# Automatic Vision-Based Parking Slot Detection and Occupancy Classification

Ratko Grbić<sup>a,\*</sup>, Brando Koch<sup>a</sup>

<sup>a</sup> Faculty of Electrical Engineering, Computer Science and Information Technology Osijek, Kneza Trpimira 2B, Osijek, HR-31000, Croatia

## Abstract

Parking guidance information (PGI) systems are used to provide information to drivers about the nearest parking lots and the number of vacant parking slots. Recently, vision-based solutions started to appear as a cost-effective alternative to standard PGI systems based on hardware sensors mounted on each parking slot. Vision-based systems provide information about parking occupancy based on images taken by a camera that is recording a parking lot. However, such systems are challenging to develop due to various possible viewpoints, weather conditions, and object occlusions. Most notably, they require manual labeling of parking slot locations in the input image which is sensitive to camera angle change, replacement, or maintenance. In this paper, the algorithm that performs Automatic Parking Slot Detection and Occupancy Classification (APSD-OC) solely on input images is proposed. Automatic parking slot detection is based on vehicle detections in a series of parking lot images upon which clustering is applied in bird's eye view to detect parking slots. Once the parking slots positions are determined in the input image, each detected parking slot is classified as occupied or vacant using a specifically trained ResNet34 deep classifier. The proposed approach is extensively evaluated on well-known publicly available datasets (PKLot and CNRPark+EXT), showing high efficiency in parking slot detection and robustness to the presence of illegal parking or pass-

Preprint submitted to Expert Systems with Applications

<sup>\*</sup>Corresponding author.

*Email addresses:* ratko.grbic@ferit.hr (Ratko Grbić ), bkoch4142@gmail.com (Brando Koch)

ing vehicles. Trained classifier achieves high accuracy in parking slot occupancy classification.

*Keywords:* parking slot detection, parking occupancy, vehicle detection, deep learning, PKLot, CNRPark-EXT

## 1. Introduction

Urban population and the number of motor vehicles are constantly increasing and thus saturating not only the road network but also space reserved for the vehicles parking. The shortage of parking space is especially pronounced in crowded urban areas where the available parking space is in high demand throughout the whole day. With limited parking space, the drivers are forced to cruise for vacant parking slots and thus are creating mobile queues which affect normal traffic flow. Apart from traffic congestion, cruising for parking space creates additional pollution and  $CO_2$  emissions, negatively impacts driving time, and creates additional costs for the driver Zhu et al. (2020).

To make it easier to find a vacant parking space and to enable more efficient utilization of available parking spaces, different parking guidance information (PGI) systems are installed by parking providers Lin et al. (2017). Such systems require accurate and real-time information about the occupancy of each individual parking slot in a certain parking area in order to provide relevant information to the nearby drivers via mobile apps or information panels Guo et al. (2014). A straightforward approach for getting information about the occupancy of each parking slot is based on the installation of some kind of sensor on each parking slot. Typically, ultrasonic sensor Chen & Chang (2011), magnetic sensor Zusheng Zhang et al. (2015); Sifuentes et al. (2011) or even the combination of two sensors SENSIT; Alam et al. (2018) is mounted on each parking slot to detect the presence of a vehicle. While such approaches can provide highly accurate information to a PGI system, they require additional costs in terms of sensors cost, installation, and maintenance. Recently, parking slot occupancy detection based on computer vision emerged as a promising source of information for PGI systems Cai et al. (2019); Acharya et al. (2018); Ng et al. (2020); Amato et al. (2017). These solutions extract information about parking occupancy from images obtained by a camera that is recording parking area from a certain viewpoint such that parking slots are at least partially visible. This approach can be attractive since it does not require any additional sensors besides the camera and can provide additional information regarding parking usage such as vehicle recognition. Since many parking areas are already covered with surveillance cameras, such approach to parking occupancy detection can be quite cost-effective with respect to sensor-based systems and can be implemented quickly and more easily. Obviously, bad weather conditions like snow or rain, significant illuminance changes during daytime and nighttime operation, different camera viewpoints, vehicle occlusion, and presence of passing and illegally parked vehicles are the main challenges when developing and implementing vision-based PGI systems.

A significant problem in most of the vision-based solutions for parking slot occupancy classification is the requirement for manual labeling of the parking slots in the input image obtained by a camera that is recording a certain parking area, i.e. a parking lot. The resulting annotations are then used to extract every individual parking lot from the camera image. Each extracted patch is then classified as occupied or not by using some kind of classifier. This classifier can follow traditional computer vision approach (typically feature extraction + SVM classifier de Almeida et al. (2015)) or can be based on deep learning (DL) approach Acharya et al. (2018); Amato et al. (2017). However, this manual labeling can be cumbersome and time-consuming if the system is going to process images from several cameras. Apart from that, in case of camera angle change or zooming, maintenance, or camera replacement the labeling procedure must be repeated. In this spirit, automatic parking space detection is recently pointed out in de Almeida et al. (2022) as a significant problem of modern vision-based parking lot management.

A bunch of training data is a must if DL classifier is to be used for parking slot occupancy classification. This lack of a consistent and representative dataset was recognized by the research community. In de Almeida et al. (2015), the PKLot dataset was proposed which is a relatively large dataset that contains labeled images from two parking scenes from different camera views in different weather conditions. Similarly, in Amato et al. (2017) CNRPark+EXT dataset is presented which contains annotated images of parking scenes in different weather conditions as well during the night. Both datasets are publicly available and are often used for the training and benchmarking of DL classifiers.

There are several attempts reported in literature which are using object detection as a source of information regarding parking slot occupancy. In Ke et al. (2020) authors are using single-stage object detector on edge devices and send the detections for further analysis by the PGI system server. In Padmasiri et al. (2020) automated vehicle parking occupancy detection is performed in real-time with two-stage detectors Faster R-CNN and RetinaNet. However, many false positive detections arise in such approach due to passing vehicles. Additionally, most of such approaches are focused on the implementation part without getting into detailed evaluation on publicly available datasets like de Almeida et al. (2015); Amato et al. (2017) and some require additional a priori knowledge to efficiently judge parking occupancy status based on input image Martin Nieto et al. (2019).

In this paper, vision-based algorithm for Automatic Parking Slot Detection and Occupancy Classification (APSD-OC) is proposed. The proposed APSD-OC removes the need for manual labeling of parking slots and further improves the accuracy of the occupancy classification. We approach the problem of parking slots position determination as a vehicle detection problem in the series of images captured by a camera through a certain period of time and by taking into account that drivers usually park their vehicles inside marked parking slots. The vehicle detection is performed on a set of input images followed by the appropriate algorithm for determining positions of the marked parking slots which includes perspective transformation and detections clustering. As such, APSD-OC can distinguish regular parking slots from parking violations, i.e. vehicles parked outside any marked parking slots, and consequently exclude the latter from further analysis. Once the parking slots are detected, in each subsequently captured image these parking slots are cropped and each is classified as occupied or vacant with the proposed classifier. The complete procedure can be easily applied to any parking surveillance problem since it does not require any additional parameters which are difficult to obtain like the camera angle, camera parameters, or homography matrix. The only required parameter to be provided by end-user is the number of visible parking slots which can be easily obtained and thus the whole algorithm can be applied to a different parking lot relatively easily. The proposed approach is extensively analyzed on the two well-known publicly available datasets and is compared with the state-of-the-art solutions showing the high efficiency both in parking slot detection and parking slot occupancy classification.

The paper is structured as follows. In Section 2 related work is overviewed, advantages and disadvantages of recent approaches are pointed out. The proposed algorithm for automatic parking slot detection and occupancy classification is presented in Section 3. The description of used datasets and how experiments are performed are given in Section 4 with the obtained results and accompanying discussion. In the end, conclusions are given with guidelines for future work.

