a,*], Noura H. Elshabasy [a], Ahmed E. Elashmawy [a], Mohamed AbdelRaheem [b]

[a] Electrical Engineering Department, The British University in Egypt, Cairo, Egypt
[b] Electrical Engineering Department, Assiut University, Assiut, Egypt

ARTICLE INFO

*Article history:*
Received 7 October 2022
Revised 6 January 2023
Accepted 17 January 2023
Available online 28 January 2023

ABSTRACT

License Plate Recognition is one of the significant enablers that can be utilized in wide range of applications in ITS and smart cities. The proposed design relies on three image processing stages to achieve license plate identification with high accuracy which are pre-processing, segmentation, and character recognition. The canny edge detection method with various thresholds, contour detection, and masking techniques are used to locate the car edges and license plate. In the experiment presented in this paper, 200 images were used to identify Egyptian car plates. The model successfully identified Arabic license plates with 93% accuracy. A prototype is implemented using ESP32 Cameras and Raspberry-Pi to test the system's performance. Moreover, a database and a website are hosted on the RPi to allow users to search for their car location in the parking lot using the car's full or partial license plate which was saved in database upon detection.



## 1. Introduction

With the rapid increase of the number of vehicles in an average city, the problem of locating an empty parking place have become a great inconvenience. Additionally, there is always the problem of trying to remember where the vehicle was parked, especially in a crowded parking lot of a shopping mall or a stadium. Moreover, there are many other common problems such as the large number of stolen cars, speed violations, and unregistered cars. All of these issues can be managed by having a license plate recognition system.

Parking problems have arisen due to the increasing number of vehicles on the streets and the poor management of available parking spaces. Fortunately, intelligent parking systems offer a solution [1–3]. Due to the enormous number of vehicles available in the parking, in addition to having more than 3 floors on average for parking, it is very frustrating to remember where you have parked. That's why this paper proposes the creation of a terminal and an application that can easily and efficiently tell the driver where their car is parked.

Smart parking systems through space detection is a great candidate to reduce the dilemma of where to park and in turn, minimize the air pollution caused by vehicles in indoor areas. Most of the contemporary smart parking systems make use of a device installed in each parking lot that detects the presence of absence of a vehicle [2], and print numbers and letters in large font next to each lot to help drivers return to their vehicles. However, since most of the indoor parking places have cameras, those cameras can be positioned in a way to take a picture of the parking lot focusing on the vehicle's license plate, then apply image processing techniques to detect the license plate number and store it in a local database that can be accessed from anywhere in the parking lot's network [2,4].

This system can be exploited in many areas other than parking lots, such as, speeding violations, security management, police station. Any entity can modify the system and use it to detect the thing that matters for them. The automated car license plate recognition models are of great importance in intelligent transport system (ITS) [5]

Optical Character Recognition (OCR) is the key technique used for this system, it is used for detecting and identifying the vehicles from their plate numbers. The car plate recognition system also uses image processing techniques such as feature extraction and representation methods.

* Correspondence author at: The British University in Egypt, Misr-Suez Road, El Shorouk City, Cairo, Egypt.

*E-mail addresses:* Mohammad.abdellatif@bue.edu.eg (M.M. Abdellatif), noura.elshabasy@bue.edu.eg (N.H. Elshabasy), ahmed.emam@bue.edu.eg (A.E. Elashmawy), m.abdelraheem@aun.edu.eg (M. AbdelRaheem).

https://doi.org/10.1016/j.asej.2023.102178
2090-4479/

The aim of this paper is to use inexpensive IoT devices such as ESP32CAM and RaspberryPi (Rpi) to monitor a parking lot, extract the license plate number, and store that number and its location in a database that can be accessed remotely from the parking lot's network. This way, any driver who wants to locate their vehicle, can connect from their phone or a terminal installed in the parking lot, enter a partial or the full license plate, and the system will direct them to their vehicle's location. This work is an extension of the work done by the same authors in [4]. However, the previous paper addressed the idea and how to do it with English letters. In this paper, the prototype has been tested with Arabic license plate numbers. Additionally, the processing technique is similar to the one used in [6] but they did not implement it on real setup.

