ORIGINAL PAPER

Street-Based Parking Lot Detection With Image Processing And Deep Learning

Ahmet Sayar[1] · Ahmet Fatih Mustacoglu[2]

Abstract

Due to the rapidly increasing number of vehicles and urbanization, the use of parking spaces on the streets has increased significantly. Many studies have been carried out on the determination of parking spaces by using the lines in the parking areas. However, the usage areas of this method are very limited since these lines are not found in every parking area. In this research, a unique study has been presented to determine the empty and occupied parking spaces in the parking area by processing the images from the cameras located at high points on the streets with depth calculation, perspective transformation and certain image processing techniques within the framework of specific features. Empty and full parking lots were determined by utilizing perspective transformation and depth measurement techniques, and the data obtained were transferred to the Real-Time Database environment. In addition to determining the parking spaces, the study also aims to inform users through the mobile application and to prevent traffic congestion, extra fuel consumption, waste of time and air pollution caused by fuel consumption.

Keywords Image processing · Deep learning · Vehicle detection · Smart parking systems · Depth analysis

1 Introduction

With the advancement of technology, the production cost of cars has decreased and the use of individual vehicles has increased. With the rapid increase in the number of vehicles in traffic in recent years, the problem of finding a parking space has started. In case there is no parking lot around the destination, parking lots on the side of the streets are more preferred than parking lots in terms of proximity. Over time, the problem of finding a parking space arose due to irregularly parked vehicles on the streets. Many studies have been developed over the years to solve this problem [1–3]. Among these studies, the separation of parking areas with lines and the use of parking meters were the most common methods. Due to

the costly installation of parking meters on each street, the parking areas in most streets today have become unattended and unsuitable for use. In a minority part, it was desired to prevent the parking problem by dividing the parking areas with the help of lines and determining the empty and full parking spaces with certain methods [4–6].

In Sect. 1, general information about the purpose, scope and methods of the study is given. In Sect. 2, the methods of finding empty and occupied parking spaces by using the commonly used basic image processing and deep learning methods are discussed, and the differences and improvements of our work with the studies in the literature are stated. Section 3 provides general information on the following subjects: determining the required parking space for the analysis of the parking area; segmentation and object detection algorithm to detect vehicles in the parking area; purpose and working principle of perspective transformation to increase the accuracy of depth analysis operations; real-world depth analysis of empty and occupied parking spaces according to the positions of the vehicles; and information about mobile and web applications developed for users interaction. The surveillance cameras are used in the related study causing to obtain a skewed view. So, the start and the end point of the vehicles are re-determined by the Pythagorean theorem.

✉ Ahmet Sayar
  ahmet.sayar@kocaeli.edu.tr

  Ahmet Fatih Mustacoglu
  ahmetfatihmustacoglu@topkapi.edu.tr

[1] Computer Engineering, Kocaeli University, Kocaeli, Turkey

[2] Computer Engineering, İstanbul Topkapı University, İstanbul, Turkey

After the diagonal finding process is completed all pixels are checked from the end point to the starting point explained in detail in this section as well. Finally, the experience and the knowledge gained from the relevant study are conveyed, the contribution of the study is given and the areas where it can be improved are indicated.

2 Related works

The related researches are briefly over-viewed through the section, and the detail information is given in Table 1;

There are many comprehensive articles on this subject, which many researchers are still working on [7, 8]. The methods used are quite similar to each other, and accuracy and improvement are generally emphasized.

Various image processing-based studies such as detection of vehicles, detection of road boundaries and traffic sign detection in smart traffic systems have been carried out. Yalcin and his colleagues [9, 10] overview the studies of road boundary detection and obstacle detection, in order to allow the movement of autonomous vehicles.

There are also studies on detecting vehicle warning signs in smart traffic systems related to Object detection [11]. Studies on vehicle recognition are more frequently encountered in the literature. Kul et al. summarized these studies in [12]. Also, Kul et al. proposed a vehicle detection system by querying the vehicle characteristics on the streaming video images with a distributed data system [13]. In addition, the performance of BGSLibrary on streaming video images was also evaluated [14].

One of the most important issues in measuring the distance between vehicles in two-dimensional images is depth measurement. Akın et al. [15] summarized the problems and difficulties that can be encountered in measuring depth in two-dimensional images in their study.

