Computer Systems Science & Engineering
DOI:10.32604/csse.2021.014915
Article

Tech Science Press

Multi Criteria Decision Making System for Parking System

Manjur Kolhar* **and Abdalla Alameen**

College of Arts and Science, Prince Sattam bin Abdulaziz University, 11990, Saudi Arabia
*Corresponding Author: Manjur Kolhar. Email: m.kolhar@psau.edu.sa

Received: 27 October 2020; Accepted: 11 November 2020

Abstract: System supported smart parking can reduce traffic by making it stress free to locate empty parking spaces, hence lowering the risk of unfocussed driving. In this study, we propose a smart parking system using deep learning and an application-based approach. This system has two modules, one module detects and recognizes a license plate (LP), and the other selects a parking space; both modules use deep learning techniques. We used two modules that work independently to detect and recognize an LP by using an image of the vehicle. To detect parking space, only deep learning techniques were used. The two modules were compared with other state-of-the-art solutions. We utilized the You Only Look Once (YOLO) architecture to detect and recognize an LP because its performance in the context of Saudi Arabian LP numbers was superior to that of other solutions. Compared with existing state-of-the-art solutions, the performance of the proposed solution was more effective. The solution can be further improved for use in the city and large organizations that have priority parking spaces. A dataset of LP-annotated images of vehicles was used. The results of this study have considerable implications for smart parking, particularly in universities; in addition, they can be utilized for smart cities.

Keywords: Parking; LBP; DL; CNN; license plate

1 Introduction

The detection of parking space is an increasingly grave problem. However, the neural network and application-specific solutions are some approaches to solve this problem. Moreover, allocating parking spaces based on the LP of a vehicle and its preferred area is considered the best solution. Nevertheless, the solution proposed in this study is suggested for high traffic, mainly in a city or any large organization where this problem can be solved using an application development approach and deep learning CNN. We used three different datasets to validate our method for parking occupancy detection. The performance of deep learning techniques is directly proportional to the volume of training data. A higher amount of data demands dynamic network architectures with more constraints and layers. The increasing number of vehicles has complemented significant improvements in parking system design. Techniques such as machine learning (ML) convolution neural networks (CNNs) and deep learning- and application-level-based license plate (LP) detection and recognition are used in these designs. These next-generation systems use a variety of smart parking techniques. License plate (LP) detection and recognition modules use completely different processes to identify LP numbers. However, the process of identifying the characters of an LP are similar. In LP detection, many variables of the captured images are considered, such as the angle, distance between the LP and camera, illumination, and weather conditions (rain, sandstorm, or winter). If the captured image of the vehicle is unclear, then more time is required by the central processing unit (CPU) or graphics processing unit (GPU) to recognize, and eventually, the entire system's performance degrades. Additionally, the recognition phase includes LP character segmentation and identification of the detected LP region. Most ML algorithms fail to analyze the captured image because of noise and disruptive backgrounds [1–7]. However, deep learning techniques such as CNNs provide excellent quality output [8–10]. Therefore, in this study, we utilized CNNs and an application-oriented approach to identify and recognize LPs. Deep learning methods were applied with and without segmentations to recognize LP characters.

In this study, a parking system was designed using deep learning and an application-oriented approach. It presents an LP recognition system for Saudi Arabian standard LPs based on deep learning architectures and an application-oriented approach. The application-based approach and deep learning method were used to recognize and detect LPs at the front and back ends of vehicles, respectively. The deep learning method used a segmentation-based approach in which YOLOv3 was used for LP detection and recognition; in this approach, an image is split into a grid of cells to form a bounding box for object–class probabilities. To identify a parking space, deep CNN was employed. We considered three different datasets to validate our approach in parking occupancy detection because the performance of deep learning techniques is directly proportional to the volume of training data. A higher volume of data demands dynamic network architectures with additional constraints and layers. Many studies have proposed solutions that require heavy CPU usage, internet protocol-based communication, installing sensors on the grounds, and expensive cameras [9,10], which makes it an expensive affair. The proposed solution does not use ground sensors; it uses a standard CPU. Furthermore, it requires reduced network-based communication overheads to process the image of the parking space. Most studies have focused mainly on either LP detection or parking occupancy detection, and thus lack crucial information regarding the complete solution of LP-based smart parking, particularly a solution obtained using machine learning (ML) methodologies. The proposed solution allocates parking spaces according to a vehicle's LP and preference. The proposed method uses a combination of an application-oriented approach and deep learning neural network techniques to detect and recognize LPs and to detect parking spaces, respectively.

The rest of the study is organized as follows. In Section 2, the latest developments in LP and parking occupancy detection are presented. The proposed solution is demonstrated in Section 3. In Section 4, an experimental study that reveals the advantages of using our proposed solution is presented. Then, it is compared with other state-of the-art solutions. In Section 5, we present the conclusions of the study.

2 State-of-the-Art

LP recognition and detection systems are developed in phases, such as binarization, character analysis, plate edges, deskew, character segmentation, optical character reader (OCR), and post-processing. These phases are developed using programming applications. However, ML can also be used to develop this system, which can improve the accuracy of the system because of the training process involved in ML. Many ML solutions exhibit considerably enhanced performance because of the availability of large-scale annotated datasets. However, such solutions are intended for real-time traffic management systems. The proposed solution is not intended for high traffic; it can be used in cities or in large organization where parking spaces can be identified using the application development approach. Moreover, deep CNN can be used for parking and identification of vehicles. A study proposed reservation-based parking, wherein various sensors are used to update the parking status dynamically, and the parking fee varies according to the driver's selection of a parking space, thus affecting the parking status [11].

In another study, the authors developed a smart parking system that uses the Internet of Things methodology to monitor and identify a single parking space [12]. They developed a mobile application that allows the driver to check the occupancy of a specific parking space and reserve it accordingly.

Geng et al. [13] proposed a mixed integer linear program for smart parking in an urban environment. In this approach, the parking system can consider a user's requirements along with parking costs and ensure that parking capacity is utilized efficiently.

A study proposed a smart parking method that analyzes the radio wave intensity of the driver's smartphone by positioning beacon equipment on the parking grounds to identify the availability of a parking space [14].

Numerous studies have provided solutions to detect and recognize LPs [15–18], using various technologies in an uncontrolled environment. A study conducted by Hsu et al. [15] used an application-based approach to recognize LP numbers; further, they categorized the vehicle LP into three parts, which provided a solution for plate detection and recommended a solution using technical parameters.

