A Deep Learning based Unoccupied Parking Space Detection Method for City Lots

Hamid Reza Tohidypour[1], Yixiao Wang[1], Panos Nasiopoulos[1], and Mahsa T. Pourazad[1,2]
[1]Dept. of Electrical and Computer Engineering, The University of British Columbia, Vancouver, Canada
[2]TELUS Communications Inc, Vancouver, Canada
e-mail: {htohidyp, yixiaow, panosn, pourazad}@ece.ubc.ca, mahsa.pourazad@gmail.com

***Abstract*— Nowadays, finding a vacant parking space in populated metropolitan cities is a challenging task, leading to serious traffic congestion with environmental and productivity ramifications. Although many different systems have been proposed and tested for unsupervised parking lot space detection over the years, they have been proven to be either impractical or costly to maintain. In this paper, we propose a deep learning based approach that uses the video captured by a vehicle mounted camera to accurately detect and count the number of available parking spaces in real-time. Our system achieves an impressive average detection accuracy of 90.59% for unoccupied spaces and 95.66% for the occupied spaces.**

Keywords- deep learning; city lots; object recognition; YOLOv4.

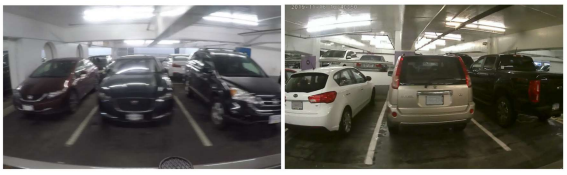
I. INTRODUCTION

Properly managing parking lots and displaying available parking information to the drivers before entering the parking lot has been a challenge. Many different systems have been employed over the years for unsupervised parking space detection for open lots, including counting the number of cars entering and exiting the parking lot or using a network of ground- or ceiling-mounted occupancy sensors [1]. Although the latter is a welcome solution, it turns out to be a costly one as sensor maintenance proved to be an expensive proposition. In this paper, we propose a deep learning based approach that uses the video captured by a vehicle mounted camera to accurately detect and count the number of available parking spots in real-time.

The rest of this paper is organized as follows. In Section II we give a detailed description of our data collection, the labeling approach, and the deep learning network architecture that we used. Section III presents the performance evaluation of our method and discusses the results. Finally, Section IV concludes our paper.

II. OUR PROPOSED METHOD

Our objective is to design a deep learning based real-time occupancy detection system, using a video feed from security and parking inspection vehicle-mounted cameras. To this end, our first task was to capture a comprehensive and representative dataset to use for training and validating our deep learning network. The second task involves the design of an effective labelling scheme that will allow our network to accurately detect the available parking spots in a parking lot.



(a) (b)

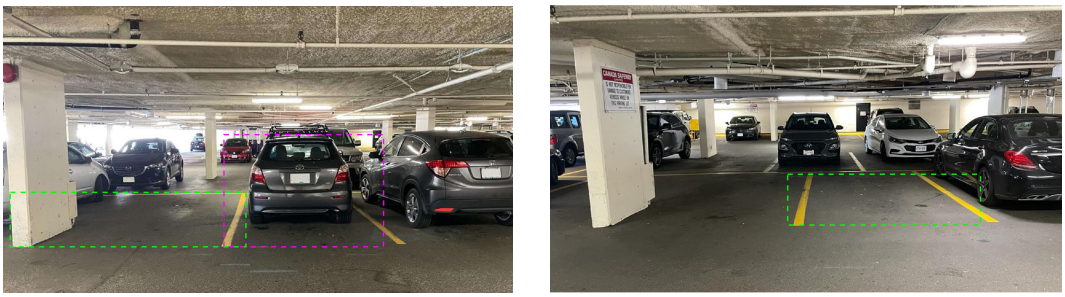
Figure 1. a) image captured by GoPro Hero4 vs b) image captured by AKASO EK7000.

A. Data Collection

In generating an appropriate dataset, we first had to determine what type of commercial camera may be used that meets our requirements such as frame rate and resolution. We compared two cameras namely the AKASO EK7000 and the GoPro Hero4 action cameras. It is important to note that the cameras are mounted on top of a moving vehicle. Figure 1 provides a visual comparison between frames captured by the GoPro Hero4 and AKASO EK7000 cameras. As it can be seen, motion blurring is observed in the GoPro Hero4 footage, with AKASO EK700 yielding consistently better clarity and less motion blurring for our application.