Algorithms used in parking lot detection using image processing and deep learning techniques are generally processed in two sub-categories. While the first algorithm processes the information received with the help of sensors in addition to the cameras on the vehicle, the other algorithm operates with the help of images from high-angle surveillance cameras without using any sensor data. An example of the work is given in [16].

Researchers from affiliated studies "The University of Melbourne" opted for 13,000 photographs and more than 700,000 tagged PKLot and Barry Street dataset for the project explained in [5].

Another remarkable study was carried out with the help of image processing techniques and a classification model explained in [6].

With a study conducted by Yeditepe University, a data set was created by collecting street images taken by road-side cameras. In this work, a Convolutional Neural Network (CNN) was created based on this dataset and the trained CNN model analyzes the street image to check if there is an empty parking lot [17].

Most of the studies mentioned and found in the literature were carried out in the parking areas that are located in the parking lots and separated by the lines [18, 19].

3 Architecture

As a preliminary stage, the parking area was determined on the image taken from the surveillance camera located on the street and the configuration file was created. The image taken from the surveillance camera is first processed with the instance segmentation model, then the masks of the objects and the frames surrounding the objects are obtained in the application. By applying perspective transformation to the vehicle masks obtained from the model, it has been determined whether the related vehicle mask is in the parking area that was figured out in the preliminary stage. The starting and ending points of the vehicles detected in the parking area were re-determined with the diagonal approach and updated. Next, the distances of the empty spaces between the vehicles were calculated according to the current start and end points obtained. Finally, the distance data obtained were presented to the interaction of the users via the mobile application.

3.1 Parking area camera configuration

As a result of the research, it was assumed that the surveillance cameras on the streets are stable at certain heights and angles. In the proposed work, operations were carried out on the live images taken from the Çayırova City Square surveillance camera in Kocaeli province. Since the parking areas on the streets can be in a different location for each camera, the parking area of interest to be observed should be determined initially with the configuration interface. While determining the parking area, dots are placed clockwise starting from the upper right corner of the desired area to the upper left corner (Fig. 1). Even if the selected points are asymmetrical, the interface connects these points. The purpose of this is the correct operation of the perspective transformation method used in the next stages.

3.2 Detection of vehicles in an image coming from a surveillance camera

First, the Surveillance camera link with the extension "m3u8" is opened with the VideoCapture function of the OpenCV library in order to process live images from the Çayırova City Square Surveillance. Since the frames read during the process will be processed one by one, the images taken from

Table 1 Summary of related works

References	Methodology	Results
[5]	A deep CNN and a binary Support Vector Machine (SVM) classifier are used	Detection accuracy of the occupancy of outdoor parking spaces are 99.7% and 96.7%
[6]	Color histograms and difference of Gaussian features, an SVM classifier, and exponential smoothing for temporal integration are used	An accuracy of 99.8% on classifying parking situations reached
[7]	This work presents the prototype of IoT-based Real-Time smart street Parking System (IoT-based RTSSPS) with accessibility of data to make it simpler for residents and drivers to locate a free parking slot at the streets	The prototype design of IoT-based RTSSPS with architecture and algorithm was presented
[8]	Two detection methods using a laser range sensor finder and a line-scan camera	A detection rate reached 76.9% and 90% respectively
[9]	The approaches in detecting road boundaries and obstacles using LIDAR sensors and the advantages of the approaches have been evaluated	The current performances and reliabilities of the algorithms need to be further improved
[10]	A lidar sensor was mounted on the vehicle with a pitch angle to detect road boundaries and obstacles for robust urban navigation of unmanned ground vehicles (UGVs)	The proposed algorithm was tested on different road platforms using an autonomous vehicle called as AKAY01
[11]	A ZED stereo camera mounted on the top of Racecar mini autonomous vehicle and Tiny-YOLO real-time object detection and classification system is used to detect and classify traffic signs	The proposed system was tested through the model on the dataset in terms of accuracy, loss, precision and intersection over union performance metrics
[12]	It is a review paper, and its main goal is to assist the researcher in the field of vehicle detection, tracking and classification	The areas of vehicle detection, tracking and classification were reviewed to provide guidance to the researchers
[13]	This article introduces a middleware system based on pub/sub messaging protocol	In this paper, topic-based publish/subscribe messaging model is used
[14]	This paper evaluates real-time performance for BGSLibrary algorithms on well-known datasets, which are Background Models Challenge (BMC) and Change Detection	The paper investigates which background subtraction algorithm is the most suitable regarding real-time performance and it is stated as Adaptative BG Learning in the paper
[15]	When the projection is changed from 3-D scene to 2-D object, the depth information of the object is lost. So, the paper focus on solving the depth estimation problem	The paper states that the accuracy rate of deep learning methods used for depth estimation from two-dimensional images by training neural networks is higher than other methods
[16]	A system is presented to detect the empty spaces available for parking between vehicles	The proposed approach computed the parking space length 10 to 50 cm better than with standard ultrasonic data. Position accuracy of the detected parking space was evaluated as below 8 cm of error in average. Length accuracy was below 16 cm in average
[17]	In the proposed work, parking spots are monitored with roadside cameras and form a dataset. A Convolutional Neural Network (CNN) based on this dataset is built. Then the trained CNN analyzes a new street image to check if there is a free spot or not	While optimizing CNN, several different neural networks with different activation functions are tried and the average of total response time was measured as around 15 s
[18]	The paper introduces a system that leverages the ubiquitous smartphone to help drivers find parking spaces in the urban environment called as ParkMaster	On-the-road experiments run in uncontrolled environments (city of Paris, Los Angeles and a small Italian village) validate ParkMaster approach, whose accuracy in estimating parking availability reaches 90%