## 2. Related Work

A rough categorization can be made on vehicle-driven and space-driven methods as proposed in Huang & Wang (2010). The vehicle-driven methods are focused on vehicle detection upon which vacant parking slots are determined. In the latter case, the focus is on direct detection of the available parking slots in an overall scene. Similarly, in Martin Nieto et al. (2019) a categorization is given regarding how methods perform parking slot classification: image segmentation based systems Huang & Wang (2010); Al-Kharusi & Al-Bahadly (2014), machine learning over parking slots patches Tschentscher et al. (2015); Amato et al. (2017) and vehicle detection techniques based on object detectors Martin Nieto et al. (2019); Padmasiri et al. (2020); Xie et al. (2015).

Also, two main directions of research can be differentiated in the literature regarding used algorithms. The first one is based on traditional computer vision techniques Al-Kharusi & Al-Bahadly (2014) which are often coupled with (shallow) machine learning methods (e.g. support vector machines - SVM) to answer the question of whether a certain parking slot is occupied or not de Almeida et al. (2015). The second direction of research is more recent and uses deep learning as its main engine to determine parking slot occupancy status Nurullayev & Lee (2019); Acharya et al. (2018); Amato et al. (2017); Padmasiri et al. (2020); Ke et al. (2020); Cai et al. (2019); Coleiro et al. (2020); Martin Nieto et al. (2019). In both cases, a certain mechanism is implemented or a priori information is given to the system which determines actual parking slots in the scene. Some of the aforementioned papers are overviewed in more detail in the rest of this section.

An intelligent parking management system, based solely on traditional image processing techniques, is proposed in Al-Kharusi & Al-Bahadly (2014). This includes colorspace transformation, morphological operation (dilate and erode), thresholding, edge detection, and Hough transform. The focus of the paper is on the overall system and the actual efficiency regarding parking slot classification is not reported. The authors also point out that the method does not have the same efficiency in different weather conditions.

Although the problem of parking occupancy can be solved with traditional computer vision techniques, most of the recent papers introduce machine learning to build a more efficient parking slot occupancy classifier. Paper de Almeida et al. (2015) is a significant work in the field of parking management since it proposed the PKLot dataset and thus enabled a systematic benchmark of different parking occupancy classification methods. Apart from that, the authors proposed a parking slot occupancy SVM classifier based upon textual descriptors such as Local Binary Patterns (LBP) and Local Phase Quantization (LPQ). When the same view of parking was used for training and testing the classifier, the recognition rate went over 99%. However, when the classifier was trained on one set of parking lots and tested on another one, the recognition rate was 89%, indicating that further research in generalization capabilities of used classifiers should be performed.

Many researchers recognized that hand-crafted visual features (SIFT, SURF, ORB, etc.) have a limited ability to adapt to variations of object appearance that are highly non-linear, time-varying, and complex. Interestingly, pretrained CNN showed as an excellent "off-the-shelf" feature extractor in many different visual recognition tasks Razavian et al. (2014). Therefore, in Acharya et al. (2018) authors are using VGG CNN Simonyan & Zisserman (2014) pretrained on ImageNet to efficiently extract features and to train a binary SVM classifier for the purpose of parking occupancy classification. The authors trained the classifier on the PKLot dataset and tested it on a specific dataset called Barry street. This dataset contains 810 images of the parking lot with 30 parking slots resulting in 24300 annotations, i.e. patches of occuppied or vacant parking slots. The proposed approach achieved 96.6% accuracy on the Barry street dataset. The authors also point out that in practice it should be able to detect the predefined areas of the parking slots automatically rather than manually identifying the boundaries.

In Ahrnbom et al. (2016) authors developed a parking slot occupancy classifier that combines Integral Channel Features (ICF) with Logistic Regression (LR) or SVM. The proposed method was designed to achieve good accuracy and robustness but keeping in mind overall method complexity so that it can run on embedded devices. At first, ten feature channels are extracted for each input image, such as color channels in LUV color space, gradient magnitude, and quantized gradient channels. Then, feature vectors are calculated from a certain feature channel in an efficient manner using the integral image approach. In the end, logistic regression and SVM classifiers are trained. Both types of classifiers are trained and tested on the PKLot dataset.

Authors in Amato et al. (2017) specifically designed a deep neural network called mAlexNet for the purpose of parking occupancy classification. The efficiency of the method is extensively tested on PKLot and CNRPark-EXT datasets and outperforms AlexNet and LPQ from de Almeida et al. (2015) in terms of classification accuracy and area under the curve (AUC). Interestingly, this deep network is three orders of magnitude smaller than the original AlexNet and can be implemented on an embedded platform like Raspberry Pi 2 model B.

In Nurullayev & Lee (2019), CarNet is proposed which is DNN that uses dilated convolutional neural network to indicate parking space occupancy status. Input to the CarNet is a 54x32 RGB image of a parking slot. The presented experiments show that CarNet outperforms AlexNet Krizhevsky et al. (2012) and other well-known DL architectures on PKLot dataset and mAlexNet Amato et al. (2017) on CNRPark-EXT dataset. While CarNet achieves quite high precision and robustness, it requires that parking slot images are manually cropped from the input image of a whole parking lot. The comparison of CarNet with our approach can be found in the results section.

Authors in Ke et al. (2020) point out that a key component of modern smart cities is traffic surveillance which needs significant computing power and storage. In case of a parking space occupancy, computing workload can be moved toward the edge, i.e. local devices equipped with cameras. The proposed approach balances computational load and data transmission volume. Therefore, a singlestage object detector called Single Shot Detector (SSD) is used on edge devices to detect vehicles. The detections are sent to the server which runs object tracking (to reduce false positive detections) and occupancy judgment algorithms. The obtained results on a parking garage use-case show that such approach yields efficient and reliable detection performance in various environmental conditions.

In Padmasiri et al. (2020), a scalable software architecture solution is proposed which enables reliable implementation of an end-to-end automated vehicle parking occupancy detection system. The vehicle detection is performed in real-time with two-stage detectors Faster R-CNN and RetinaNet. The proposed approach is tested on the PKLot dataset and the efficiency of the object detector is expressed in average precision (AP). The status of each parking lot is obtained by an object detection algorithm but the actual parking slots position detection is missing so that moving vehicles are often detected as occupied parking spaces. The authors put significant effort into system usability so the developed web-based and mobile-based applications enable end-users easy finding of free parking slots.

Video-based parking occupancy detection is proposed in Chen et al. (2020). Hereby, YOLOv3 Redmon & Farhadi (2018) based on MobileNet version 2 is used for vehicle detection. To train YOLOv3, authors have firstly labeled the CNRPark dataset. Additionally, the authors implemented a voting mechanism to prevent false positive classifications of parking spaces caused by large vehicles passing by. The solution was evaluated using the CNRPark-EXT dataset, a simulated model of a street with parking lots, and real images taken with a camera. The information regarding parking lot occupancy is further processed by a streetlight control system. The authors report high accuracy on the CNR-Park+EXT dataset. Unfortunately, test data and annotations are not publicly available.

Obviously, manual labeling of the parking spots as proposed in Amato et al. (2017); Nurullayev & Lee (2019) can be cumbersome and time-consuming, especially in the case of parking with large number of parking slots. Additionally, in the case of camera movement, the labeling process must be repeated. The need for automatical identification of the parking slots boundaries in practical applications is clearly pointed out in Acharya et al. (2018); de Almeida et al. (2022). There are several attempts in the literature to perform automatic parking slot detections. In Martin Nieto et al. (2019) authors proposed automatica detection of each parking slot when car park area is rectangular and forming a parking grid. However, additional information must be provided, e.g. user must specify the corners of the parking area and the number of slots, as well as camera homography matrix has to be estimated. Recently, two step automatic parking slot detection is proposed in Patel & Meduri (2020). At first, vehicle detection is performed using Faster R-CNN or Yolov4 and then vehicle tracking is performed to distinguish between stationary vehicles and moving vehicles. While the proposed apporach obtains pretty high recall values for CNRParkEXT datasets (even 100% during busy days when all slots are occupied), it appears that false positives can occur due to illegally parked vehicles.