In our implementation, an AKASO EK700 camera was mounted on top of a moving vehicle. Since the angle and distance of the camera from the parking spaces may affect the shape of the objects of interest in the frame, we tried to cover all practical scenarios of distances and angles. The former was achieved by capturing data from multiple parking lots around the city while the latter by using two vehicles of different sizes, a sedan and an SUV. In addition, we tried to cover a huge variety of parking scenarios, with different types of cars, cars that were not "properly" parked and parking lots with different ways of indicating parking spaces. This huge variety of data aims at helping our network to, on one hand, avoid overfitting and on the other to accurately detect the availability of parking spots in unique and unknown situations. A large number of videos was captured and converted to single frames for labelling and training the network. To avoid overfitting from using similar frames, we chose only the frames with some significant difference. A total of 4000 unique frames was used for training our network.

 ISBN: 978-1-61208-977-5



(a) (b)

Figure 2. In both frames the car travels from right to left. a) Only the fully visible occupied space is labeled with magenta dashed rectangles and one unoccupied parking space, which is labeled with the green dashed rectangle. b) A frame with two unoccupied spots and one occupied space, which is not fully visible to the camera.

B. *Labelling*

Our main objective is to design a deep learning based approach that detects unoccupied parking spots and counts the uniquely available parking spaces. Assigning the proper label classes is one of the challenges and a key factor in determining the success of our proposed approach. It is worth noting here that the only difference between unoccupied and occupied parking spaces is the presence of vehicles. For this reason, in order to facilitate the training process, we decided to add a class for occupied parking spots as well, thus assigning each frame labels and bounding boxes for occupied and unoccupied parking spaces.

In the case of object recognition, it is common practice to label all the objects of interest that are present in the scene. In our implementation, labeling of parking spaces has to follow a predetermined rule that will allow the deep learning network to learn to track the spots from frame to frame and thus accurately count the number of unoccupied spots. It is worth noting that a key feature of a parking space is the presence of two lines that indicate the borders of the spot. On the other hand, the difference between a an unoccupied and an occupied parking space is the presence of a car in the latter case. Given the above observations, we decided to have two labels, occupied and unoccupied parking spaces. Since in this project, we want to count the number of unique unoccupied parking spots, in every frame we considered to label only the first unoccupied spot we encounter, if it exists. In the case of occupied spots, we label in each frame all the paces that we clearly see, i.e., almost perpendicular to the camera and we see the car. Figure 2 shows an example of two labeled frames. In both frames the car is traveling from right to left. In Figure 2 (a) we observe two occupied spaces. In this case, only the occupied space that is fully visible to the camera is labeled with the bounding box shown by a magenta dotted line and the unoccupied space labeled by a bounding box shown with a green dotted line. Figure 2 (b) depicts one occupied space, which is not fully visible to camera, and two unoccupied spaces. In this case, only the first unoccupied space is labeled with the bounding box (green dotted line). As the vehicle-mounted cameras move forward at an expected speed of vehicles driving in a parking lot (15kmph-20kmph), our counting algorithm described in detail is Section III considers only the next unique frame and thus unoccupied space.

C. *Deep Learning Network*

Our goal here is to use a Deep Learning network architecture that is designed for object detection and offers real-time performance. Although there are many widely used neural networks for object recognition, ranging from Faster R-CNN to Mask-RCNN and YOLO (You Only Look Once), the performance of the 4th generation of the latter seems to meet all our requirements [2] [3] [4].

YOLO is a family of one-stage object detectors that are accurate and fast and their architecture allows them to be trained much easier than the other state-of-the-art networks.

YOLOv4 introduces a set of advanced features, such as spatial attention modules, a new activation function, new data augmentation schemes and Self-Adversarial Training, which significantly improve the performance of the network compared to previous versions.

Figure 3 depicts a comparison of YOLOv4 with other state-of-the-art object detection networks, clearly showing that it is much more accurate than YOLOv3 and much faster than EfficientDet with comparable performance [4].

To make sure that we chose the correct network architecture, we trained both the EfficientDet and YOLOv4 using our dataset. YOLOv4 started converging much faster, using only one CPU and a smaller mini-batch size, while EfficientDet could not reach similar performance, mainly due to memory requirements and possible overfitting due to the limited size of our dataset. For these reasons, we decided to use YOLOv4 for our task.

III. PERFORMANCE AND EVALUATIONS

We trained YOLOv4 using a state-of-the-art computing cluster [5]. Our dataset consisted of a large number of video sequences, which were converted to 4000 unique frames that were used for training and validation (80% and 20%, respectively). Note that the suggested default settings of the YOLOv4 for batch size is 64 and the learning rate is 0.001. In addition, the recommended threshold for intersection over union for the mean average precision (mAP) is 0.5, while the resolution of input images is 608x608 pixels, i.e., inputs are downscaled to that size by YOLOv4. When we used these