Table 1 continued

| References	Methodology	Results
[19]	This paper presents a vision-based approach to detect the vacancies of parking spaces in an outdoor context	The paper introduces; first a new parking model based on the vertical on-street surfaces in order to handle the problem of inter-spaces occlusion and to correctly represent the parking independently of its disposition according to the surveillance camera
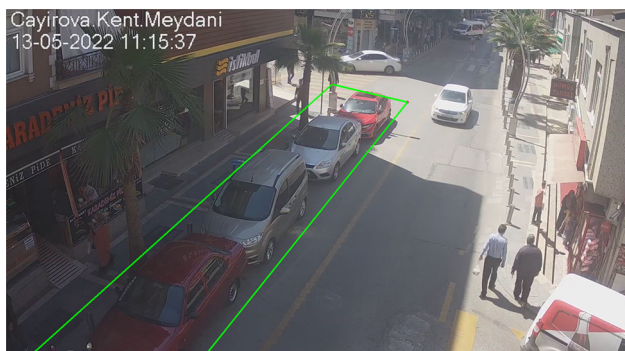
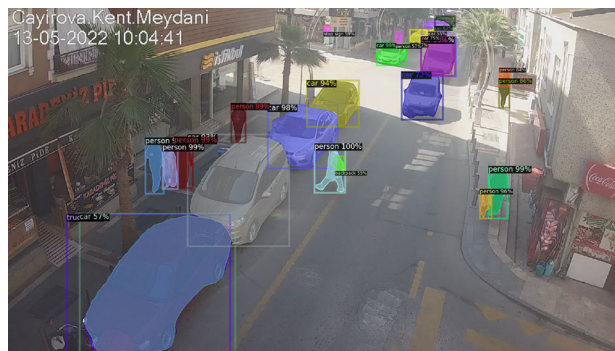**Fig. 1** Camera Configuration Interface

Fig. 2 Visualizing Model Outputs

the camera should not be buffered before the images are processed. OpenCV defines an automatic buffer number in the function and it is adjusted again so that the buffer number is 0 in order to instantly act on the last image.

Before processing images from the camera, the Detectron2[1] pre-trained Mask R-CNN segmentation model [20] to be used for vehicle detection and segmentation is loaded into the GPU. Computers' processors are insufficient to perform these operations, so GPUs with parallel computing capability are used in this study. After the model is loaded on the GPU, the threshold value to be used when processing images is determined as 0.48 when the area covered by the relevant surveillance camera is examined.

After the model and the camera settings are made, the snapshot in the live stream is taken and given as input to the model. The masks and overlay frames of the objects detected as output are accessed in Fig. 2.