A three-layer framework-based Bayesian hierarchical framework was proposed to detect a parking space. The parking space inference depends on the images that may have varied luminance, shadow effects, distortions, and occlusions. Hence, they suggested various methods to overcome these problems concerning a parking space [19]. Complementing the aforementioned study, Tschentscher et al. [20] evaluated their proposed solution using three independent datasets and proved that a video-based and cost-effective occupancy detection method in the parking lot is advantageous over other related solutions.

In this study, we used the PKLot dataset, which was developed by De Almeida et al. [8], they captured this dataset under various weather conditions, such as sunny, rainy, and overcast periods. Hence, this dataset is suitable for training and testing of any proposed solution for parking detection and allocation. In this study, to detect a suitable parking lot, local binary patterns (LBPs) and local phase quantization, along with a machine classifier were applied to the dataset. We used a support vector machine classifier for both training and testing. The performance of the proposed solution was excellent, with a recognition rate >99%.

Amato et al. [9] recommended a deep CNN-based decentralized and effective solution to detect parking spaces. They compared their proposed solution by using PKLot, their dataset (in which nine smart cameras captured the "CNRPark-EXT dataset with a different point of view and different perspectives on different days with different weather and light conditions"), and well-known AlexNet proposals; and the performance of their proposed solution was comparable with AlexNet.

Two studies [10,21] have used recurrent neural networks along with long short-term memory and segmentation-free approaches to train and to identify the sequential features of an LP. In the study by Li et al. [21], contextual information was extracted and the errors caused, particularly because of the segmentation-free approach, were removed. Alternatively, Kessentini et al. [10] used a semi-automatic labeling approach that does not remove errors but reduces time and cost of manual labor required for labeling.

Saudi Arabian LPs consist of two sections: Arabic and English characters, as shown in Fig. 3. In this study, we considered the English section only because it is the transliteration of the Arabic characters. The proposed application captures an image of the vehicle's LP, crops it, and passes it to the CNN module for character recognition.

3 Proposal

The parking area of the university (sample area) is spread across 100 acres and is used by students and faculty members. We placed two cameras to capture images of the parking lot and the entrance. The proposed method consists of two parts: LP detection and recognition module and parking space detection and selecting the free space. We appended the captured images in our database for future utilization. Fig. 1 shows the flow chart of our proposed method.

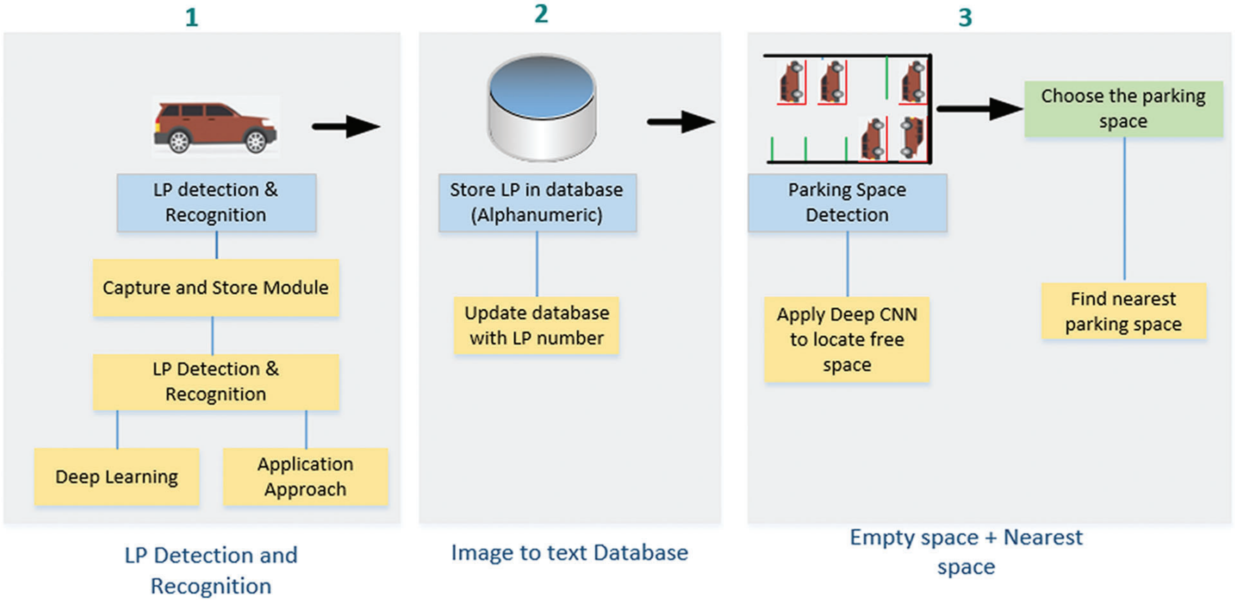
1
2
3
LP detection & Recognition
Capture and Store Module
LP Detection & Recognition
Deep Learning
Application Approach
Store LP in database (Alphanumeric)
Update database with LP number
Parking Space Detection
Apply Deep CNN to locate free space
Choose the parking space
Find nearest parking space
LP Detection and Recognition
Image to text Database
Empty space + Nearest space

Figure 1: Flow of the proposed smart parking solution

3.1 Capture and Store Module

The capture and store module, as shown in Fig. 2, enables capturing the image of a vehicle that enters the university. The captured image focuses on the LP number. The capture module was developed using Microsoft's DirectX end-user that has a run-time library. It is a built-in application programming interface (API) that allows capturing images using a camera. The captured image is stored in a container. We used the Lightning Memory-Mapped Database because it is quick and uses a key-value store method, unlike a relational database. Furthermore, it ensures efficient storage and access. Because it returns direct pointers to the stored keys and values through its API, and thus prevents direct access to the database, it considerably reduces expensive processing of the CPU.