## 3. Proposed Approach to Automatic Parking Slot Detection and Occupancy Classification

A parking slot is defined as an area that is designed for a single vehicle parking and is usually marked with painted white or blue lines on a road surface. Automatic parking slot detection is defined as the localization of a parking slot within the input image obtained by a camera that is recording the parking area. One way to describe the parking slot location in the image is by using a rectangular bounding box (BB) which is defined by its center, width, and height in pixels. Our approach for automatic parking slot detection relies on the processing of the camera images which are acquired with a certain sampling frequency. This is typically 5 minutes in practice. The core assumption of our proposed approach is that parking slots are parts of an image where repeated vehicle detections occur since the drivers are forced to park the vehicle inside the marked area. However, one should be aware that drivers sometimes make parking violations, i.e. they park vehicles in a prohibited space such as entrance or sidewalk, and which can lead to congestion or even accidents. Such locations should be ignored in parking slot detection since they are not valid parking slots. An example of a parking lot image from the PKLot dataset is shown in Figure 1. Properly parked vehicles are those parked inside slot markings and are marked with blue BBs. The rest of the vehicles are making parking violations and are marked with yellow BBs. Once the detection of parking slots is finished, newly acquired images are processed to obtain the status of each parking slot (is it occupied or vacant) and the information can be sent to a PGI system.

The proposed Automatic Parking Slot Detection and Occupancy Classification (APSD-OC) algorithm contains two main parts as can be seen in Figure 2 with grey overlays. The upper part performs parking slot detection outputting the corresponding rectangular area for each detected parking slot. Once the de-



**Figure 1:** An example of a parking lot image from PKLot dataset. Properly parked vehicles are marked with blue bounding box.

tection part is finished, the bottom part determines the status of each detected parking slot using a deep classifier.

Automatic parking slot detection is based on the processing of available parking lot images which were captured using a camera with a fixed position and angle. These images should be collected over a period of time in which all parking slots are used for vehicle parking so they can be properly detected. To each obtained image an object detector is applied which outputs a rectangular BB for each detected vehicle inside the image. Once all images are processed, BB centers clustering is performed to reveal locations where vehicles are detected consistently. However, since parking cameras can be mounted in different physical locations and angles, images can be obtained from different perspectives which can affect the efficiency of the clustering process due to different BB center densities in areas closer to the camera versus areas more far back. Therefore, BB centers are firstly transformed using a homography matrix which relates the original camera view and the bird's eye view of the parking lot. Once the cluster centers are obtained in bird's eye view, these are further examined by analyzing the distribution of BB centers that belong to each cluster. By doing so it can be distinguished to a certain degree if a cluster center corresponds to a regular parking slot or to some kind of unmarked parking lot area where sporadical or illegal parking occurs. Parking slot occupancy classification can be relatively easily performed once the precise locations of parking slots are available. All detected parking slots are cropped from the input image and are processed using a deep classifier which outputs the probability that each parking slot is occupied.



Figure 2: The block diagram of the proposed APSD-OC algorithm.

## 3.1. Vehicle Detection Using YOLOv5

The first step in the proposed APSD-OC algorithm is to detect vehicles in a sequence of N input images taken by a camera recording the same parking lot for a certain period of time. A vehicle detection implies locating a vehicle in the image, typically with the rectangular BB which tightly surrounds the vehicle. Therefore, each detected vehicle in image is represented with a vector  $\mathbf{v}_i = [\mathbf{c}_i, w_i, h_i]$  where  $\mathbf{c}_i = [x_i, y_i]$  is BB center,  $w_i$  is BB width, and  $h_i$  is BB height. An example of vehicle detection in an input image from PKLot dataset is shown in Figure 3a where BBs are marked with blue color and blue dots represent corresponding BB centers.

For the task of vehicle detection, we use YOLOv5, an object detector from the YOLO family of end-to-end deep learning models designed for real-time object detection. This model was selected as it has proven to have solid performance on our dataset images. YOLO object detectors Redmon & Farhadi (2018), compared to R-CNNs Ren et al. (2015), replace the need for a region proposal step by splitting the input into a grid of cells, each containing a predefined number and shape of BBs over which an objectiveness score and a class prediction are made. Objectiveness score is the probability of a particular cell containing the object, while the class prediction is the probability distribution over the classes. This architecture choice enabled the efficiency of YOLO family. For the purpose of detecting vehicles in images a version yolov5x of the YOLOv5 model pretrained on the COCO dataset is used Jocher et al. (2021). COCO dataset contains 200,000 labeled images with 1.5 million object instances and 80 object categories. During inference, we only considered classes "car" and "truck" and used a confidence threshold of 0.5 during the processing of all images. Before entering the network images were resized to  $1280 \times 1280$  px.

In this algorithm step, vehicle detection is performed for N available images for a given camera as shown in Figure 3b (just centers of BBs are shown). The resulting BBs are stored for further processing with a clustering algorithm in order to detect actual parking slots and to discard locations corresponding to illegal parking, bus stops, etc.

#### 3.2. Perspective Transformation

Generally speaking, a camera that is recording a parking lot should be mounted in such a way that all parking slots can be seen as much as possible, i.e. vehicle presence and its position can be clearly distinguished in the obtained image. Therefore, installation of the camera depends on several factors for a particular parking lot which include camera field-of-view, number of parking lots, presence of the trees or other structures, and so on. For example,



(a) Vehicle detection in a single input image. (b) Vehicle detection in multiple input images.

Figure 3: Vehicle detection using Yolov5 in input images from PKLot dataset.

the PKLot dataset de Almeida et al. (2015) is created using a low cost full high definition camera (Microsoft LifeCam, HD-5000) positioned at the top of a nearby building to minimize the occlusion between adjacent parked vehicles.

Figure 3a shows a single input image from PKLot dataset with highlighted BBs and their centers using detector from Subsection 3.1. If N such images are processed using this detector, as shown in Figure 3b, then it can be noticed that clusters of BB centers emerge on positions where vehicles are often parked. Due to the camera perspective, adjacent clusters that correspond to parking slots closer to the camera appear more distant than the adjacent clusters corresponding to parking slots that are further away from the camera. Therefore, to make the clustering process more efficient, the APSD-OC algorithm uses perspective transformation to obtain a bird's eye view (or top view) of a parking lot, minimizing the discrepancy between the distance of the adjacent clusters for any part of the parking lot.

The aforementioned perspective transformation can be achieved with the usage of 3x3 homography matrix **H** which maps each BB center image coordinates  $\mathbf{c}_i = [x_i, y_i]^{\top}$  to the bird's eye view  $\mathbf{c}'_i = [x'_i, y'_i]^{\top}$  according to  $[x'_i, y'_i, 1]^{\top} = \mathbf{H}[x_i, y_i, 1]^{\top}$ . In practice, a homography matrix is not known and has to be estimated. For example, in Martin Nieto et al. (2019) the homography matrix for each camera is obtained using four points from each camera viewpoint and each point correspondence in an image extracted from a top view. The top view is in this case obtained from Google Earth. Obviously, such an approach requires certain manual work and requires additional information (top view) which sometimes might not be available.

To make the proposed APSD-OC algorithm more general we automate homography matrix **H** estimation by using the approach of obtaining bird's eye view from an image proposed in Abbas & Zisserman (2019). Hereby, homography is parametrized with only four parameters corresponding to the vertical vanishing point and ground plane vanishing line (horizon) in the image. These are regressed directly using a CNN. The CNN is trained on a large synthetic dataset which contains ground truth for the horizon line and the vertical vanishing point and which is built using CARLA simulator. The dataset is created by randomly changing camera height, field of view, roll and tilt angle thus obtaining different camera positions and orientations that can be found in practice.

Figure 4a shows the perspective transformation of the image shown in Figure 3b. It can be noticed that the distance between BB centers that correspond to adjacent parking slots is approximately the same in the whole image when the perspective transformation is applied, i.e. in bird's eye view.