The rest of the paper is organized as follows, Section 2 gives a brief overview of the related work and the contribution of this paper. Section 3 presents the methodology, the system model, and illustrates the detailed license plate detection algorithm. Section 4 describes the prototype implementation. Section 5 presents the performance analysis of the system. And finally, Section 6 provides concluding remarks about the proposed work.

## 2. Related work and contribution

Many researchers have investigated the area of image processing and text recognition and the best techniques to be used to extract the text purely and enhance the accuracy of the system. In [2], the authors proposed a technique with 81% accuracy to calculate the number of parked cars in any indoor parking area. They used a comparison result technique between the plate position and the maximum value position to count the number of cars. They have used the Retinex method, the edge processing histogram, integral intensity projection, smoothing filter, first-order derivative, and peak detection to detect the license plate area. In the proposed system, due to the limited processing power of the IoT devices, simpler OCR techniques are used.

The authors in [3] presented a convolutional neural network system for license car plate recognition and it resulted in 90% accurate trials in 100 times in Sudan. The idea of region-based convolutional neural networks has been used for detection and localization. The faster-RCNN technique outperformed in accuracy than the pure technique. This system has used Convolutional Neural Network in order to increase the robustness of the system. However, RCNN requires high processing power that limits its deployment in IoT devices.

Authors in [6] used the canny algorithm with projection technique to develop a system for Arabic car plate detection and produced accuracy of 92.4% and 96% for Arabic and English language respectively. The ORC algorithm used in the proposed system follows a similar methodology. However, the authors did not test in a real life scenario.

In [7], the authors studied an Automatic License Plate Recognition (ALPR) system in Spain with the help of an infrared (IR) camera. Similarly in Lebanon and Saudi Arabia, Arabic Alphabets have been included in the interests and many trials were made to develop similar approaches in Arab countries, seeking for a well addressed smart parking areas. The proposed system uses similar concepts but with low-cost cameras.

Authors in [8] presented a deep literature review on the existing models in Arabic text recognition. It detailed the execution, inform the differences, pros and cons that aids in adopting or enhancing these models in accordance with the advanced requirements.

In [9], a group of students created a similar prototype as the one presented in this paper, an IoT based cloud integrated system is deployed to ease the parking system of a university with QR codes and parking flaps. Functional and non-functional requirements are discussed. With the help of an IoT module, the availability of the parking spots is monitored. The system can reserve a slot for a car, pay for billing and check available parking slots. However, they do not focus on the license plate numbers and are not able to help in finding a parked vehicle.

Lastly, authors in [9] developed an on-board vehicle transceiver device (VTD) that can be added without the need to install new pieces in each parking area. Through this device, users will be able to reduce many expenses that come from installations and complexity in the system, infra-structure investment. A sensing technique composed of motion detection and parking occupancy monitoring is achieved by a two-fold sensing approach. The sensor (device) gives a signal when the vehicle is in a parking area, by the help of a proposed radio frequency (RF) wake-up technique.

**Main Contribution:** The system proposed in this paper uses low cost IoT devices that are easy to install. They run an ORC algorithm that does not require high processing power. Moreover, the extracted text is placed in a local database and results can be searched by the users from anywhere in the network. An experimental prototype is created to test the performance of the system and its feasibility in smart cities applications. Smart cities require high speed, easy interfaced and low-cost applications and that's what this paper is targeting.

## 3. Methodology and detailed algorithm

This paper proposes a cost-effective smart system that can reduce the amount of time spent searching for parked vehicles in a mall garage or a parking lot as well as help locate the already parked vehicle. Fig.1 shows the proposed system model which includes two main blocks, the image capturing block and the image processing block. The image capturing block is the first step in Fig.1 which is achieved by using ESP32 cameras; while the image processing technique is divided into three stages, as indicated by steps 2 to 4 in Fig.1, which are:

1. Pre-processing: Modifying the captured image of the license plate to be in an acceptable format to allow the next stages to perform adequately and increase the accuracy of our model.
2. Segmentation: Splitting the modified image into sub-images of classifiable objects (characters or numbers) for the OCR to successfully detect the license plate characters.