3.3 Application of perspective transformation to parking area

By definition, perspective is a projection method used to transfer the view of the observer to a 2-dimensional plane depending on the location of the object. The images of the parking areas on the streets are usually at different angles, the calculation values will need to be changed in each area when analyzing the images. When perspective transformation is considered in this framework, effects such as scaling,

spreading, inverted image and shift on the given area are eliminated. The purpose of using perspective transformation within the scope of the study is to reduce the effect that will occur as a result of changing the position of the vehicles. Thanks to transformation, the angle of view of the parking area can be changed within certain restrictions, regardless of the angle, after the image is taken from the camera.

The area determined in the camera configuration step should be corrected according to the human view. One of the main reasons for doing this is that the frames of the vehicles in the cameras that see the area differently cover the places outside the vehicle's mask instead of only covering the turning points of the vehicle. In this case, it is very difficult to make the necessary approximations for depth calculations. In order to increase the accuracy of the calculations, the configuration file is first read to get only the vehicles in the designated parking area.

Detectron2 segmentation model outputs of the snapshot taken from the surveillance camera are thrown into an array containing only the car class. Certain operations are performed to determine that each car detected in the image is in the parking area of interest. Since the background data other than the detected car must be deleted, the segmentation data of the car is 1, that is, all other pixels in the white image are converted to 0 (white). The points determined for the parking area in the configuration step are read from the configuration file and a perspective transformation matrix is obtained with the help of OpenCV functions. Perspective transformation is applied to the masked image so that the relevant car prediction can be processed and is given in Algorithm 1.

---

[1] Detectron2 https://ai.facebook.com/tools/detectron2/

Algorithm 1 Applying Perspective Transformation to an Image

1: $region \Leftarrow configure\_points$
2: $clength \Leftarrow get\_distance(points[0], points[1])$
3: $clength \Leftarrow get\_distance(points[0], points[3])$
4: $dst \Leftarrow [[width, 0], [width, length], [0, length], [0, 0]]$
5: $M \Leftarrow cv2.getPerspectiveTransform(region, dst)$
6: $perspective\_image \Leftarrow cv2.warpPerspective(img, M, (cwidth, clength))$

Fig. 3 Applying perspective transformation to every car mask

In this way, it is understood whether the relevant car mask belongs to the parking area of interest and prevents unnecessary processing of passing cars. While determining whether the vehicle mask is in the parking area of interest, threshold is applied to the masked and perspective-transformed image, and then, the car in the image is framed. If the frame ends above or starts below the middle of the area of interest, then the vehicle is crossing the road or leaving the parking area. As a result of this operation, the specified situations are prevented. The reason why only the car masks in the parking area of interest are displayed in the mask images in Fig. 3 is that when the perspective transformation is applied to the masks of the other cars in the image, the projection of the relevant masks do not have a projection on the perspective of the parking area. Thus, it is ensured that vehicles outside the parking area are eliminated (Fig. 3).

3.4 Figuring out start and end points of cars

Since there are certain image distortions due to the use of homography in perspective-transformed car images, it is necessary to minimize the effect of these image distortions on the computation. The starting and ending points of the cars can be found clearly with the help of the frame (bounding box) when viewed from a straight angle at the level of the car. Since surveillance cameras are used in the related study, it is inevitable to obtain a skewed view. As a result, instead of accepting the start and end points of the vehicles as the frame start and end points, these points are re-determined by a simple operation. Considering the end point of the cars as the end of the hood and taking into account the homography shift, the diagonal approach is performed on the car frame with perspective transformation. Since the car frames will be rectangular, finding the diagonal length is provided by the Pythagorean theorem.

After the diagonal finding process is completed, all pixels in this diagonal are taken as the lines on the right in Fig. 4 and assigned to a list. The pixel list is checked from the
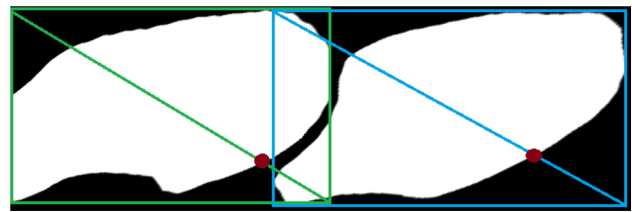

Fig. 4 Finding the end point of vehicles and diagonal approach

Table 2 Average length and width values by vehicle type

Vehicle type	Average length (m)	Average width (m)
Sedan	4.50	1.80
Hatchback	4.10	1.70
Suv	4.60	1.90
Jeep	4.75	1.90
Pickup	5.00	1.95

end point to the starting point. Pixel values are black and white only because the image is masked. The first pixel to encounter white is also the first pixel to cross the diagonal. By performing these operations, the end point of each car's hood is found with a more precise approach (Fig. 4 red intersection points).