## 3.3. Bounding Box Centers Clustering Using DBSCAN

Figure 3b shows the image of the parking lot with overlayed vehicle detections in N images and the corresponding bird's eye view is shown in Figure 4a (for the sake of simplicity just BB centers are shown). It can be noticed that BB centers tightly group around the center of each parking slot, but also there are a number of detections outside any parking markings which are due to illegal vehicle parking, passing vehicles, or even false vehicle detections. The former ones look like high-density regions and the latter ones appear as low-density regions (or outlying observations). In this case, it is beneficial to use some kind of density based spatial clustering in the bird's eye view.

Here we opt for a well-known DBSCAN algorithm Ester et al. (1996). DB-SCAN relies on density based notion of clusters and it can detect clusters of arbitrary shape. Given the set of points, DBSCAN groups together points that are close to each other in certain space (usually Euclidean distance is applied)



(a) Transformed BBs centers of vehicle detections for N input images.
(b) Centers of clusters obtained by DBSCAN algorithm applied to transformed BB centers.



(c) Noise points of DBSCAN algorithm.



**Figure 4:** Example of clustering of BB centers using DBSCAN in bird's eye view - an example from PKLot dataset (UFPR05 camera).

and if their count exceeds a predefined threshold. As such, DBSCAN requires only two parameters: *eps* defines how close two points should be to each other to be considered as a part of the same cluster and *minPoints* which is the minimum number of points for a region to be considered as a cluster. Points that are not reachable from any other point belonging a cluster are considered as outliers or noise points.

Figure 4b shows the application of DBSCAN algorithm on transformed BB centers from Figure 4a. Each cluster of BB centers is represented with mean

value  $\mathbf{m}_i$  of belonging points in bird's eye view. It can be noticed that most cluster centers  $\mathbf{m}_i$  correspond to parking slots locations. However, some centers can be found in the middle of the road representing clusters that are formed by the detections of illegaly parked vehicles. Noise detections, or more precisely centers of such vehicle detections, are shown in Figure 4c. Most of these noise points are due to passing cars or very short illegal parking (e.g. short stop behind a properly parked vehicle). For the next step of the algorithm only cluster centers  $\mathbf{m}_i$  are considered while the noise points are discarded.

#### 3.4. Cluster Centers Filtering and Inverse Transformation

The last step in automatic parking slot detection is filtering of the obtained cluster centers  $\mathbf{m}_i$  and their transformation to the original view. Most of  $\mathbf{m}_i$ values correspond to parking slots, see for example Figure 4b. However, some of these values can correspond to physical locations where parking violations often occur. To efficiently filter out such values, we take into account that the deviation of BBs centers around the corresponding mean values is usually much greater in case of parking violations than the deviation of BBs centers around the corresponding mean value for regular parking since in the latter case drivers are forced to park the vehicle inside slot markings. More precisely, for each  $\mathbf{m}_i$  we define  $s_i$  as the sum of standard deviations of x and y coordinates of corresponding BB centers. These  $s_i$  values are then sorted in ascending order and every value outside the range [Q1 - 1.5 \* IQR, Q3 + 1.5 \* IQR] is discarded. Hereby, Q1 is the first quartile of the BB centers, Q3 is the third quartile of the BB centers, and Inter-Quartile Range is IQR = Q3 - Q1. After that, the first  $n_{bottom}$  values are selected as final parking slots where  $n_{bottom}$  corresponds to the number of visible parking slots and must be provided by the user.

This step is illustrated in Figure 4d where each cluster center  $\mathbf{m}_i$  has a corresponding standard deviation  $s_i$ . It can be noticed that cluster centers  $\mathbf{m}_i$  corresponding to parking slots have a low value of  $s_i$  while these values are significantly larger in locations where parking violations occur. For this particular parking lot the parameter  $n_{bottom}$  is equal to 44. The cluster centers

 $\mathbf{m}_i$ , which are filtered out in this algorithm step, are shown in yellow color.

The very last step in automatic parking slot detection is the transformation of cluster centers  $\mathbf{m}_i$  to the original view using matrix  $\mathbf{H}^{-1}$ . Since the second part of the proposed APSD-OC algorithm predicts the occupancy of each parking slot, it is necessary to add area  $\mathbf{a}_i$  in form of BB around each  $\mathbf{m}_i$  which will be cropped and analyzed by the deep classifier. The area is defined as the mean of BBs that belong to the certain cluster. This is illustrated in Figure 5 where  $\mathbf{m}_i$  and area  $\mathbf{a}_i$  are shown in original camera view for two different parking lots from PKLot and CNRPark-EXT dataset.



(a) Filtered cluster centers  $\mathbf{m}_i$  (in blue) cor- (b) Area  $\mathbf{a}_i$  of each detected parking slot. responding to the parking slots.



(c) Filtered cluster centers  $\mathbf{m}_i$  (in blue) cor- (d) Area  $\mathbf{a}_i$  of each detected parking slot. responding to the parking slots.

Figure 5: Cluster centers filtering and inverse transformation.

## 3.5. Occupancy Classification Using Deep Classifier

Obtained parking slot locations in original view are portions of an image which can be cropped and used as input to an image based parking occupancy classifier. For this purpose, a ResNet34 deep classifier pretrained on ImageNet was fine-tuned and benchmarked on various splits from the PKLot and CNRPark-EXT datasets. ResNet architecture introduced residual modules which allowed for building deeper networks without vanishing and exploding gradients. Residual modules achieve this by utilizing skip connections along which gradients flow more easily.

We followed the Amato split regarding training/testing data Amato et al. (2017), so we can easily compare our DL classifier with current state-of-theart. Before training, we replace the ResNet34 head with randomly initialized weights and freeze the rest of the layers. We use a learning rate finder Smith (2015) to determine the base learning rate. The learning rate finder determines the optimal learning rate by starting a training with a low learning rate and doubling it for each subsequent minibatch until loss starts increasing. Our optimal learning rate is then an order of magnitude less than the learning rate when the loss starts increasing. A learning rate scheduler is also used with linear warmup on 30% of batches and cosine annealing on the rest, based on the 1 cycle policy Smith & Topin (2017). We also used cyclical momentum Smith (2018) which we vary in the opposite direction of the learning rate. The used optimizer is Adam. Initial momentum is set to 0.95, minimum momentum is set to 0.85and a final momentum is set to 0.95. We train like this for one epoch, unfreeze, then do the same over 5 epochs with discriminative learning rates sliced from  $base_lr/200$  to  $base_lr/2$ .

Two examples of the application of the proposed APSD-OC algorithm on the parking lot from the PKLot dataset are shown in Figure 6. It can be noticed that regular parking slots were successfully detected while the parking violations are ignored. Vacant and occupied parking slots were successfully classified by the trained deep classifier in both examples and are marked with green and red BBs, respectively. Similarly, Figure 7 shows the example of APSD-OC algorithm application on a camera from CNRPark-EXT dataset. It can be noticed that parking slots are successfully detected although significant occlusion by the trees is present (see Figure 7a). Two parking violations and a passing vehicle are successfully ignored in Figure 7b.



(a) Parking lot with all vehicles properly (b) Crowded parking lot with many parking parked. violations.

Figure 6: Two examples from PKLot dataset and final output of the proposed APSD-OC algorithm.



(a) Parking lot with several properly parked(b) Crowded parking lot with a passing ve-vehicles.hicle and two parking violations.

Figure 7: Two examples from CNRPark-EXT dataset and final output of the proposed APSD-OC algorithm.