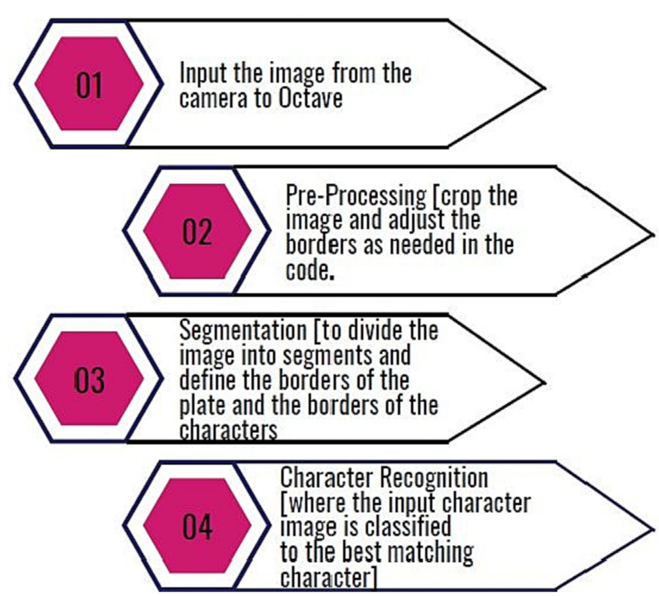


**Fig. 1.** License care plate recognition steps.

3. OCR: Identifying the license plate in question where optical character recognition is used to identify the individual characters and numbers from the sub-images obtained from the segmentation stage.

The text output of these stages is stored in a database along with the additional details pertaining to the application using our system model. For example, in our test case scenario, this system model is used in a University parking lot where the image of the car's license plate is captured upon parking in a certain parking place. Each parking space is equipped with a wireless camera that is tagged with a space number and identified in the network by an IP address. The license plate after being identified is then fed into the database along with the parking lot's number. Any user in the network can search the database from a local website for their car using the full or partial license plate number. Upon receiving the request, the website displays the exact location where the car is parked, along with a time-stamped photo of the parked car. The paper follows the initial steps made in [4] and practically works on implementing it in actual garage. The garage is a single floor, divided into multiple lots.
The system model's accuracy relies on using the proper methods in the image processing technique. The three stages in our algorithm are:

A. **Pre-processing**
The performance of any image processing technique is dependent on adequately adjusting the captured images and presenting them in an acceptable format that the subsequent stages can perform well on. This takes place in the pre-processing stage where the captured images pass through a series of processes such as resizing, rotation and scaling. The modified images are then converted to gray-scale from its original RGB format. This step is necessary to add speed and robustness in processing the images in the following stages. The gray-scale image is then interpolated using the nearest neighbor technique to produce a smoother image to operate on [10].
Then, Canny edge detection technique [11] is used to identify the boundaries followed by contour detection to locate the license plate. Canny edge detection operates on the image through several stages to be able to reduce the noise and locate the region of interest. The first stage is applying Gaussian filter to remove the high frequency noise that may affect the accuracy of edge detection. This filter results in image blurring which is useful for better edge detection. In our design, we relied on bi-lateral filter which is a non-linear filter with weights that are Gaussian distributed [12]. These weights take into consideration both the radiometric differences between pixels as well as the Euclidean distance of pixels. The filtered image can be written as a filtered function of the input image as follows:

$$R(x) = W_g^{-1} \sum_{x_j} W_i(||S(x_j)S(x)||)W_x(||x_j\, x||) \tag{1}$$

Where R(x) is the filtered image, S(x) is the original image, $W_i$ is the radiometric weights based on intensities which is Gaussian distributed, $W_s$ is the spatial weights for smoothing the spatial coordinates which is Gaussian distribution, x are the coordinates of the current pixel, $\gamma$ is the square window size centered at x and $W_g$ is the normalization weight $W_g$-which is defined as

$$W_g = \sum_{x_j \in \gamma} W_i(\|S(x_j) - S(x)\|)\, W_x(||x_j - x||) \tag{2}$$