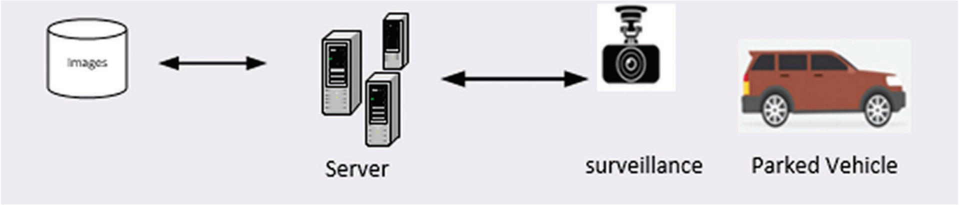

Figure 2: Capture and store module

Fig. 3 shows the capture module; each module was developed using an API of DirectX. Source filter, as the name suggests, uses input from the source camera and then passes it downstream for further processing. The source filter does not understand anything about the received data. However, it has neighbor filter called the parser filter that understands the format and locates the starting point of the elementary stream that wants to be exposed; then, the parser filter creates an output pin to render the parsed feed to the transform filter, which processes the received media and eventually creates an output stream. A video of the received media is displayed on a screen and also saved to a database. The store module stores the media, which are then transmitted for further investigation. Renderer filters generally accept media, play it on a display, and store it on an external device.

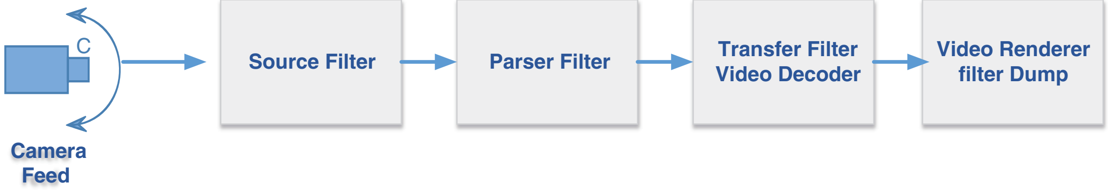

Figure 3: Graph-edit of capture module from hard drive or camera

3.2 Our Approach

There are a plethora of studies that provide solutions for LP number detection and recognition. The proposed module provides solutions for both front and rear ends of a vehicle; therefore, it uses application- and CNN-based approaches to detect and recognize LPs, respectively. Fig. 4 shows our approach used for LP detection and recognition that is employed as an open-source automatic license plate recognition (ALPR) module, which employs the LBP protocol. LP detection and identification for university parking does not require an ML-based solution because of the less intensity of traffic compared with highway and city traffic. Hence, we used an open-source-based ALPR system. Fig. 5 shows the placement of the camera used to capture an LP-only image. For such a system, the placement of the camera is crucial because capturing an image of the entire vehicle will require additional time to identify the LP, and eventually, adversely affect its performance and result in a decreased confidence value. We applied the Pythagorean Theorem to identify the missing sides of the right-angled triangle shown in Fig. 5. The two sides form a 90° angle, and thus, the hypotenuse is the longest side of the triangle. Such a right-angled triangle is called a 30°–60°–90° triangle. Hence, trigonometric functions can be applied to determine the lengths of the missing sides.

Tangent $\alpha = \frac{side\ A}{side\ B}$ (According to the proposed approach, angle A is 60°, angle B is 90°, and angle C is 30°). The distance between sides of the right-angled triangle can be determined using the following formulas:

$$\tan \mathrm{A} = \frac{BC}{AB} \text{ and } \tan \mathrm{C} = \frac{AB}{BC} \tag{1}$$

The proposed approach had eight phases; each phase was further improved using a GPU. However, CPU-based performance is suitable for real-time applications, wherein decisions can be made promptly and instantaneously for the next course of action. These phases are called the detection phase, binarization, char analysis, plate edges, deskew, character segmentation, OCR, and post-processing. The detection phase uses the LBP algorithm to detect an LP; this process can be managed by the GPU or the CPU. In binarization, a binary image is created, which means each pixel represents either 0 or 1. However, in this study, we created multiple binary images of each LP to identify all possible characters. Additionally, we separated the text from the background to remove noise, which occurs in the form of lines and unwanted objects. In the character segmentation phase, LP characters were segmented using a vertical histogram to identify space between them and character boxes; disqualifying character spaces that sync with the rest of the character set of the LP were removed.

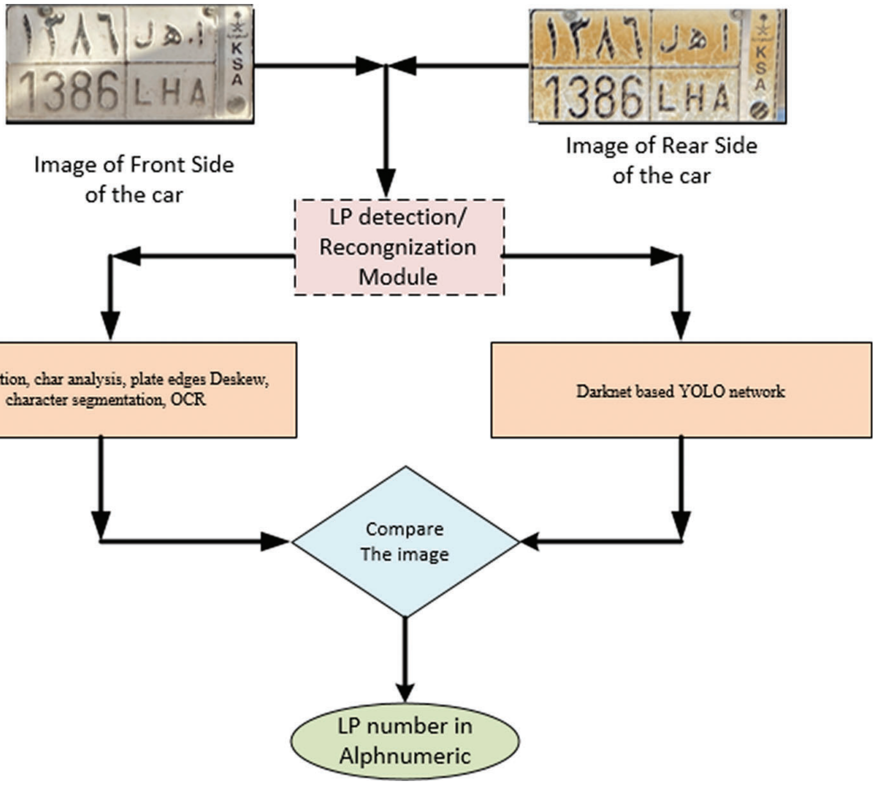
1386 LHA
KSA
Image of Front Side of the car
1386 LHA
KSA
Image of Rear Side of the car
LP detection/ Recongnization Module
Binarization, char analysis, plate edges Deskew, character segmentation, OCR
Darknet based YOLO network
Compare The image
LP number in Alphnumeric