## 4. Results and Discussion

Two publicly available datasets are used for APSD-OC algorithm evaluation: PKLot de Almeida et al. (2015) and CNRPark-EXT Amato et al. (2017). PKLot dataset contains the images from the parking lot of the Federal University of Parana (UFPR) and the Pontifical Catholic University of Parana (PUCPR), Brazil. Images were captured every 5 minutes in time interval over 30 days. In that way, three different weather conditions are present in the dataset: sunny, cloudy and rainy. The images have 1280 x 720 pixels resolution and were stored without compression in JPEG format. Two different parking lots were captured. The images are organized into three folders with respect to the location of capturing: UFPR04, UFPR05, and PUCPR. The first two contain images of the parking lot captured from the 4th and 5th floor of the UFPR building, while the last one contains images of the parking lot captured from the 10th floor of the PUCPR administration building. Each valid parking space, i.e. parking slot which is signed with a yellow or white line is annotated with the oriented BB and can be easily extracted from the whole parking image. The CNRPark-EXT dataset is an expansion of the original CNRPark dataset Amato et al. (2017), containing labeled images of the parking lot in the campus of the National Research Council (CNR) in Pisa, Italy. The parking lot contains a total of 164 parking slots which are captured by 9 cameras with different points of view and different perspectives from November 2015 to February 2016. In that way different weather and light conditions were captured: sunny, rainy and overcast. However, parking slots are annotated with non-rotated BBs which often do not cover precisely or entirely the parking slot area. The sample images from each dataset can be seen in Figure 8. The statistics regarding each dataset can be found in Table 1. Clearly, PKLot has significantly higher numbers of annotated parking slots. However, CNRPark-EXT has a lot of challenging images due to the occlusion of nearby objects such as lamps, trees, or other vehicles.

The evaluation of the proposed APSD-OC algorithm is performed with respect to the efficiency of the parking slot detection from the camera images and with respect to the efficiency of the parking slot occupancy classification throught the set of experiments. In the first part of experiments, images from the parking camera are fed to the algorithm which determines positions of parking slots. In the second part of experiments, cropped images of the parking slots



(a) PKLot PUCPR sunny

(b) PKLot UFPR04 cloudly

(c) PKLot UFPR05 rainy



(d) CNRPark-EXT C1 sunny (e) CNRPark-EXT C8 overcast (f) CNRPark-EXT C3 rainy

Figure 8: Sample images from PKLot and CNRPark-EXT datasets in different weather conditions.

Dataset	Image res-	Sample	# of views	# of im-	# of anno-
	olution	$\mathbf{time}$		ages	tations
PKLot	$1280\mathrm{x}720~\mathrm{px}$	5 minutes	3	12,417	695,900
CNRPark-	$1000 \mathrm{x} 750 \mathrm{px}$	5 minutes	9	4,278	144,965
EXT					

Table 1: PKLot and CNRPark-EXT datasets properties.

are fed to the classifier which determines parking slot occupancy status.

## 4.1. Parking Slot Detection Results

Although PKLot dataset and CNRPark-EXT dataset contain annotations of the regular parking slots, these cannot be used to evaluate the proposed automatic parking slot detection algorithm since some of the regular parking slots are not annotated within the dataset. This means that some false positive detection can arise during detector evaluation which are in fact regular parking spaces.

Therefore, for parking slot detection evaluation, we manually counted the

number of parking slots for each parking camera in the PKLot dataset and CNRPark-EXT dataset. Only parking slots that are visible and which are marked with (white) separating lines are counted. In that way, we obtained a total of 41, 44, and 170 regular parking slots for UFPR04, UFPR05 and PUCPR. In comparison, PKLot paper de Almeida et al. (2015) reported 28, 40 and 100 parking slots for UFPR04, UFPR05, and PUCPR parking cameras. In the case of the PUCPR parking camera, we analyzed only a middle part of the camera image which contains six rows of parking slots. A similar practice was used in the PKLot dataset paper, where only part of the whole image was used for cropping of the training and testing images for classification purposes. The discarded upper part of the image corresponds to the parking area which is far from the camera and consequently, vehicles appear quite small in the image. The discarded lower part of the image corresponds to the area where vehicles are not clearly or completely visible. Regarding the CNRPark-EXT dataset, whole input camera images were used in the case of cameras C2-C8, while in the case of camera C1 and C9 part of the image was analyzed.

Standard metrics for object detections are used for efficiency of parking slot detection:

- TP detection of existing parking slot
- FP detection of non-existing parking slot
- FN existing parking slot not detected
- Precision = (TP)/(TP + FP)
- Recall = (TP)/(TP + FN)

Table 2 shows the obtained parking slot detection results for PKLot dataset. The available images for each camera were divided in a way that chronologically first 30%, 50%, 80%, or all 100% of camera images are used for parking slot detection. It can be noticed that FN detections are significantly dropping as the volume of the available data is increased, especially for the PUCPR parking lot. This is expected since more and more vehicles are parked inside parking slots thus satisfying *minPoints* parameter of the DBSCAN algorithm. On the other hand, FP results are not significantly changing, proving the robustness of the proposed approach to false vehicle detections, passing vehicles, parking violations, and so on. For UFPR04 and UFPR05 parking cameras precision and recall go beyond 90% when over 50% of the available data is used, while in the case of PUCPR this happens at 80% since this is a much larger parking lot. For example, the algorithm has only one FP and one FN detection for UFPR05 camera when 100% of the data is used. This result can be also seen in Figure 5b. Interestingly, we noticed that UFPR04 camera is moved at a certain point in time and our assumption about fixed camera position and angle is not satisfied. Therefore, results for 80% and 100% of UFPR04 camera images are not reported in Table 2.

Sample	тр	$\mathbf{FP}$	$\mathbf{FN}$	Precision	Recall	# of
$\mathbf{size}$				[%]	[%]	parking
						slots
			UFPR0	)4		
30%	36	5	5	87.80	87.80	41
50%	38	3	3	92.68	92.68	41
80%	R	esults not rep	ported due to	the camera move	ement	41
100%	R	esults not rep	ported due to	the camera move	ement	41
			UFPR0	)5		
30%	36	8	8	81.82	81.82	44
50%	40	3	4	93.02	90.91	44
80%	42	2	2	95.45	95.45	44
100%	43	1	1	97.73	97.73	44
PUCPR						
30%	142	11	28	92.81	83.53	170
50%	147	10	23	96.63	86.47	170
80%	155	9	15	94.51	91.18	170
100%	158	7	12	95.76	92.94	170

Table 2: Results of parking slot detection on PKLot dataset.

Table 3 shows the obtained parking slot detection result for CNRPark-EXT dataset. Hereby, the same pattern can be observed like in PKLot dataset results - more data mostly ensure higher precision and recall in parking slot detection. Interestingly, for the number of cameras (C2, C4, C5, C6, C7, C8, C9) 100% precision is obtained since in these camera images there are practically no parking violations so each detection corresponds to the actual parking slot. However, for most of the cameras obtained recall is somewhat lower than for the PKLot dataset which can be attributed to the occlusion of parking slots with nearby objects like trees. In such parking slots vehicles are often not properly detected in first algorithm step.

## 4.2. Parking Slot Occupancy Classification Results

For the parking slot occupancy classification standard metrics are used:

- TP occupied parking slot classified as occupied,
- TN vacant parking slot classified as vacant,
- FP vacant parking slot classified as occupied,
- FN occupied parking slot classified as vacant,
- Accuracy = (TP + TN)/(TP + TN + FP + FN),
- AUC area under the Receiver Operating Characteristic (ROC).

To make a fair comparison with recent approaches, we followed guidelines from PKLot dataset de Almeida et al. (2015) and considered 50% of the images available in the subsets UFPR04, UFPR05, and PUCPR for training and 50% for testing. We performed single parking lot training and multiple parking lot testing since we are mostly interested in the generalization power of classifiers. This results in three different classifiers where each classifier is evaluated on three different test datasets. The number of images containing occupied or vacant parking slots can be found in Table 4. It can be noticed that the PUCPR parking lot has a higher number of images in comparison with UFPR04 and