3.5 Calculating free spaces between cars

After the starting and ending points of the cars are found, these points are kept in an array. The sequence of cars stored in the array is processed to find empty parking spaces between each vehicle. The lengths of the cars for use in this approach will be assumed to be an average constant.[2] The average length and width of the car obtained as a result of the research are expressed in Table 2;

When calculating the distance between the cars, the fixed length is assumed as 450 cm (4.5m). Certain variables were used while calculating the distance, and the explanations of these variables and their unit values are given in Table 3. Also, the following steps are followed when calculating the distance between two cars.

Pixel density refers to the number of pixels per cm for the real-world length of the objects in the picture. Pixel density calculation is calculated for the midpoint of the reference objects in the image. The pixel densities at the midpoints of the two cars;

$$car1\_px\_step = \frac{car1\_px}{car1\_length} \tag{1}$$

---

[2] automobiledimension https://www.automobiledimension.com/

Table 3 Variables used for depth approach and their explanations

Variable	Description	Value
car1_length	Actual length of the first car	cm
car1_px	Number of pixels occupied by the length of the first car in the image	pixel
car2_length	Actual length of the second car	cm
car2_px	Number of pixels occupied by the length of the second car in the image	pixel
car1_to_car2_gap_px	Number of pixels occupied by the end point of the first car and the starting point of the second car	pixel

$$car2\_px\_step = \frac{car2\_px}{car2\_length} \tag{2}$$

The density distribution in the whole picture is approached by looking at the changes in pixel intensities according to the positions of the cars in the picture. Thus, the calculation of the distance between two cars can be made.

First of all, the pixel difference between the midpoints of the cars was calculated by Eq. 3;

$$car1\_to\_car2\_mid\_px = \frac{car1\_px}{2}$$
$$+ car1\_to\_car2\_gap\_px + \frac{car2\_px}{2} \tag{3}$$

Then, the pixel density variation between two pixels is approximately calculated and given in Eq. 4;

$$px\_per\_cm\_density\_change$$
$$= \frac{car1\_pixel\_step - car2\_px\_step}{car1\_to\_car2\_mid\_px} \tag{4}$$

However, since the density changes nonlinearly throughout the picture, the margin of error in this approach needs to be reduced. The accuracy of the amount of pixel density variation per cm is increased and given in Eq. 5;

$$px\_per\_cm\_density\_change$$
$$= \left(\frac{car1\_px\_step}{car2\_px\_step}\right)^{apprx\_power}$$
$$* \left(\frac{car1\_px\_step - car2\_px\_step}{car1\_to\_car2\_mid\_px}\right) \tag{5}$$

If the pixel length of the distance to be calculated is more than the pixel length of the previous or next reference car, the density difference moves away from the approximation values. In order to reduce this margin of error, the approximation error correction coefficient was calculated as an additional correction in the approximation formula. This coefficient is called as apprx_power. The coefficient of apprx_power is determined as indicated in the pseudocode in Algorithm 2;

Algorithm 2 Determination of the apprx_power coefficient

1: **if** $gap\_at\_end > car2\_px$ **then**
2:   $apprx\_power \Leftarrow 1$
3: **else**
4:   $apprx\_power \Leftarrow 2$
5: **end if**

To calculate the length of the empty parking area after the density change calculation is made, a starting point is selected to the left of this area and the nearest pixel density can be calculated. (2nd car is chosen as an example.) The pixel density at this starting point is calculated as follows;

$$px\_step = px\_per\_cm\_density\_change$$
$$* \left(\frac{car2\_px}{2}\right) \tag{6}$$

The inverse of the pixel density at each pixel point in the image gives the real-world length of that point. For this reason, the real-world length of each point is calculated by moving pixel by pixel along the length of the parking area to be calculated. In order for the accuracy rate to converge to reality, a new pixel density value is calculated by subtracting the amount of pixel density change for the next pixel in each iteration given in Algorithm 3;