Figure 4: Illustration of the proposed LP detection and recognition system based on Automatic License Plate Detector

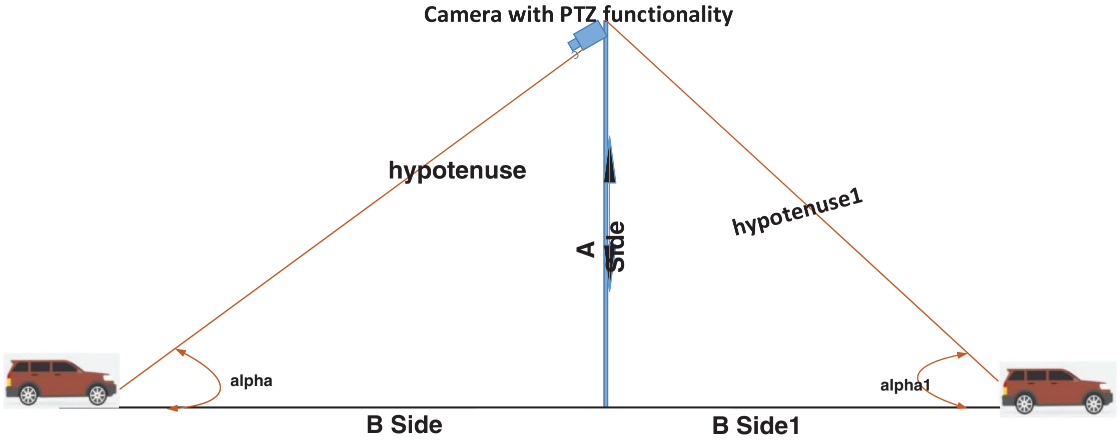

Figure 5: Mounting of the camera at the main entrance of the university

3.3 Calculation of LBP Protocol

To compute each pixel using the LBP algorithm, the grayscale value of an LP was compared with its adjacent pixels that constitute the LP pattern. A pixel evaluation between the center pixel and its adjacent pixels using the following equation yielded either a 1 or 0, which indicated whether the midpoint pixel has a high grayscale value and created a bit value.

$$\sum\nolimits_{p=0}^{p-1} s\left(g_p - g_c\right)2^p \tag{2}$$

where, g_c represents the grayscale value of the center pixel, and gp represents the adjacent pixel that ranges between 0 and P − 1. A histogram of the image is defined as follows:

$$\sum\nolimits_{x,y} I\{f_l(x,y) = i\},\ i = 0,\ldots..\ n-1, \tag{3}$$

Each pixel can be identified and mathematically described as follows:

$$s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 1 \end{cases} \tag{4}$$

Thus, it categorizes an image into regions.

$$\mathrm{H_{i,j}} = \sum\nolimits_{\mathrm{x,y}} \mathrm{I\{f_l(x,y) = i\}\ I\left\{(x,y) \in R_j\right\},\ i = 0,\ldots,n-1,\ j = 0,\ldots,\ m-1} \tag{5}$$

The aforementioned histogram presents information regarding the LP's different locations and regarding the pixel pattern. The labels are gathered over a tiny region to yield a regional-level description, and eventually, these regional histograms are used to reconstruct the LP. The final calculation is performed by summing i and j. Likelihood static ratio $L(S,M) = \ -\sum\limits_i S_i\ logM_i$. The likelihood ratio was used to reveal that some pixels may provide additional information; hence, we assigned a weight for each region based on the edge of the plate, where wj is the weight of the region j.

$$(S,\ M) = \ \sum_{i,j} w_j \frac{\left(s_{i,j} - m_{i,j}\right)}{\left(s_{i,j} + m_{i,j}\right)} \tag{6}$$

3.4 Deep Learning for Rear-End LP

To detect and recognize an LP at the rear end of the vehicle, we used a darknet framework with 21 architectural layers, 15 convolutional layers, and 4 max-pooling layers. We used a dataset quite similar to the test dataset because of which adjusting the entire network is unfeasible. Some layers of darknet were replaced with a linear classifier because they required excessive CPU processing. However, few more convolutional layers were acquainted using the remaining max-pooling layers. These newly introduced convolutional layers adjusted with the existing training data occupied as a more cutting-edge non-linear classifier. Numerous people from different countries have suggested numerous solutions to detect an LP because of the unique nature of the problem. These solutions use edge-, color-, texture-, and character-based approaches; however, these approaches have disadvantages in the various technical aspects of image processing, particularly in the detection of an LP. Advantages of these solutions were examined, and they were incorporated in solutions such as deep CNN networks. You Only Look Once (YOLO) is a promising solution proposed for real-time image detection. A YOLO solution requires only 22 ms to process one image, which is approximately seven times faster than Faster RCNN. Different versions of YOLO have been released; version 3 of YOLO is smarter and faster than versions 1 and 2. Moreover, the classifier network of its initial version was programed to receive an image of 224 × 224 resolution and intensified to 448 × 448 resolution for detection. Eventually, the aim is to adjust filters to improve mapping score. The aforementioned methods are suitable for heavy traffic maintenance and crime-related problems. An LP recognition system architecture with a modified YOLOv3 architecture and numerous modified filters was implemented in this study to detect the LP (rear end of the vehicle), as shown in Fig. 5. The system receives input from the camera and passes it to the CNN network to produce vector bounding boxes and class prediction. Tiny YOLO divides the image into S × S grid cells; in this approach, each cell is responsible for predicting an image within its vicinity, that is, cell value. Each cell predicts bounding boxes labeled as B and the confidence score for each of these boxes. Hence, this score indicates the confidence score and accuracy rate. Furthermore, each bounding box represents confidence scores and prediction, such as x, y, w, and h; x, y represents the center of the box and indicates the bounds of the grid cell; and w, h represents the width and height of the image. The confidence level of an image represents the intersection over the union between the predicted box and any ground truth box.