Sample	TP	$\mathbf{FP}$	FN	Precision	Recall	# of
size				[%]	[%]	parking
			<u> </u>	1		slots
200%	20	6	Camera .	22.86	80.56	26
50%	29	4	5	88.57	86.11	36
80%	26	4	10	86.67	72.22	36
100%	30	5	6	85.71	83.33	36
10070			Camera :	2	00100	
30%	9	1	2	90.00	81.82	11
50%	8	3	3	72.73	72.73	11
80%	10	0	1	100.00	90.91	11
100%	11	0	0	100.00	100.00	11
			Camera 3	3		
30%	22	4	4	84.61	84.61	26
50%	24	2	2	92.31	92.31	26
80%	23	3	3	88.46	88.46	26
100%	25	1	1	96.15	96.15	26
			Camera 4	4		
30%	36	3	5	92.31	87.80	41
50%	38	0	3	100.00	92.68	41
80%	38	1	3	97.44	92.68	41
100%	39	0	2	100.00	95.12	41
			Camera	5		
30%	46	0	7	100.00	86.79	53
50%	46	0	7	100.00	86.79	53
80%	49	0	4	100.00	92.45	53
100%	49	0	4	100.00	92.45	53
			Camera	6		
30%	44	1	9	97.78	83.02	53
50%	48	1	5	97.96	90.57	53
80%	48	0	5	100.00	90.57	53
100%	48	0	5	100.00	90.57	53
			Camera '	7		
30%	48	0	9	100.00	84.21	57
50%	49	0	8	100.00	85.96	57
80%	52	0	5	100.00	91.23	57
100%	52	0	5	100.00	91.23	57
Camera 8						
30%	52	0	4	100.00	92.86	56
50%	49	0	7	100.00	87.50	56
80%	51	0	5	100.00	91.07	56
100%	51	0	5	100.00	91.07	56
			Camera 9	9		
30%	29	0	3	100.00	90.63	32
50%	31	0	1	100.00	96.88	32
80%	30	0	$^{2}$ 26	100.00	93.75	32
100%	29	0	3	100.00	90.63	32

 ${\bf Table \ 3:} \ {\rm Results} \ {\rm of} \ {\rm parking} \ {\rm slot} \ {\rm detection} \ {\rm on} \ {\rm CNRPark-EXT} \ {\rm dataset}.$ 

UFPR05 since the corresponding camera covers a significantly higher number of parking slots. Also, subsets are imbalanced, vacant slots being majority class for UFPR04 and PUCPR parking lots, and occupied slots being majority class for UFPR05 parking lot. However, this imbalance is not significant and does not require any special approach to classifier training and testing.

Danking lot	Subset	# of parking slots		
Parking lot	Subset	Occupied	Vacant	Total
	Train	23,050	29,871	52,921
UFPR04	Test	23,075	29,847	52,922
	Total	46,125	59,718	105,843
UFPR05	Train	48,967	33,925	82,892
	Test	48,459	34,434	82,893
	Total	97,426	68,359	165,785
PUCPR	Train	96,736	115,375	212,111
	Test	97,493	114,619	212,112
	Total	194,229	229,994	424,223

Table 4: Summary of created PKLot subsets.

The comparison regarding classification accuracy is made with CarNet Nurullayev & Lee (2019) and mAlexNet Amato et al. (2017). The obtained results are presented in Table 5 for different combinations of training and testing subsets. The best result is shown in bold for each considered case. The obtained results show that our proposed deep classifier obtains the best result in seven out of nine possible subset combinations. This strongly indicates that the proposed approach to parking occupancy classifier learning results in high generalization power. This is especially pronounced when learning on the PUCPR subset and testing on the UFPR04 subset where significantly different camera perspectives are present. In this case, our approach has 4% higher accuracy than state-ofthe-art approach CarNet Nurullayev & Lee (2019).

The proposed classifier efficiency on the PKLot dataset in terms of AUC is reported in Table 6. The comparison is performed with CarNet Nurullayev & Lee (2019), mAlexNet Amato et al. (2017) and the approach based on integral

Mathad	Tusining subset	Testing accuracy		
Method	Training subse	UFPR04	UFPR05	PUCPR
	UFPR04	99.98%	95.47%	99.19%
Ours	UFPR05	95.29%	99.92%	98.08%
	PUCPR	$\mathbf{98.62\%}$	98.60%	99.93%
	UFPR04	95.60%	97.60%	98.30%
CarNet	UFPR05	95.20%	97.50%	98.40%
	PUCPR	94.40%	97.70%	98.80%
mAlexNet	UFPR04	99.54%	93.29%	98.27%
	UFPR05	93.69%	99.49%	92.72%
	PUCPR	98.03%	96.00%	99.90%

Table 5: Accuracy of parking slots classification on PKLot dataset.

channel features proposed in Ahrnbom et al. (2016). Again, our proposed approach achieves the best result in seven out of nine possible subset combinations and obtains AUC greater than 0.99 for each subset combination thus confirming its high generalization abilities.

In the case of the CNRPark-EXT dataset, we also followed the dataset split proposed in Amato et al. (2017) so we can directly compare our classifier with mAlexNet, AlexNet and partially with Nurullayev & Lee (2019). According to Table 7, our proposed classifier outperforms mAlexNet and AlexNet for each train/test subset combination by a large margin. For example, when training our classifier only on CNRPark dataset, which contains images from only two cameras and images captured during sunny days, it obtains significantly higher accuracy (more than 4%) than AlexNet when testing on CNRPark-EXT TEST dataset which contains 9 cameras and different weather conditions. When the training subset contains CNRPark dataset and images from CNRPark-EXT train C1-C8, then accuracy on CNRPark-EXT TEST dataset goes over 99% indicating that very high classifier accuracy can be obtained if the training subset contains different viewpoints and weather conditions. The proposed classifier also outperforms state-of-the-art CarNet on CNRPark-EXT dataset as shown in Table 7. More detailed experiments were conducted related to viewpoint changes and weather conditions like in Amato et al. (2017). The classifier was trained on images taken with a single camera (C1 and C8) and tested against images of all other cameras in the CNRPark-EXT dataset. The obtained results regarding viewpoint generalization of the built classifier are reported in Table 8. In almost all cases proposed classifier outperforms mAleXNet, achieving accuracy over 95% in all cases except in case of training on images taken by camera C8 and testing on images taken by camera C1 since these are two cameras with significantly different viewpoints (C1 is a side view of a parking lot and C8 is a front view of a parking lot). However, the difference in the obtained accuracy of the proposed classifier on the same test camera does not differ more than 1% when training on C1 or C8 which is not true for mAlexNet. It can be concluded that the proposed classifier has better viewpoint robustness.

Similar behavior can be observed when training the classifier on images captured during particular weather condition (sunny, rainy, or overcast) and testing against images captured during remaining weather conditions. The obtained results regarding weather generalization are shown in Table 9. It can be noticed that the proposed classifier significantly outperforms mAlexNet and obtains accuracy over 98% in all cases. Certainly, the proposed classifier is very robust to the weather conditions changes as well.

#### 5. Conclusions

In this paper, vision-based algorithm called APSD-OC is proposed. APSD-OC automatically detects parking slots and classifies each parking slot as occupied or free. As such it contains two main parts. In the first part, the locations of parking slots in the input image are determined. Hereby, vehicles are detected in a series of input images using YOLOv5. After that, the detections centers are transformed to bird's eye view using a homography matrix which is obtained by a CNN. The transformed detections are clustered using DBSCAN algorithm. The resulting centers are filtered and are projected back to the original view

where they correspond to the locations of parking slots. In the second part of the algorithm, the detected parking slots are classified as occupied or vacant using a ResNet34 based classifier.

The proposed APSD-OC algorithm is evaluated on two publicly available datasets: PKLot and CNRPark-EXT. The evaluation is carried on by analyzing parking slot detection and parking slot occupancy classification performance of the proposed APSD-OC algorithm. The obtained results show that detection precision and recall goes well beyond 90% as more input images are used in the detection procedure on both datasets. The proposed algorithm is robust to the presence of parking violations and passing vehicles which are often appearing in images of the PKLot dataset. The trained deep classifier shows high accuracy, obtaining AUC over 0.99 for different combinations of training and testing subsets in the case of PKLot dataset thus significantly outperforming CarNet and mAlexNet. The same can be observed in the case of CNRPark-EXT dataset, proving high classifier robustness to viewpoint change and weather conditions.

Our future work will include estimation of the number of regular parking slots from the shape of the vehicle detections distribution. Apart from that, we will try to utilize the fact that parking slots have a certain spatial relationship to obtain even more accurate parking slot detection.