The next step is to find the intensity gradients of the image. Sobel operator can be used to calculate the first derivative along both the horizontal (x-axis) $G_x$ and vertical (y-axis) $G_y$ directions [13]. Then, the edge gradient can be calculated from the outputs of Sobel operator as follows: $\sqrt{G_x^2 + G_y^2}$ where the mask values are

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \text{ and } G_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Then a minimum cut-off suppression technique is applied to detect strong edges where the edge strength of each pixel is compared with the neighboring pixels in each gradient direction and the strongest pixel within the mask window is preserved while the other pixels are suppressed. At this stage, the real edges are detected; however, there are some noises that might exist resulting in falsely detected edges that might affect the performance of the subsequent stages. Two additional steps are considered to account for these falsely detected edges which are

- **Dual threshold:** Two thresholds are set, a high and low threshold on the gradient values. If an edge pixel has a gradient value higher than the high threshold, it is marked as a strong edge. If it is in-between the two chosen thresholds, it is marked a weak edge. But if it falls below the low threshold, it is suppressed. The reason for preserving both strong and weak edges at this stage is as follows: If a weak edge is connected to a strong edge, then it is most probably a real edge, but if it is unconnected, then it is most probably a noise.
- **Hysteresis:** This is a tracking technique to identify weak edges that are connected to at least one strong edge using blob analysis (comparison to the 8 pixels in its neighborhood). Such pixels are preserved while the others are suppressed.

At this stage, the real edges have been detected. Contour detection is then applied to extract the license plate through finding a rectangle shaped polygon. The license plate is now ready to be fed to the next stage.

B. **Segmentation**
The segmentation process is necessary to split the image into sub-images that can be easily classified as individual characters and numbers [14]. This text segmentation relies on projection profiling technique for feature extraction. There are two directions to perform profile projection which are the horizontal projection and vertical projection where the projection histogram counts the pixel values in the horizontal and vertical directions respectively [15]. In LPR, only horizontal projection is needed to split the characters [4,5].

C. **OCR**
After splitting the characters, OCR is used to identify each character by comparing the binary formats to existing templates.

## 4. Prototype implementation

A full model diagram is shown in Fig. 2. The whole system cost about 150 USD to implement. The components used to build this prototype are:

- ESP32 Cameras.
- Raspberry Pi (RPi).
- Web server and Database server.

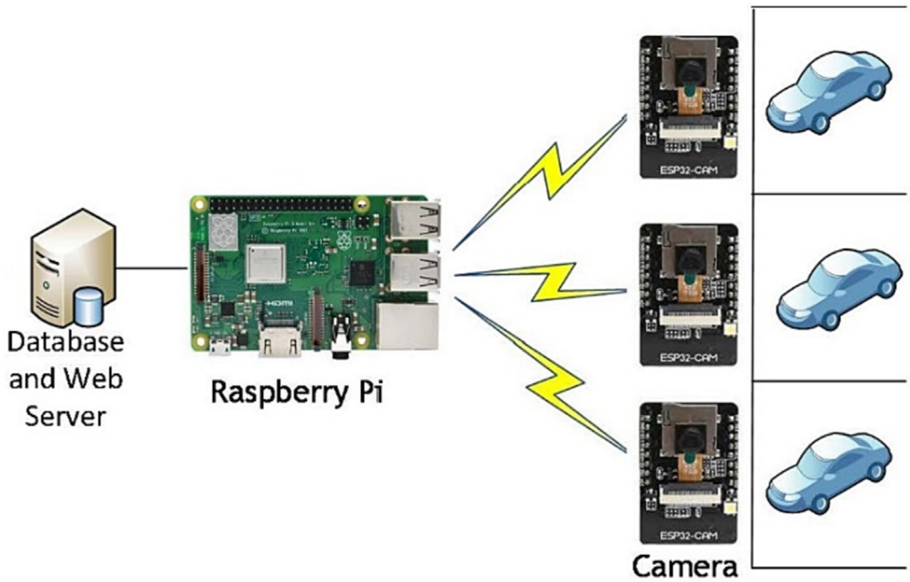


**Fig. 2.** Model diagram.

In this model, the Raspberry pi can also act as the access point, as well as the web and database servers.