The proposed CNN YOLO module yields a confidence score for its predictions for each captured LP image bounding box. We considered this confidence score as a threshold value to arrive at the final prediction. Higher the value of the confidence score, higher the probability of achieving perfect detection, and vice versa. However, the placement of the camera and condition of the university's entry point were such that the vehicle will slow down, thus making it easy for the camera to capture the LP image of the vehicle. Hence, this arrangement indicates that the confidence value does not affect the prediction value. By setting up the camera as shown in Fig. 4, we trained the darknet YOLO framework similar to the suggested network (Fig. 6) on the LP regions. We used this suggested network over the full LP-only image without any pre-processing. Hence, the LP image received from the camera was cropped. The training data consisted of an annotated dataset generated using the aforementioned camera setup. Each LP plate was annotated using the bounding boxes for each of its elements. The following section describes the algorithm devised for automatic annotation of LP elements. LP alphanumeric characters were annotated for the training sets. We used an open-source Python-based software Labellmg, which is a graphical image annotation tool that labels object bounding boxes in images such as in an LP image. To reduce the time required to compute a considerable number of LP datasets, we modified the Labllmg software to annotate a set of LP images. As shown in Algorithm 1, the dataset of the LP images is the input and the annotated images are the output. To train the proposed architecture, an iterative process assisted the algorithm to recognize the different features of the LP components. For training, we used the algorithm suggested by Kessentini et al. that trains the labeled LP image automatically; however, we used automated annotated LP images as input to Algorithm 1.

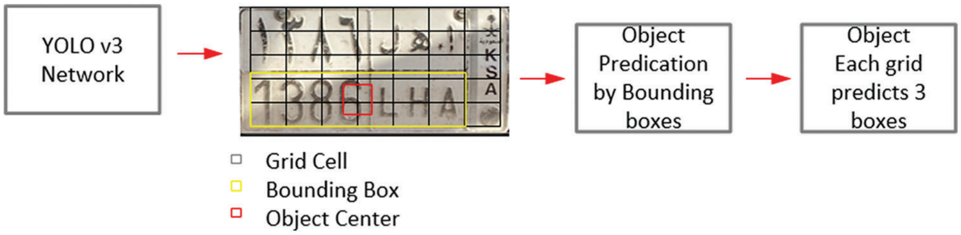
YOLO v3 Network
Object Predication by Bounding boxes
Object Each grid predicts 3 boxes
□ Grid Cell
□ Bounding Box
□ Object Center

Figure 6: LP recognition using YOLO network

To generate automatic annotation, we passed a set of images (Lbells_Img) as input parameters at the LP component level and used Seq_N_Lbells without annotated LP images and their corresponding label sequences. LP character-level images were obtained as outputs, which were automatically annotated. We trained YOLOv3 using our dataset on the network model M0. In each iteration of loop_varraible, we constructed a set of images Seq_N_Lbells, which are identified by our network model denoted by $M_{loop_{varraible}-1}$. Hence, the YOLOv3 architecture was updated on LPimage loop_varraible to receive the network model $M_{loop_{varraible}}$. This process continued till all images of Seq_N_Lbells were labeled. We executed the aforementioned algorithm on our in-house dataset PSAUVdata, which consisted of 6123 color images of vehicles captured under different weather conditions from the live stream camera placed, as shown in Fig. 5, at the university entrance. These images were labeled according to the characters in Saudi Arabian LP plates. The proposed method annotated these images correctly using <10 iterations. This resulted in a 96.8% annotated dataset in <10 iterations with 600 thoroughly annotated images. We attained 89 intersection over union for the PSAUV data dataset, which enabled us to further this experiment as mentioned following sections.

3.5 Image Database Management

Each image was stored in the database along with an extended markup language (XML)-supported file and included the coordinates of all the parking spaces and their labels (occupied/vacant), which could be retrieved whenever needed because they were stored in the database. The LP number once extracted from a parked vehicle was stored as data. As we can store a data image and its other attributes, such as the name of the vehicle owner, coordinates of the vehicle location, and Boolean value, which indicates whether a vehicle is appropriately parked (stipulated place). This management module can manage and retrieve the information of the owner of a vehicle. Initially, it contained only the vehicle number, and the parking spaced was allocated according to the classroom location.

4 Parking Detection Module and Selection of Parking Space

Using the proposed approach, a suitable parking location is allocated according to the classroom session, and an image is captured using a high-end smart camera. Once the vehicle enters the university, the high-end camera captures its image, processes the image, and displays available parking spaces on the screen; the driver can drive through the route mentioned on the display. The route management module selects the parking location according to the nearest neighbor algorithm. Many efficient nearest neighbor algorithms have been proposed; however, they require high-processing equipment to process 60 million parameters and 500,000 neurons and require five convolutional layers arranged using a deep CNN. Such high-processing calculations require high-end processing systems, where a GPU and CPU can perform the processing. Hence, for cost-effectiveness, AlexNet required modification. Moreover, AlexNet could differentiate 1.2 million high-resolution images during the ImageNet LSVRC-2010 competition because of its capability to use GPU and CPU efficiently to non-saturate neurons for a convolution operation. Hence, image processing using GPU and CPU is not required for campus-based parking space detection. Fig. 7 shows a scaled version of AlexNet to suit the minimum requirements of vehicle parking space detection. In the proposed method, we defined convolution, pooling, and local response normalization (LRN), except dropout, and fully connect in Python functions with TensorFlow.

Algorithm 1: Automatic annotation of LP components using LeblImg

```
Input: A list of cropped LP images $Lbells\_Img = \{img_1, \ldots img_N\}$ $a\ list\ of\ LP\ annoted\ images$
A subgroup images $Seq\_N\_Lbells = \{img_1, \ldots img_M\}$, and their respective labels
Lbels = $\{lbels_1, \ldots lbles_M\}$ none of them annotated.
Output: Labeled LP images $LPimage = \{l_0, \ldots l_k\}$, $it\ is\ a\ set\ of\ image\ with\ respective\ labled\ elements.$
1. Train YOLOv3 network on images $i\ to\ acquire\ trained\ model\ m_0$
2. Initialize the iteration variable $loop\_varraible = 1$
3. $LPimage = \varphi$
4. repeat
5. $LPimage_{loop\_varraible} = \varphi$
```

(Continued)