 ISBN: 978-1-61208-977-5

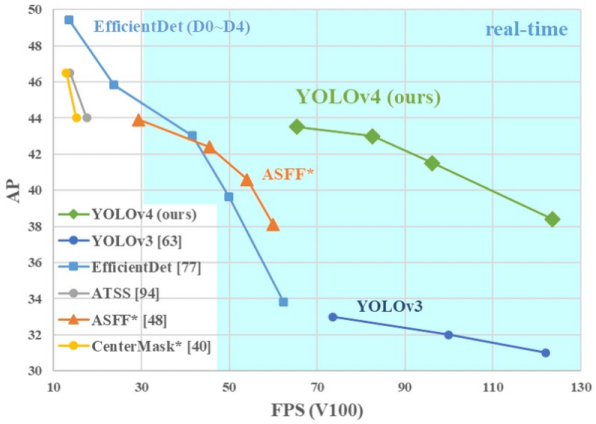


Figure 3. YOLOv4 is much more accurate than YOLOv3 and much faster than EfficientDet with comparable performance [2].

recommended settings for training this network, the mAP reached 94% at the early stages of training, gradually dropping to mAP of 90.7%, showing that the hyperparameters are not well optimized for our task. To address this issue, we investigated using different hyperparameters for YOLOv4, in which we reduced the learning rate to 0.0001 while increasing the batch size to 128. Total training time was approximately 48 hours. Figure 4 shows the loss vs iteration number for the training stage and the mAP vs iteration number for these new settings. As can be seen in the figure, the loss plateaus during training, eventually leading to 96% accuracy for the best weight sets, which we use for our testing model.

In order to test our model, we used a MacBook Pro laptop with M1 chip with 8-core CPU, 8-core GPU and 16-core Neural Engine. The laptop proved to be very powerful and run our model in real-time using a webcam feed, performing real-time predictions at 30fps.

Our model's prediction accuracy is measured based on its ability to classify the unoccupied and the occupied parking spots classes. In addition, a counting algorithm was developed that used the occupancy detection model to count the number of distinct parking stalls.

Several testing conditions related with the parking lot may affect the occupancy confidence levels of our detection model. To eliminate many unknown variables, we tried to design our model to be as robust as possible to such conditions. As discussed before, we consider vehicle speeds ranging from 15km/h to 20km/h, a range acceptable for security and inspection cars traveling in a parking lot. Our labeling process and the architecture of our network allow us to account for this range of speeds and the fact that the speed of the vehicle most likely will vary during different rounds in the parking lot. As the camera is mounted on the vehicle, the speed at which it moves through the parking lot determines how many frames are captured per area/parking space. The above is an accurate statement, as the relative distance of the camera from the parking space stays almost the same and our tests showed that small variations do not significantly affect the number of frames per area.

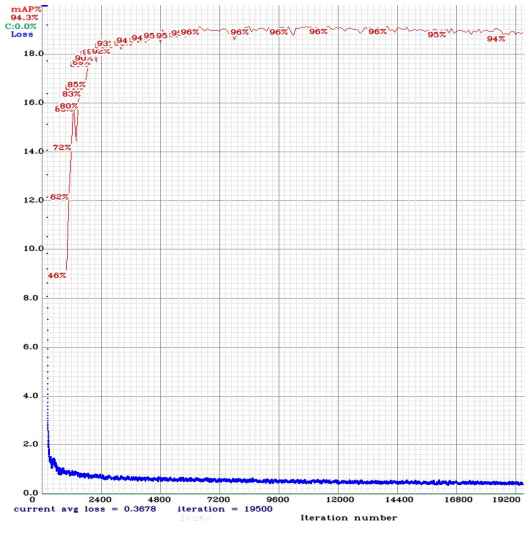


Figure 4. The loss and mAP@0.5 plot for YOLOv4. Total training time was approximately 48 hours. The mAP@0.5 reaches 94.3%, with the best mAP@0.5 for 96%. The average loss value floats around 0.3678.

Prediction accuracy is also determined by the prediction threshold. With each prediction performed by the model, the model attaches a confidence level which indicates how likely a class object exists within the bounding box. Using a threshold (Thr) level allow us to only consider predictions with a confidence level higher or equal to the threshold level; those are the ones that will be displayed with bounding boxes.

Performance evaluation of our deep learning trained model is based primarily on its accuracy in detecting unoccupied and occupied parking spaces, using several testing scenarios. The scenarios of main interest consist of unoccupied and occupied regular and angled spaces, reserved parking spaces and handicap spaces. Exceptional cases include the presence of a wall on one side of a parking spot instead of a parking line, cars parked over parking lines, obscure parking lines due to erosion of paint, corner spots, poorly visible parking spots due to rain, and parking spots using special color lines instead of the regular white parking lines.