Algorithm 3 Calculation of Free Space Between Two Cars

Require: $dist = 0$
1: $i \Leftarrow gap\_at\_end$
2: **while** $i \neq 0$ **do**
3:   $dist \Leftarrow dist + 1/px\_Step$
4:   $px\_step \Leftarrow px\_step - px\_per\_cm\_density\_change$
5:   $i \Leftarrow i - 1$
6: **end while**

There are two ways to check if the number of cars used as a reference is less than two to find empty parking spaces in the parking area of interest in the image. The first case is that no cars can be detected. In this case, if the actual length of the parking space is known, the length of the parking space is divided by the average car length to find the number of empty parking spaces. However, if there is no information about the length of the parking space, then it is considered as one free space. In the second case, if there is only one car in

the parking lot and the actual length of the parking space is known, the actual length of a single car is subtracted from the length of the parking space to find the number of free parking spaces. However, if there is no information about the length of the parking space, according to the position of the car in the parking area, two empty parking spaces are indicated on the right and left of the car. In this specification, if there is an area smaller than the car to the right or left of the car, these areas are ignored so that they do not affect the number of empty parking spaces.

3.6 Mobile application and data processing

"Computer Vision" and "Deep Learning" methods were used in images taken from street surveillance cameras. With this study, it was ensured that the vehicle entrance and exit, full and empty parking spaces in the parking area were determined by close measurements. Using this information, a mobile application has been developed that interacts with the user and informs the users about the parking area. Flutter[3] language developed and supported by "Google" is used for the development of mobile applications on both platforms, both Android and IOS. "Google Firebase"[4] functions are used for storing, managing and transferring the data processed on the server side to the mobile application.

4 Results

In the proposed work, the parking spaces on the streets were determined by making use of the areas where the vehicles on the images taken from the cameras. With the help of the detectron2 Mask R-CNN model, the total detection time of the masks and frames of the cars in the parking lots is approximately 5 s on a computer running Ubuntu with 16GB RAM, Ryzen 5 5600x 3.7GHz processor. The metrics of the model are indicated in Fig. 5.

Empty and full parking lots were determined by perspective transformation and depth measurement techniques, and the data obtained were transferred to the Real-Time Database environment. Through the mobile application, users are provided with instant updates, notification that the parking spaces they choose are empty, and directions when necessary.

A new approach has been made by using depth measurement techniques and perspective transformation instead of line-based solutions used in the detection of parking spaces on the streets. In this way, it is possible to detect empty and occupied parking spaces with high accuracy in parking areas where there are no lines or marked spaces.

---

[3] Flutter https://flutter.dev/.

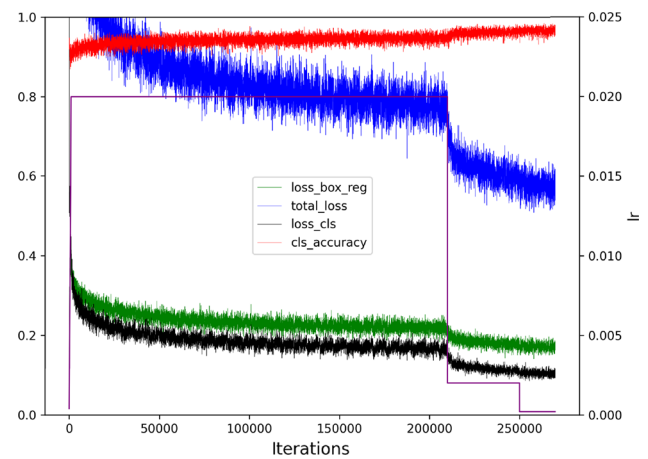[4] Google Firebase https://firebase.google.com/.

Fig. 5 Detectron2 Mask R-CNN R101 3X model metrics

In order to advance the study and obtain more precise results, the perspective transformation process should be reduced to the human gaze in an error-free and closest way. This reduction can be achieved by fixing the cameras in a certain position or by keeping the camera angle as constant as possible with new approaches. In addition to the mask removal method with segmentation, "key-point detection" models can be developed to use more precise approaches in vehicle detection and depth analysis.

Declarations

Conflict of interest There is no conflict of interest

Ethical approval Not applicable.