Algorithm 1 (continued).

```
for each images $Img$ in $Seq\_N\_Lbells$
    Test $Img$ with YOLOv3 using $M_{loop_{varraible}-1}$ to detect labels
    Concat LP labels with sequence label
    If (LabelImg $(LPimage)$ == Lbells) then
        $LPimage_{\ loop\_varraible} \leftarrow LPimage\ loop\_varraible\ \cup \{LPimage\}$
        $LPimage\ \ \leftarrow LPimage \backslash \{LPimage\}$
    end if
end for
$LPimage \leftarrow LPimage\ \cup Lbells\_Img$
Fine tuning of YOLO on li to obtain M $loop\_varraible$ models
Increment $loop\_varraible$
Until $LPimage == \varphi$ and $loop\_varraible\ == \varphi$
```

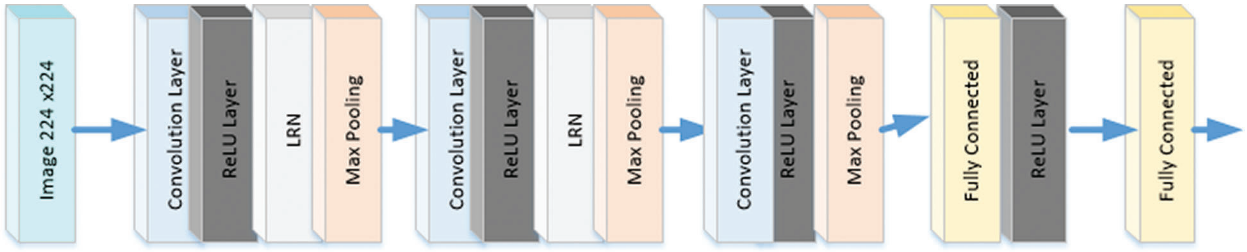

Figure 7: mmAlexNet, 11 × 11, and a stride of 4 as the first convolution layer, max-pooling has, two as stride with three as to size and fully connected as a smooth-max way

According to mAlexNet [9], the proposed method can be explained as follows. A 224 × 224 RGB image is input to AlexNet, which passes the image through the convolutional layers, with initial settings of size 11 and strides 4 feature maps. ReLU (linear rectification), LRN, and max-pooling used size 2 and strides 3. Next, a second convolutional layer of size 5 × 5 and a maximum pooling layer of size 3 × 3 was observed. A maximum pooling layer trails the three convolutional layers with a filter of size 3 × 3, astride. The Output Layer, unlike AlexNet, has a SoftMax output layer with two possible values. Once parking occupancy is detected, the next step is to identify a free parking space according to the classroom location, as the classroom locations are fixed according to the department-wise timetables. Hence, latitudinal and longitudinal values of the classroom location were disseminated to the server in a timely manner according to the time, subject, and faculty information stored in the database. Each parking image file had XML file descriptions about the GPS location and status of the space (occupied or vacant). The XML file was used to label the images according to their location coordinates and to schedule a parking area. Fig. 8 shows a flowchart used to identify a parking space close to the classroom location.

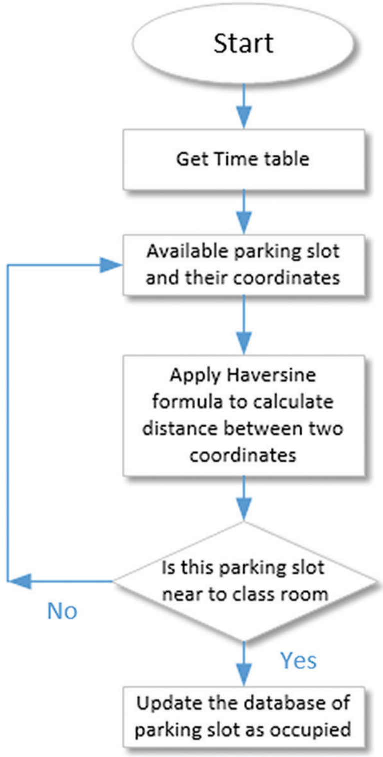
Start
Get Time table
Available parking slot and their coordinates
Apply Haversine formula to calculate distance between two coordinates
Is this parking slot near to class room
No
Yes
Update the database of parking slot as occupied

Figure 8: Flow to detect the shortest location of the parking area from the classroom

Apply the Haversine formula to calculate the distance between the latitude and longitude coordinates considering r as the radius of the earth, we obtain the following equation:

$$d(\varphi_1\gamma_1\varphi_2\gamma_2) = 2r\ arcsin\left(\sqrt{\left(sin^2\ \frac{\varphi_2 - \varphi_1}{2}\right) + cos(\varphi_1)cos(\varphi_2)\left(sin^2\ \frac{\gamma_2 - \gamma_1}{2}\right)}\right) \tag{7}$$

In this study, we used two independent modules that work on two different aspects of image processing, that is, to detect and recognize an LP and to identify a suitable parking space. Hence, we used two different datasets for two different purposes. PSAU-Vehicle entrance, in short, PSAUVdata, consisted of vehicle images gathered from the university entrance. There were 6123 color images captured from the live stream camera placed according to Fig. 4, which were used to detect and recognize LPs. The second dataset named CNRPark-EXT was a collection of vehicle images with approximately 150,000 available and engaged parking lots. This dataset was collected in 2015 using two cameras and has been utilized by many researchers for different purposes, including parking occupancy detection. However, until now, this dataset had not been used for location-based parking occupancy detection. Initially, CNRPark appended the released dataset with images collected under various weather conditions and called it CNRPark-EXT. It was categorized into training, validation, and test groups to be used in training and testing classification algorithms. The dataset had 144965 labeled parking space patches.

In addition to CNRPark, PKLot, which encompasses 12,417 images obtained from two different parking lots during different weather conditions, was used in this study. The dataset was further categorized into three categories that had further subcategories for different seasons and was arranged according to acquisition data. We downloaded the CNRPark-EXT dataset from the website, PKLot. Tab. 1 presents the various datasets.

Table 1: Datasets from various sources

Dataset and their subsets	Free space	Occupied	Total space
UFPRO4-Training	31,956	24,554	56,510
UFRR04-Testing	27,764	21,571	49,335
UFPRO5-Training	36,275	46,994	83,269
UFPRO5-Testing	32,084	50,432	82,516
PUCPR-Training	118,689	93,804	212,493
PUCPR-Testing	111,351	100,425	211,776
CNRParkOddEven	4,181	8,403	12,584
CNRPark-EXT-Training	46,877	47,616	94,493
CNRPark-EXT-Testing	13,549	18,276	31,825

5 Evaluation

The evaluation of the proposed method was based on two categories because two significant sections were used, namely LP detection and recognition and parking space detection system. Hence, two different datasets were used for the proposed method. Initially, we evaluated the proposed method for LP detection and recognition protocol by using PSAUVdata collected over a period of >6 months under various weather conditions, such as severe heat and sandstorm, excluding the rainy season because of its less intensity. We placed two cameras at the entrance that focused on the front and rear LPs of any vehicle.