## References

- Abbas, S. A., & Zisserman, A. (2019). A Geometric Approach to Obtain a Bird's Eye View From an Image. In 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW) (pp. 4095– 4104). IEEE. URL: https://ieeexplore.ieee.org/document/9022574/. doi:10.1109/ICCVW.2019.00504. arXiv:1905.02231.
- Acharya, D., Yan, W., & Khoshelham, K. (2018). Real-time image-based parking occupancy detection using deep learning. CEUR Workshop Proceedings, 2087, 33–40.

- Ahrnbom, M., Astrom, K., & Nilsson, M. (2016). Fast Classification of Empty and Occupied Parking Spaces Using Integral Channel Features. In 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (pp. 1609–1615). IEEE. URL: http://ieeexplore.ieee.org/ document/7789690/. doi:10.1109/CVPRW.2016.200.
- Al-Kharusi, H., & Al-Bahadly, I. (2014). Intelligent Parking Management System Based on Image Processing. World Journal of Engineering and Technology, 02, 55-67. URL: http://www.scirp.org/journal/doi.aspx? DOI=10.4236/wjet.2014.22006. doi:10.4236/wjet.2014.22006.
- Alam, M., Moroni, D., Pieri, G., Tampucci, M., Gomes, M., Fonseca, J., Ferreira, J., & Leone, G. R. (2018). Real-Time Smart Parking Systems Integration in Distributed ITS for Smart Cities. *Journal of Advanced Transportation*, 2018, 1–13. URL: https://www.hindawi.com/journals/jat/2018/1485652/. doi:10.1155/2018/1485652.
- de Almeida, P. R., Oliveira, L. S., Britto, A. S., Silva, E. J., & Koerich, A. L. (2015). PKLot – A robust dataset for parking lot classification. *Expert Systems with Applications*, 42, 4937– 4949. URL: http://dx.doi.org/10.1016/j.eswa.2015.02.009https:// linkinghub.elsevier.com/retrieve/pii/S0957417415001086. doi:10. 1016/j.eswa.2015.02.009.
- de Almeida, P. R. L., Alves, J. H., Parpinelli, R. S., & Barddal, J. P. (2022). A systematic review on computer vision-based parking lot management applied on public datasets. *Expert Systems with Applications*, 198, 116731. URL: https://www.sciencedirect.com/science/article/ pii/S0957417422002032. doi:https://doi.org/10.1016/j.eswa.2022. 116731.
- Amato, G., Carrara, F., Falchi, F., Gennaro, C., Meghini, C., & Vairo, C. (2017). Deep learning for decentralized parking lot occupancy detection. Expert Systems with Applications, 72, 327–334. URL: https://

linkinghub.elsevier.com/retrieve/pii/S095741741630598X. doi:10. 1016/j.eswa.2016.10.055.

- Cai, B. Y., Alvarez, R., Sit, M., Duarte, F., & Ratti, C. (2019). Deep Learning-Based Video System for Accurate and Real-Time Parking Measurement. *IEEE Internet of Things Journal*, 6, 7693-7701. URL: http: //arxiv.org/abs/1902.07401http://dx.doi.org/10.1109/jiot.
  2019.2902887https://ieeexplore.ieee.org/document/8660429/. doi:10.1109/JIOT.2019.2902887. arXiv:1902.07401.
- Chen, L.-C., Sheu, R.-K., Peng, W.-Y., Wu, J.-H., & Tseng, C.-H. (2020). Video-Based Parking Occupancy Detection for Smart Control System. Applied Sciences, 10, 1079. URL: https://www.mdpi.com/2076-3417/10/3/ 1079. doi:10.3390/app10031079.
- Chen, M., & Chang, T. (2011). A parking guidance and information system based on wireless sensor network. 2011 IEEE International Conference on Information and Automation, ICIA 2011, (pp. 601–605). doi:10.1109/ ICINFA.2011.5949065.
- Coleiro, A., Scerri, D., & Briffa, I. (2020). Car parking detection in a typical village core street using public camera feeds. In 2020 IEEE 10th International Conference on Consumer Electronics (ICCE-Berlin) (pp. 1–6). doi:10.1109/ICCE-Berlin50680.2020.9352169.
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining KDD'96 (pp. 226–231). AAAI Press.
- Guo, W., Ji, Y., Wang, W., Blythe, P., & Tang, D. (2014). Understanding drivers' perspective on parking guidance information. *IET Intelligent Transport Systems*, 8, 398-406. URL: https://digital-library. theiet.org/content/journals/10.1049/iet-its.2012.0167. doi:10.1049/iet-its.2012.0167.

- Huang, C. C., & Wang, S. J. (2010). A hierarchical bayesian generation framework for vacant parking space detection. *IEEE Transactions on Circuits* and Systems for Video Technology, 20, 1770–1785. doi:10.1109/TCSVT. 2010.2087510.
- Jocher, G., Stoken, A., Borovec, J., NanoCode012, Chaurasia, A., TaoXie, Changyu, L., V, A., Laughing, tkianai, yxNONG, Hogan, A., lorenzomammana, AlexWang1900, Hajek, J., Diaconu, L., Marc, Kwon, Y., oleg, wanghaoyang0106, Defretin, Y., Lohia, A., ml5ah, Milanko, B., Fineran, B., Khromov, D., Yiwei, D., Doug, Durgesh, & Ingham, F. (2021). ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations. doi:10.5281/zenodo.4679653.
- Ke, R., Zhuang, Y., Pu, Z., & Wang, Y. (2020). A Smart, Efficient, and Reliable Parking Surveillance System With Edge Artificial Intelligence on IoT Devices. *IEEE Transactions on Intelligent Transportation Systems*, (pp. 1–13). URL: https://ieeexplore.ieee.org/document/9061155/. doi:10.1109/TITS.2020.2984197. arXiv:2001.00269.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1 (pp. 1097–1105). Curran Associates Inc.
- Lin, T., Rivano, H., & Le Mouel, F. (2017). A Survey of Smart Parking Solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18, 3229– 3253. doi:10.1109/TITS.2017.2685143.
- Martin Nieto, R., Garcia-Martin, A., Hauptmann, A. G., & Martinez, J. M. (2019). Automatic Vacant Parking Places Management System Using Multicamera Vehicle Detection. *IEEE Transactions on Intelligent Trans*portation Systems, 20, 1069–1080. URL: https://ieeexplore.ieee.org/ document/8371300/. doi:10.1109/TITS.2018.2838128.

- Ng, C.-K., Cheong, S.-N., & Foo, Y.-L. (2020). Low Latency Deep Learning Based Parking Occupancy Detection By Exploiting Structural Similarity. In *Lecture Notes in Electrical Engineering* (pp. 247-256). volume 603. URL: http://link.springer.com/10.1007/978-981-15-0058-9\_ 24. doi:10.1007/978-981-15-0058-9\_24.
- Nurullayev, S., & Lee, S. W. (2019). Generalized parking occupancy analysis based on dilated convolutional neural network. *Sensors (Switzerland)*, 19. doi:10.3390/s19020277.
- Padmasiri, H., Madurawe, R., Abeysinghe, C., & Meedeniya, D. (2020). Automated Vehicle Parking Occupancy Detection in Real-Time. In 2020 Moratuwa Engineering Research Conference (MERCon) August (pp. 1-6). IEEE. URL: https://ieeexplore.ieee.org/document/9185199/. doi:10.1109/MERCon50084.2020.9185199.
- Patel, R., & Meduri, P. (2020). Car detection based algorithm for automatic parking space detection. In 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1418–1423). doi:10. 1109/ICMLA51294.2020.00220.
- Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (pp. 512– 519). IEEE. URL: https://ieeexplore.ieee.org/document/6910029. doi:10.1109/CVPRW.2014.131.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv:1804.02767.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In Neural Information Processing Systems. URL: http://arxiv.org/abs/1506.01497. arXiv:1506.01497.

