YOLOv4 Street Parking Detection through Autonomous Vehicle Datasets

Kayla Ippongi
Johns Hopkins University
Email: kippong1@jhu.edu

***Abstract*—Finding parking in metropolitan areas remains a persisting and challenging task for any individual. With the imminent rise of level 5 autonomous vehicles, the task of finding street parking and how our current infrastructure might supervise these systems remains an open question. The argument that the need for street parking will diminish relies on the exhaustive adoption of shared autonomous vehicles rather than privatized. As future public support and adoption for autonomous vehicles remain undetermined, being able to identify vacant street parking is a worthwhile challenge to look into. This paper proposes a dataset to aid in streamlining street parking detection for autonomous vehicles. Three open datasets are utilized for camera frames - Argo AI, Waymo and NuScences presented in a variety of US urban cities with varying weather and obstacles. These datasets are annotated and then compared against two deep learning models, YOLO and Faster R-CNN to evaluate metrics.**

I. Introduction

The complexities and nuisances of parking in large cities and congested areas remain a withstanding problem. According to research, drivers in New York City spend an average of 107 hours per year searching for parking, amounting to $2,243 per driver in wasted fuel and costs [1].

There are many existing solutions being made to mitigate this widespread parking problem. The majority include the usage of IoT and sensor technologies, for example, companies such as Nwave [2] and SmartParking [3] uses in-ground sensors and overhead indicators in parking garages to collect data and send real-time information regarding parking space occupancy.

With the rise of autonomous vehicles, many wonder if the concept and need for parking will die out and how our current infrastructure will adapt to the rise of self-driving vehicles. Fully autonomous vehicles dismantle the need for close proximity parking as AVs can be sent to wait in areas with less congestion or cheaper fares [5]. Rather, these vehicles have several options - park in a designated parking lot, go home and park, park in a rural area/suburb outside the city, and lastly, cruise to avoid parking [4].

According to research comparing AVs and their parking strategies in a microsimulation model, cruising was the most cost-effective majority of the time. [4] However, this has the potential to create and exacerbate a congestion problem. In a traffic microsimulation model few as 4000 AVs injected into the streets of San Francisco would put cruising speeds below 2km/h [4]. This can be avoided if shared autonomous vehicles become more widely adopted, however, the future of AV's acceptance by the public and their shared implications still remains open-ended.

The goal of this paper is to add annotations to existing datasets that will aid in streamlining finding street parking for autonomous vehicles and determine if we can detect parking spaces with decent precision. This dataset is an aggregation of open datasets from Argo AI [7], Waymo [8] and NuScences [9]. The original datasets consist of several terabytes of information and this study will deal with a portion of each of them.

This paper is split into several sections. Section 2 outlines related work that has been previously been done regarding parking detection from the drivers perspective and their respective findings. Section 3 details the three datasets we will be utilized in the paper, along with guidelines for object annotation. Section 4 describes the approach and metrics we'll be using to compare the performance of each dataset. Section 5 and 6 detail the results and analysis of the experiments and room for future work.

II. Related Work

In a similar study, researchers utilized dash camera footage on a regular vehicle from a combination of municipal garage parking and street parking for early detection of vacant spots [6]. They created a loss function with respect to distance away from the spot to enable early detection. To the best of my knowledge, this seems to be the only paper analyzing parking from the driver's perspective. Their dataset consists of 5,800 clips of positive and negative annotations across 22 different locations, both indoor and outdoor and consists of a variety of parking types - parallel, perpendicular and on-street parking in diverse settings. In their case, a clip was annotated as positive if 20% of the clip captures an empty parking spot and negative otherwise. They compared their results against 2 models 3D CNN + LSTM and a 3D CNN and found that the latter had stronger classification results using their proposed loss function with about 85% accuracy.

Their results and analysis provide extremely important contributions towards parking detection, especially as they are able to define a loss function that allows for early detection. Some things that the paper did not touch upon were bounding boxes of the parking spaces and the validity of a vacant spot. A clip is labeled positive if an empty spot is present in the frame but does not permit localization. This paper will attempt to address these matters by utilizing bounding box annotation and

detection and follow rules that would define a valid parking space.

III. Data sets

This collection is an aggregation of open datasets from Argo AI [7], Waymo [8] and Nuscences [9]. These come from a variety of large US cities including Boston, San Francisco, Miami, and Pittsburgh. Additionally, they include diverse scenes from downtown, suburban, nighttime, construction areas, rain, etc. While these datasets offer 6+ camera angles for a single scene, this collection will focus on the 3 front-facing cameras - center, left and right to simulate looking ahead for a street parking spot. The variations of the same scene from different camera angles are one of the main reasons why these datasets were used and will also help our model to learn better.