References

1. Park, K., Lee, D., Park, Y.: Video-based detection of street-parking violation. In: 2007 International Conference on Image Processing, Computer Vision, and Pattern Recognition, IPCV 2007, pp. 152–156 (2007)
2. Bulan, O., Loce, R.P., Wu, W., Wang, Y.R., Bernal, E.A., Fan, Z.: Video-based real-time on-street parking occupancy detection system. J. Electron. Imaging **22**(4), 041109 (2013)
3. Chen, L.-C., Sheu, R.-K., Peng, W.-Y., Wu, J.-H., Tseng, C.-H.: Video-based parking occupancy detection for smart control system. Appl. Sci. **10**(3), 1079 (2020)
4. Tătulea, P., Călin, F., Brad, R., Brâncovean, L., Greavu, M.: An image feature-based method for parking lot occupancy. Future Int **11**(8), 169 (2019)

5. Acharya, D., Yan, W., Khoshelham, K.: Real-time image-based parking occupancy detection using deep learning. Res. Locate **4**, 33–40 (2018)

6. Tschentscher, M., Neuhausen, M.: Video-based parking space detection. In: Proceedings of the Forum Bauinformatik. Citeseer, pp. 159–166 (2012)

7. Tomar, P., Kaur, G., Singh, P.: A prototype of iot-based real time smart street parking system for smart cities. In: Internet of Things and Big Data Analytics Toward Next-Generation Intelligence, pp. 243–263. Springer (2018)

8. Hirahara, K., Ikeuchi, K.: Detection of street-parking vehicles from panoramic street image. In: Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, vol. 2, pp. 993–998. IEEE (2003)

9. Yalcin, O., Sayar, A., Arar, O., Akpinar, S., Kosunalp, S.: Approaches of road boundary and obstacle detection using lidar. IFAC Proc. Vol. **46**(25), 211–215 (2013)

10. Yalcin, O., Sayar, A., Arar, O., Apinar, S., Kosunalp, S.: Detection of road boundaries and obstacles using lidar. In: 6th Computer Science and Electronic Engineering Conference (CEEC), vol. 2014, pp. 6–10. IEEE (2014)

11. Satılmış, Y., Tufan, F., Şara, M., Karslı, M., Eken, S., Sayar, A.: Cnn based traffic sign recognition for mini autonomous vehicles. In International Conference on Information Systems Architecture and Technology, pp. 85–94. Springer (2018)

12. Kul, S., Eken, S., Sayar, A.: A concise review on vehicle detection and classification. In: 2017 International Conference on Engineering and Technology (ICET), pp. 1–4. IEEE (2017)

13. Kul, S., Eken, S., Sayar, A.: Distributed and collaborative real-time vehicle detection and classification over the video streams. Int. J. Adv. Robot. Syst. **14**(4), 1729881417720782 (2017)

14. Kul, S., Eken, S., Sayar, A.: Evaluation of real-time performance for bgslibrary algorithms: a case study on traffic surveillance video. In: 2016 6th International Conference on IT Convergence and Security (ICITCS), pp. 1–4. IEEE (2016)

15. Akın, Z., Sayar, A.: Challenges in determining the depth in 2-d images. In: 2022 International Conference on INnovations in Intelligent SysTems and Applications (INISTA), pp. 1–6. IEEE (2022)

16. Abad, F., Bendahan, R., Wybo, S., Bougnoux, S., Vestri, C., Kakinami, T.: Parking space detection. In: Proceedings of the 14th World Congress Intelligent Transportation System, pp. 1–8 (2007)

17. Gören, S., Óncevarlk, D.F., Yldz, K.D., Hakyemez, T.Z.: On-street parking spot detection for smart cities. In: IEEE International Smart Cities Conference (ISC2), vol. 2019, pp. 292–295. IEEE (2019)

18. Grassi, G., Jamieson, K., Bahl, P., Pau, G.: Parkmaster: an in-vehicle, edge-based video analytics service for detecting open parking spaces in urban environments. In: Proceedings of the Second ACM/IEEE Symposium on Edge Computing, pp. 1–14 (2017)

19. Masmoudi, I., Wali, A., Jamoussi, A., Alimi, M.A.: Trajectory analysis for parking lot vacancy detection system. IET Intel. Transp. Syst. **10**(7), 461–468 (2016)

20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)