Each camera is placed at a certain parking space and has a specific number related to its location. The distance between the camera and the plate was set to be about 1 m and the camera relied on the light source from the parking lot itself which is usually well lit. The camera is connected wirelessly to the RPi which instructs the camera to capture the required license plate image upon detecting a parked car at this specific location. Then, the image is locally processed on the Rpi and the license plate along with the necessary data are added to the database. A website hosted on the Rpi is designed to allow the user to search for their parked vehicle by entering a partial or full license plate. Upon entering the required data, the website fetches the matched license plate entry and all its related information from the database and presents it to the user through the website.

A conceptual view for the proposed model and its flow diagram is shown in Fig.3. The Camera is responsible for detecting the car in the parking lot and capturing the car's plate. Then the captured picture is sent to the RPi which supports OCTAVE for further processing. Octave is a free software that offers high level programming and can be used on RPi to perform complicated numerical calculation. An OCTAVE code was written that extracts the details of the license plate (Arabic letters and numbers) and saves them along with the location of the car in a database. The code was tested with multiple test license plates. Octave runs the script that uses OCR to extract vehicle plate characters. If the extraction is successful, the data will be saved to a database. If the extraction fails, the process will be repeated after a request is sent to capture a new image until the process successfully extracts the license plate and feeds it to the database. The main cognitive content of OCR is to construe text where restrained in an image under the standard encoding scheme representing them in ASCII format [8].The Raspberry Pi can be the hotspot and serve up to 19 clients and by this, one Rpi can serve a complete small garage on its own.

The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display and information extraction. Here, the captured image will go through different stages which are Pre-processing Stage, Segmentation and Recognition Stage.

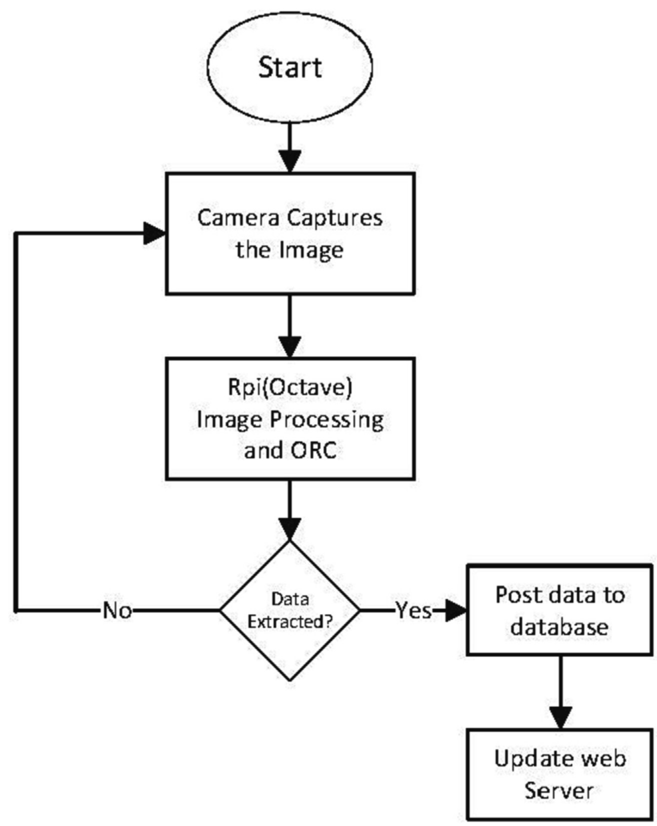


**Fig. 3.** Conceptual diagram for the proposed model.

The first stage, pre-processing stage, is where the image is cleaned and any unnecessary parts of the full image are removed, just crop the plate number only, as in Fig. 4. Then the image is converted to a binary image to easily distinguish the character section by defining a threshold (boundary). At this step, the Numerical section and the Alphabetical section should be separated to reduce any error in the recognition stage since it can recognize the "1"


**Fig. 4.** Pre-processing stage: (a) Cropped image. (b) Extracted characters. and (c) Segmentation stage.

as “”أ. The silver bar between the numbers and characters as shown in Fig.4 is the threshold.