- SENSIT (). Wireless smart parking sensors for occupancy detection. URL: https://www.nedapidentification.com/products/sensit/.
- Sifuentes, E., Casas, O., & Pallas-Areny, R. (2011). Wireless Magnetic Sensor Node for Vehicle Detection With Optical Wake-Up. *IEEE Sensors Journal*, 11, 1669–1676. URL: http://ieeexplore.ieee.org/document/5680571/. doi:10.1109/JSEN.2010.2103937.
- Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, (pp. 1–14). URL: http://arxiv.org/abs/1409.1556. arXiv:1409.1556.
- Smith, L. N. (2015). No More Pesky Learning Rate Guessing Games. CoRR, abs/1506.0. URL: http://arxiv.org/abs/1506.01186. arXiv:1506.01186.
- Smith, L. N. (2018). A disciplined approach to neural network hyperparameters: Part 1 - learning rate, batch size, momentum, and weight decay. CoRR, abs/1803.0. URL: http://arxiv.org/abs/1803.09820. arXiv:1803.09820.
- Smith, L. N., & Topin, N. (2017). Super-Convergence: Very Fast Training of Residual Networks Using Large Learning Rates. CoRR, abs/1708.0. URL: http://arxiv.org/abs/1708.07120. arXiv:1708.07120.
- Tschentscher, M., Koch, C., König, M., Salmen, J., & Schlipsing, M. (2015). Scalable real-time parking lot classification: An evaluation of image features and supervised learning algorithms. In 2015 International Joint Conference on Neural Networks (IJCNN) (pp. 1–8). doi:10.1109/IJCNN.2015. 7280319.
- Xie, H., Wu, Q., Chen, B., Chen, Y., & Hong, S. (2015). Vehicle Detection in Open Parks Using a Convolutional Neural Network. In 2015 Sixth International Conference on Intelligent Systems Design and Engineering Appli-

cations (ISDEA) (pp. 927-930). IEEE. URL: http://ieeexplore.ieee. org/document/7462769/. doi:10.1109/ISDEA.2015.233.

- Zhu, Y., Ye, X., Chen, J., Yan, X., & Wang, T. (2020). Impact of Cruising for Parking on Travel Time of Traffic Flow. Sustainability, 12, 3079. URL: https://www.mdpi.com/2071-1050/12/8/3079. doi:10.3390/ su12083079.
- Zusheng Zhang, Ming Tao, & Huaqiang Yuan (2015). A Parking Occupancy Detection Algorithm Based on AMR Sensor. *IEEE Sensors Journal*, 15, 1261– 1269. URL: http://ieeexplore.ieee.org/document/6919252/. doi:10. 1109/JSEN.2014.2362122.

Method	Training subset	Testing subset	AUC
Ours	UFPR04	UFPR04	0.9999
CarNet	UFPR04	UFPR04	0.9790
ICF+LR	UFPR04	UFPR04	0.9994
ICF+SVM	UFPR04	UFPR04	0.9996
PKLot	UFPR04	UFPR04	0.9999
Ours	UFPR04	UFPR05	0.9989
CarNet	UFPR04	UFPR05	0.9935
ICF+LR	UFPR04	UFPR05	0.9928
ICF+SVM	UFPR04	UFPR05	0.9772
PKLot	UFPR04	UFPR05	0.9595
Ours	UFPR04	PUCPR	0.9995
CarNet	UFPR04	PUCPR	0.9982
ICF+LR	UFPR04	PUCPR	0.9881
ICF+SVM	UFPR04	PUCPR	0.9569
PKLot	UFPR04	PUCPR	0.9713
Ours	UFPR05	UFPR04	0.9943
CarNet	UFPR05	UFPR04	0.9796
ICF+LR	UFPR05	UFPR04	0.9963
ICF+SVM	UFPR05	UFPR04	0.9943
PKLot	UFPR05	UFPR04	0.9533
Ours	UFPR05	UFPR05	0.9999
CarNet	UFPR05	UFPR05	0.9989
ICF+LR	UFPR05	UFPR05	0.9987
ICF+SVM	UFPR05	UFPR05	0.9988
PKLot	UFPR05	UFPR05	0.9995
Ours	UFPR05	PUCPR	0.9978
CarNet	UFPR05	PUCPR	0.9791
ICF+LR	UFPR05	PUCPR	0.9779
ICF+SVM	UFPR05	PUCPR	0.9405
PKLot	UFPR05	PUCPR	0.9761
Ours	PUCPR	UFPR04	0.9985
CarNet	PUCPR	UFPR04	0.9845
ICF+LR	PUCPR	UFPR04	0.9829
ICF+SVM	PUCPR	UFPR04	0.9843
PKLot	PUCPR	UFPR04	0.9589
Ours	PUCPR	UFPR05	0.9981
CarNet	PUCPR	UFPR05	0.9938
ICF+LR	PUCPR	UFPR05	0.9457
ICF+SVM	PUCPR	UFPR05	0.9401
PKLot	PUCPR	UFPR05	0.9152
Ours	PUCPR	PUCPR	0.9998
CarNet	PUCPR	PUCPR	0.9986
ICF+LR	PUCPR	PUCPR	0.9994
ICF+SVM	PUCPR	PUCPR	0.9994
PKLot	PUCPR	PUCPR	0.9999

 ${\bf Table \ 6:} \ {\rm Results \ of \ parking \ slot \ occypancy \ classification \ on \ PKLot \ dataset.}$ 

Method	Training	Testing	Testing	AUC
	$\mathbf{subset}$	subset	accuracy	
Ours		CNDDarl. EVT	97.66%	0.9969
mAlexNet	CNRPark	UNRPARK-EA I	93.52%	0.9838
AlexNet		1651	93.63%	0.9877
Ours	CNDDark   EVT	CNDDarls EVT	99.34%	0.9994
mAlexNet	TDAIN CL CR	UNRPARK-EA I	95.88%	0.9937
AlexNet	TRAIN CI-Co	1651	96.85%	0.9957
Ours			99.67%	0.9981
mAlexNet	CNRPark+EXT	CNRPark-EXT	97.71%	0.9967
AlexNet	TRAIN	TEST	98.00%	0.9974
CarNet			98.11%	N/A

 Table 7: Results of parking slot occupancy classification on CNRPark-EXT dataset - parking lot generalization.

The state of the set	To attack and and	Testing accuracy		
Training subset	Testing subset	Ours	$\mathbf{mAlexNet}$	
	CNRPark-EXT C1	-	-	
	CNRPark-EXT C2 $$	99.19%	94.85%	
	CNRPark-EXT C3	$\mathbf{97.28\%}$	93.11%	
	CNRPark-EXT C4	97.89%	96.00%	
CNRPark-EXT C1	CNRPark-EXT C5	97.60%	95.91%	
	CNRPark-EXT C6	97.25%	95.61%	
	CNRPark-EXT C7 <b>96.99%</b>		91.43%	
	CNRPark-EXT C8	97.43%	94.61%	
	CNRPark-EXT C9	95.75%	90.96%	
	CNRPark-EXT C1	92.79%	92.39%	
	CNRPark-EXT C2 $$	99.99%	94.51%	
	CNRPark-EXT C3	97.01%	93.66%	
	CNRPark-EXT C4	98.29%	97.53%	
CNRPark-EXT C8	CNRPark-EXT C5	97.49%	$\mathbf{97.93\%}$	
	CNRPark-EXT C6	NRPark-EXT C6 96.72%		
	CNRPark-EXT C7	95.55%	93.53%	
	CNRPark-EXT C8	-	-	
	CNRPark-EXT C9	96.47%	94.65%	

 Table 8: Results of parking slot occupancy classification on CNRPark-EXT dataset - view-point generalization.

Training subset	Testing subset	Te	sting accuracy
Training subset	Testing subset	Ours	$\mathbf{mAlexNet}$
CNRPark-EXT	OVERCAST	99.69%	97.80%
SUNNY	RAINY	99.02%	95.79%
CNRPark-EXT	SUNNY	98.95%	91.63%
OVERCAST	RAINY	98.11%	94.68%
CNRPark-EXT	SUNNY	98.54%	93.53%
RAINY	OVERCAST	99.39%	98.27%

**Table 9:** Results of parking slot occupancy classification on CNRPark-EXT dataset - weathergeneralization.