5.1 LP Detection and Recognition

We used intersection over union (IoU) to evaluate the accuracy of an LP detector on our in-house dataset, which was formalized as follows:

$$IoU = \frac{A \cap B}{A \cup B} \tag{8}$$

The prediction was correct only if IoU ≥ 0.5, where A and B are the prediction and ground truth bounding boxes. Furthermore, precision and recall metrics were employed to measure the accuracy of the LP detectors. Precision is the percentage of correct predictions, and recall is the number of true positives divided by the sum of the number of true positives and false negatives, and they are calculated as follows:

$$recall \ = \frac{TP}{(TP + FN)} \tag{9}$$

$$precision \ = \frac{TP}{(TP + FP)} \tag{10}$$

where, ***TP*** is true positive, and ***FP*** and ***FN*** are false positive and false negative, respectively.

Simultaneously, we ran an experiment where the IoU values ranged from 0.4 to 0.8; our approach produced a significant result. Tabs. 2 and 3 show the results of the two distinct datasets used for recall and precision evaluation metrics. The result of LP detection was recorded only when the overlap between ground and truth bounding boxes was higher than the IoU value.

Table 2: Evaluation of LP detection concerning evaluation metric precision and recall rates (IoU ≥ 0.5)

Dataset	Recall	Precession
WadiPSAUVdata	98%	97%
AlkharjSAUVdata	97%	96%

Table 3: Evaluation of LP detection with respect to evaluation metric recall rates under various IoU values (0.3, 0.4, 0.5, 0.6, 0.7, 0.8, etc.)

Dataset	Different IoU values					
	0.3	0.4	0.5	0.6	0.7	0.8
WadiPSAUVdata	98.26%	97.67%	97.22%	96.23%	95.99%	69%
AlkharjSAUVdata	98.56%	97.54%	97.19%	96.19%	95.18%	68%

Although the proposed system was executed under different weather conditions, the results indicate that the proposed system performed extremely well (Tab. 4) and demonstrated a high confidence level even under different seasons within KSA. The confidence level of the proposed algorithm varied during sandstorms; however, it was still >80%; hence, the proposed detection method is considered accurate. According to the proposed method, we captured images of both rear- and front-end LPs in sandstorm weather; results of LP detection sometimes front or rear LP detection has resulted lower than 60%; however, when we compared the resultant text LP numbers, they were accurate. Tab. 5, shows comparison report, we executed the proposed method on different datasets (e.g., Fifty States car LPs dataset) that host LP images of different US regions. The confidence levels were extremely high.

Table 4: Results under different weather conditions

Dataset	Seasons		
	Winter	Sandstorm	Hot
WadiPSAUVdata	98%	80%	99%
AlkharjSAUVdata	97%	85%	99%

Table 5: Comparison report on different datasets

Dataset	Recall	Precession
WadiPSAUVdata	98%	99%
AlkharjSAUVdata	97%	99%
Reference [21]	96.8	92.6
Reference [22]	98.38	98.53
Reference [20]	99.82	100

5.2 Parking Space Detection

During the second part of the evaluation, the parking space detection method used CNRPark and PKLot datasets and our dataset. Tab. 1 presents a dataset for training and testing, downloaded from their respective websites. We used the following state-of-the-art datasets [8,9]:

We applied two evaluation metrics commonly used by most state-of-the-art research proposals to evaluate their performances, namely area under the curve (AUC) and receiver operating characteristics (ROC). ROC is a graph that shows the false positive rate on the x-axis and true positive rate on the y-axis for various threshold values. We gathered many different datasets to evaluate the proposal. This approach enabled us to test the efficiency and compare it with other approaches. These experiments assessed the appropriateness of detecting the presence of a vehicle in a parking space and compared this obtained result with other proposals. Furthermore, the features extracted for training were from the UFPR04, UFPR05, or PUCPR and CNRPark-EXT datasets and were assessed using the corresponding testing sets. We considered a single parking space for our experiment unlike Paulo R. L. de Almeida et al. The comparison report shown in Tab. 6 reveals that the proposed method is comparable with top performers of LBP and mAlexNet variants. Thus, we can conclude that the performance of the proposed solution was superior to that of other state-of-the-art solutions. Different subsets yielded different results for different proposals.

Table 6: Comparison report with others proposed methods

Proposals along with train on data sets	Sub dataset test set	Accuracy In %	AUC
CNRParkEven			
mmAlexner	CNRParkOdd	90.69	0.92
mAlexNet*	CNRParkOdd	90.71	0.92
LBP**	CNRParkOdd	87.21	0.92
CNRParkOdd			
mmAlexner	CNRParkEven	90.13	0.94
mAlexNet*	CNRParkEven	90.07	0.94
LBPQ (variant LBP)**	CNRParkEven	87.65	0.95
UFPR04			
mmAlexner	UFPR04	99.54	0.99
mAlexNet*	UFPR04	99.54	0.99
LBQ (variant)**	UFPR04	99.55	0.99
UFPR05			
mmAlexner	UFPR05	93.71	0.98
mAlexNet*	UFPR05	93.69	0.98
LBQ (variant LBP)**	UFPR05	98.90	0.93
PUC			
mmAlexner	UFPR05	98.11	0.99
mAlexNet*	UFPR05	98.03	0.99
LBQ (variant LBPqd)**	UFPR05	87.15	0.94

Our approach, which is a modified version of mAlexNet (mAlexNet is modified version AlexNet), outperformed other methods considered in this study. Furthermore, mAlexNet has best performance, however, our proposed technique reported a higher accuracy than those reported in the Max Ensemble. Tab. 7 compares our proposed solution with AlexNet and mAlexNet. We conducted two sets of experiments, one for the training dataset and another for testing. We selected different datasets for each of these experiments. We used CNRPark and PKLot datasets for training and PKLot dataset for testing. This comparison allowed us to measure the accuracy and the AUC evaluation metric for the aforementioned techniques. AlexNet has a slight edge over mAlexNet among most datasets, and it outperformed the rest of the methods considered in this study. The proposed method outperformed LBP variants. We noted that under severe sandstorms, reading the LP becomes difficult because of lack of visibility; however, parking lot detection remains unaffected.