The Segmentation Stage, as in Fig. 4, is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. The segmentation grandness lies in the fact that it will lead to easier and more exact character recognition. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. Each letter or number need to be recognized to obtain the real character; this would be done in the recognition stage by correlating the obtained character by the real character in the trained data set. The results are then saved in the database.

## 5. Performance analysis

### 5.1. Evaluation metric

There are several evaluation metrics that can be used to assess the performance of image processing models. These metrics include accuracy, precision, recall, and F1 score. Accuracy is the most commonly used metric which is the ratio of the correctly detected outcomes compared to the total number of trials (or samples). This metric is the one used in this paper to assess the performance of our system. There are normally four outcomes which are true positive (TP), true negative (TN), false positive (FP) and false negative (FN). The correctly detected outcomes are decided by the count of the TP and TN outcomes. Consequently, the accuracy can be defined as $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$

### 5.2. Analysis

In order to test this prototype, a stochastic collection of images is captured to test the proposed system. Simulation results can be classified into two phases: The success rate in detecting and extracting the plate, as well as the success rate in correctly recognizing the characters.


**Fig. 5.** The website's output.

**Table 1**
Proposed model results.

| Plate Number | Extracted Result as seen in the Terminal | Percentage of Success |
|---|---|---|
| 912 ج ف ي | 912 ج ف ي | 100% |
| 1342 أ ج ص | 342 ج ص | 70% |
| 237 ه ن ي | 32 ن ي | 66% |
| 7491 أ ب م | 491 ب م | 57% |
| 395 ن ط ب | 395 ن ط ب | 100% |
| 475 ب ل د | 475 ب ل د | 100% |
| 8126 ط ر د | 8126 ط ر د | 100% |

Fig. 5 shows the output of the web site after a user search using one letter of the license plate, the output was truncated for demonstrative purposes. The user types their partial or full car plate number and press the search button; the website displays all the results in chronological order. The vehicle's plate picture will appear with the location, the extracted plate number, and the timestamp of when it was parked.

Plate Extraction Stage is the main factor in the value of the accuracy of this model. If the code fails to extract then the OCR stage will be incorrect. Table 1 shows some samples of the recognition trials and the corresponding efficiency. After examining n samples where n is greater than 200 samples. It can be concluded that percentage of success is 93% which is a very promising result compared to the findings mentioned in the related work section.

Another prospective that should be taken into consideration is the time taken to analyze the picture and extract the data to be added in the database and the time it takes to appear on the terminal when requested. In our system, it took on average 5.7 s to extract the data and about 2 s for the results to appear on the website when requested.

As shown in Table 1, the errors mainly occur in “”أ and “”ء and this is a main challenge in all references as it extensively happens because of the similarity between “”أ and the threshold that separates the plate numbers and alphabets in the Egyptian car plates. Moreover, as stated in [8], there are many short marks in the Arabic texts. These marks can be placed above or below the alphabets. They are written as strokes and can alter recognition due to their location either above or below the characters, like dots and full stops.

In the future work, error of this nature could be reduced by employing more advanced image processing techniques or use Artificial Intelligence techniques to train the algorithm on better classification of letters. Additionally, a large-scale deployment will be made in order to fully test all aspects of the algorithm.

## 6. Conclusion

This paper shows that it is possible to have a low cost fully automated IoT system for smart parking. A terminal with a friendly interface allows the users to search using the license plate number and the system will reply with where and when was the vehicle parked. This is important for those who don't easily remember where they parked. Moreover, it can be used as a way to deduct the bill for the time you spent in the parking by easily adding a tab for paying the money according to the place you parked at.

The system is composed of an ESP32 wireless camera connected to the RPI wirelessly through a local Wi-Fi network. The RPi performs the image processing and OCR, as well as host the database and web servers. Results show that the OCR system works with high accuracy, and the whole system operates efficiently and smoothly.

## Declaration of Competing Interest

The authors declare that there is no conflict of interest with any institutions.

## References


[1] Kanan R, Arbess H. An IoT-based intelligent system for real-time parking monitoring and automatic billing. In: 2020 *IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*.