Our test results are summarized in Table I, which shows the detection accuracy (AP) of the two classes, occupied and unoccupied. Real-time predictions were done on the laptop positioned inside the car that was receiving a 30fps video feed from a camera mounted on the top of the vehicle. We observe that for confidence threshold 30% the probability of predicting an unoccupied parking space is 90.59% while for the probability of predicting occupied parking spaces is 95.66%, while for confidence threshold of 50% the corresponding predictions precisions are changed to 88.86% for the unoccupied spaces and 95.52% for the occupied spaces. This means that our network mostly assigns probabilities greater than 50% to the detected classes. However, in order to avoid missing any unoccupied spaces, we decided to use the confidence threshold of 30%.

The fact that occupied parking spaces may be identified by both the presence of lines and cars, which in terms of a neural network means more features, leads to a slightly higher prediction accuracy for the occupied parking spaces than the unoccupied ones.

Our unoccupied parking space counting algorithm is built on the premise that our model always detects the first parking space and only one unoccupied space in each frame. The car speed is used for determining which frame will be considered next, making sure that we will "see" a new parking space introduced in the next frame. For instance, at 15 km/h the car will cover 4 meters in 1 second. At the same time, our camera captures 30fps. Since our measurements have shown that a parking space in a city parking lot is

TABLE I. TEST RESULTS FOR REAL TIME SCENARIO.

Parking Status	Average Precision	
	Thr = 30%	Thr = 50%
Unoccupied	90.59%	88.86%
Occupied	95.66%	95.52%

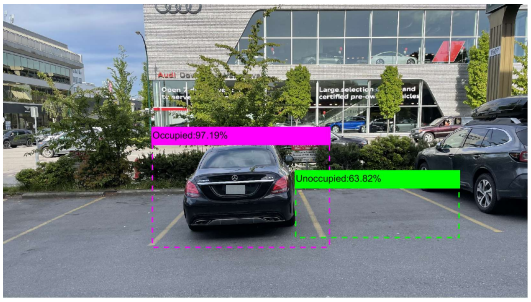


Figure 5. An example of test images with the predicted bounding boxes, labels, and the probabilities assigned to them.

approximately 3.5 meters wide, we concluded that we should process every 22nd frame so we ensure that we have a new parking space introduced to the frame. This calculation is based on the assumption that the parking lot is normal size and the car camera is facing the side of the passenger, which is closer to the traveling car. Such a scenario is shown in Figure 5. In case that the parking lot is larger in size or we want to position the camera to view the cars on the driver's side, thus increasing the distance between the camera and the parking spaces, the algorithm may be adjusted to consider a different number of frames. For example, if the distance is double as shown in Figure 6, a new parking space will be introduced every 16 frames. This example shows the challenging case of a rainy scene with wet asphalt. As it can be seen in Figure 6, our model was able to correctly detect the parking space and the occupied space next to it. The probability that was assigned to the unoccupied space was 37.13%, which is less than the probability that was assigned to unoccupied space in Figure 4, due to the weather conditions. However, this shows that assigning confidence threshold of 30% makes our model robust to the ambient conditions.

IV. CONCLUSIONS

In this paper, we introduced a deep learning based detection method for unoccupied parking spaces in a city parking lot. We developed a unique labeling scheme, which allowed us to detect one new unoccupied parking spot in a

Figure 6. An example test image for the case that distance is two times larger than the case of Figure 5.

 ISBN: 978-1-61208-977-5

frame and thus accurately count the total number of vacant spaces in the parking lot. We used the YOLOv4 network and we achieved a detection accuracy of 90.59% for unoccupied spaces and 95.66% for the occupied spaces. As our training and validation dataset was only 4000 frames due to privacy issues, we are confident that this network can achieve even higher accuracy with a larger dataset and the same labeling scheme.

ACKNOWLEDGMENT

We would like to acknowledge the contribution of C. Yao, M. I. Islam, R. Chu, S. Ly, and B. Wang for data collection and initial work on the subject. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC – PG 11R12450), and TELUS (PG 11R10321). This research was enabled in part by support provided by WestGrid (www.westgrid.ca) and Compute Canada (www.computecanada.ca).

REFERENCES

[1] V. Paidi, H. Fleyeh, J. Håkansson, and R. G. Nyberg, "Smart parking sensors, technologies and applications for open parking lots: a review," IET Intelligent Transport Systems, vol. 12, no. 8, pp. 735 – 741, May 2018.

[2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," The 28th International Conference on Neural Information Processing Systems, vol. 1, pp. 91-99, December 2015.

[4] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," Proc. IEEE International Conference on Computer Vision, IEEE Press, pp. 2980-2988, Oct. 2017, doi: 10.1109/ICCV.2017.322.

[5] Compute Canada state-of-the-art advanced research computing network. Available from: https://www.computecanada.ca [retrieved: April 5, 2022].

 ISBN: 978-1-61208-977-5