Table 7: Comparison report with others proposed methods

Proposals	Sub dataset test set	Accuracy In %	AUC
CNRPark			
mmAlexner	PKLot Test	95.01	0.97
mAlexNet*	PKLot Test	95.28	0.99
AlexNet*	PKLot Test	95.60	0.99
Train on PKLot			
TRAIN	PKLot Test	97.90	0.96
mmAlexner	PKLot Test	98.07	0.99
mAlexNet*	PKLot Test	98.81	0.99
AlexNet*			
LBP**	PKLot Test	65.31	0.63

6 Conclusion

In this study, we presented a smart parking system, particularly for a university; however, this study can also be utilized for smart cities and highway traffic for LP detection and recognition and parking lot detection based on the driver’s priority. This study contributes to the following aspects: LP detection and recognition, space detection, and allocation of a suitable space according to the classroom location. Each module considered different approaches and scope. To establish this characterization, we used and acknowledged substantial image samples collected from various sources. This study used the application-oriented approach, which provided different modules for plate detection, segmentation, and recognition along with efficiently capturing and verifying the modules. Our experimental study concludes that smart placement of cameras and other variables can yield superior results compared with other solutions. Second, the rest of the two modules were combined to identify a suitable parking. In this module, we modified the popular AlexNet algorithm to suit our university settings. We used deep learning techniques to construct the required framework to detect a suitable parking space and then applied the Haversine formula to identify the parking space nearest to the classroom. The proposed solution was compared with state-of-the-art solutions. The image database contained in the initial version of our solution will be made available to the research community after receiving permission from the higher management of the university.

Acknowledgement: The authors appreciate the Prince Sattam Bin Abdulaziz University, Saudi Arabia, for providing research resources and equipment and various research programs to encourage research among faculty members.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] C. Arth, F. Limberger and H. Bischof, “Real-time license plate recognition on an embedded DSP-platform,” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, MN, USA, pp. 1–8, 2017.

[2] C. N. Anagnostopoulos, “License plate recognition: A brief tutorial,” *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 1, pp. 59–67, 2014.

[3] L. Hu and Q. Ni, “IoT-driven automated object detection algorithm for urban surveillance systems in smart cities,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 747–754, 2018.

[4] B. Shi, X. Bai and C. Yao, “An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 11, pp. 2298–2304, 2017.

[5] N. Bellas, S. M. Chai, M. Dwyer and D. Linzmeier, “FPGA implementation of a license plate recognition SoC using automatically generated streaming accelerators,” in *Proc. IEEE International Parallel & Distributed Processing Sym.*, Rhodes Island, Greek, pp. 1–8, 2006.

[6] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini and E. Magli, “Robust license plate recognition using neural networks trained on synthetic images,” *Pattern Recognition*, vol. 93, no. 1, pp. 134–146, 2019.

[7] V. Paidi, H. Fleyeh and R. G. Nyberg, “Deep learning-based vehicle occupancy detection in an open parking lot using thermal camera,” *IET Intelligent Transport Systems*, vol. 14, no. 10, pp. 1295–1302, 2020.

[8] P. R. de Almeida, L. S. Oliveira, A. S. Britto Jr, E. J. Silva Jr, A. L. Koerich *et al.,* “PKLot-A robust dataset for parking lot classification,” *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, 2015.

[9] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini *et al.,* “Deep learning for decentralized parking lot occupancy detection,” *Expert Systems with Applications*, vol. 72, no. 1, pp. 327–334, 2017.

[10] Y. Kessentini, M. D. Besbes, S. Ammar and A. Chabbouh, “A two-stage deep neural network for multi-norm license plate detection and recognition,” *Expert Systems with Applications*, vol. 136, no. 1, pp. 159–170, 2019.

[11] H. Wang and W. He, “A reservation-based smart parking system,” in *Proc IEEE Conf. on Computer Communications Workshops*, Shanghai, China, pp. 690–695, 2011.

[12] A. Khanna and R. Anand, “IoT based smart parking system,” in *Proc. IOTA*, Pune, India, pp. 266–270, 2016.

[13] Y. Geng and C. G. Cassandras, “A new “smart parking” system infrastructure and implementation,” *Procedia Social and Behavioral Sciences*, vol. 54, no. 1, pp. 1278–1287, 2012.

[14] T. Ebuchi and H. Yamamoto, “Vehicle/Pedestrian localization system using multiple radio beacons and machine learning for smart parking,” in *Proc ICAIIC*, Okinawa, Japan, pp. 86–91, 2019.

[15] G. S. Hsu, J. C. Chen and Y. Z. Chung, “Application-oriented license plate recognition,” *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, 2013.

[16] S. Draghici, “A neural network based artificial vision system for licence plate recognition,” *International Journal of Neural Systems*, vol. 08, no. 01, pp. 113–126, 2011.

[17] M. Y. Arafat, A. S. M. Khairuddin, U. Khairuddin and R. Paramesran, “Systematic review on vehicular licence plate recognition framework in intelligent transport systems,” *IET Intelligent Transport Systems*, vol. 13, no. 5, pp. 745–755, 2019.

[18] M. R. Asif, Q. Chun, S. Hussain and M. S. Fareed, “Multiple licence plate detection for Chinese vehicles in dense traffic scenarios,” *IET Intelligent Transport Systems*, vol. 10, no. 8, pp. 535–544, 2016.

[19] C. C. Huang and S. J. Wang, “A hierarchical bayesian generation framework for vacant parking space detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 20, no. 12, pp. 1770–1785, 2010.

[20] M. Tschentscher, C. Koch, M. König, J. Salmen and M. Schlipsing, “Scalable real-time parking lot classification: An evaluation of image features and supervised learning algorithms,” in *Proc. IJCNN*, Killarney, Ireland, pp. 1–8, 2015.

[21] H. Li, P. Wang, M. You and C. Shen, “Reading car license plates using deep neural networks,” *Image and Vision Computing*, vol. 72, no. 1, pp. 14–23, 2018.

[22] Z. Selmi, M. B. Halima and A. M. Alimi, “Deep learning system for automatic license plate detection and recognition,” in *Proc. ICDA*, vol. 1, Kyoto, Japan, pp. 1132–1138, 2017.