[2] Pramoun C, Sirikornkarn C, Pramoun T. Projection for automatic finding the vacant of car. In: 2019 *34th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*.

[3] Elhadi OS Yasir. Arabic-numbers recognition system for car plates. In: 2019 *International Conference on Computer,* Control, Electrical, and Electronics Engineering.

[4] Noura ES, Mohammad Mahmoud A. IoT for smart parking. In: 2019 International Conference on Advances in the Emerging Computing Technologies (AECT).

[5] Ghouwayel IK, Sweidan Y, Samir O, Ali A. An efficient algorithm for automatic recognition of the Lebanese car license plate. In: 2015 Third International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE).

[6] Antar R, Alghamdi S, Alotaibi J, Alghamdi M. Automatic number plate recognition of Saudi license car plates. In: Engineering, *Technology \& Applied Science Research*, Greece.

[7] Menon A, Omman B. Detection and recognition of multiple license plate from still images. In: 2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET).

[8] Alwaqfi Y., Mohamad M. A Review of arabic optical character recognition techniques \& performance. International Journal of Engineering Trends and Technology (IJETT) – Editor's Issues – 2020.

[9] Ajchariyavanich C, Limpisthira T, Chanjarasvichai N, Jareonwatanan T, Phongphanpanya W, Wareechuensuk S, et al. Park king: an IoT-based smart parking system. In: 2019 *IEEE International Smart Cities Conference (ISC2)*.

[10] Rukundo O, Cao H. Nearest neighbor value interpolation. In: arXiv preprint arXiv:1211.1768.

[11] Canny J. A computational approach to edge detection. In: IEEE Transactions on pattern analysis and machine intelligence.

[12] Tomasi CAMR. Bilateral filtering for gray and color images. In: Sixth international conference on computer vision (IEEE Cat. No. 98CH36271).

[13] Sobel I, Feldman G. A 3x3 isotropic gradient operator for image processing, 1968.

[14] Nobile N, Suen CY. Text segmentation for document recognition. In: Handbook of document image processing and recognition. London: Springer; 2014. p. 257–90.

[15] Mahto MK, Bhatia K, Sharma R. Combined horizontal and vertical projection feature extraction technique for Gurmukhi handwritten character recognition. In: 2015 International Conference on Advances in Computer Engineering and Applications.


**Mohammad M. Abdellatif** is an Associate Professor at the British University in Egypt (BUE). Dr. Mohammad holds a BSc '04 in Electronics and Communications Engineering from Assiut University, Assiut, Egypt, an MSc '06 in Telecommunication Engineering from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, and a PhD '15 in Telecommunications engineering from the electrical and computer engineering department of the University of Porto, Portugal. His research interest includes Internet of Things (IoT), Cognitive Radio Networks (CRN) and Wireless Sensor Networks (WSN).

**Noura H. Elshabasy** is a demonstrator in the British University in Egypt (BUE). She has received the B.Sc. degree in 2017 From the British University in Egypt. And currently is an M.Sc. student in Cairo University. Her research interest is in vehicular IoT systems, VVLC links and I2V.

**Ahmed E. Elashmawy** is an assistant lecturer in the British University in Egypt (BUE). He received his B.Sc. degree in 2011, and an MSc degree in 2018 from Mcgill University, Montreal, Canada. He is currently pursuing for his PhD degree from University of Waterloo, Canada. His research interests include Physical Layer security, Resource allocations, 5G, Cloud Radio Access Network, Optimization techniques, Cognitive radio, IoT, digital signal processing, Embedded systems, and hardware design.

**Mohamed Abdelraheem** received the B.Sc. and M.Sc. degrees in Electrical Engineering from Assiut University, EGYPT in 2004 and 2010, respectively, and the Ph.D. in Electrical and Computer Engineering from Virginia Tech, US in 2015. He is currently an Associate Professor of Electrical Engineering at Assiut University, Egypt. His research interests include wireless networking, IoT, and embedded systems. Dr Abdelraheem is currently the head of information technology department, Egyptian-German College of Technology (EGCTECH). He is a PI and Co-PI for a number of research projects from Egyptian funding entities like STIFA, NTRA, and ITIDA.