These datasets come with a variety of different detected object classes including vehicles, pedestrians, road obstructions, signs, crosswalks, etc. While there is a substantial amount of metadata included in each of these sets, pre-trained model weights are not included, giving researchers the opportunity to build models to detect the offered classes. In an ideal experiment, we would like to use all metadata available to us, but for the scope of this paper and since we're only interested in one class, this paper will solely rely on the extracted frames and manual annotations to localize vacant parking spaces.

A total of 1,500 frames were manually annotated for each of the three datasets, with equal parts positive and negative annotations, resulting in a total of 4,500 frames. Each of the individual datasets comes proportioned with separate train and test scenes to ensure that there is no overlap, which we will be using here. A summary of the corresponding datasets, train and test sizes, and cities are detailed in Table I.

A. Annotation Guidelines

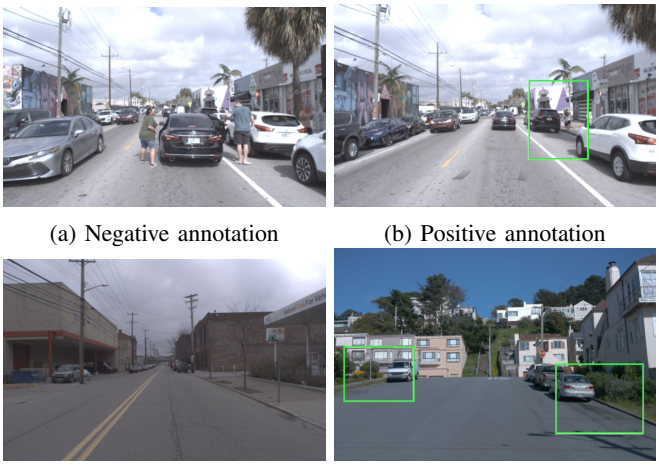
The choice of a positive annotation attempts to imitate the human behavior of looking for a parking spot. These follow the standard rules of street parking, and exclude parking in the general restricted areas:

- on crosswalks, in front or behind
- in front of entrances
- in front of stop signs or fire hydrants
- along sidewalks with yellow or red markings
- along streets with presence of no parking signs or bus signs

There a lot of parameters and variables that define a valid street parking space. Even to the human eye, identifying parking spots can be a challenging task to determine, especially street parking that is not explicitly labeled on the ground. The majority of the time, the validity of a spot is relative to the time or even day of the week. If a parking meter or parking stall lines are not present, the validity of a vacant spot can be difficult to assess. To minimize the constraints of this identification problem and enable consistency across the dataset, the following are variables that determine a positive annotation when parking is ambiguous.

- Presence of another car, in front or behind
- Presence of a raised curb that is unmarked in color
- Absence of any of the items in the restricted list above

To follow uniformity, positive annotations are marked with either the front or the back of another car at the midpoint of the frame. Several example annotations are given in Figure 1. In Figure 1a we see no annotations present because there are cars in all visible parking spaces. In figure 2c, while it seems to meet the standards for valid parking - visible and uncolored curb, no obstructions or entrances, it's still unidentifiable whether we are able to park here especially since we do not see any other cars parked. To figure out if this is a legal area to park, we would need the current time and readings of any signs within the block. An ideal model would include this information but for the scope of this project, this ambiguity is deemed as negative. Lastly, in Figures 2b and 2d, we see positive annotations based on the visible unmarked curb and presence of other cars.



(a) Negative annotation (b) Positive annotation

(c) Negative annotation (d) Positive annotation

Fig. 1: Examples of annotations

	Train	Test	Steps	Location
Argo	1,500	170	3,000	Miami & Pittsburgh
Nuscences	1,500	170	3,000	Boston
Waymo	1,500	170	3,000	San Francisco
Combined	4,500	510	6,000	Combined cities

TABLE I: Dataset summary

IV. Approach

The goal of this paper is to determine if we can localize and detect vacant street parking spaces with reasonable precision utilizing frames from autonomous vehicle data collection

Finding a vacant street parking space can be defined as a single object detection problem. There are many existing object detection techniques that can detect objects with reasonable precision. Many build upon the Convolutional Neural Network (CNN) framework which is comprised of several

layers - convolutional layer, nonlinear layer, pooling and fully connected layer [14]. Attempts to improve speed were made in R-CNN, Fast R-CNN, and Faster R-CNN.

R-CNN creates bounding box proposals using a selective search approach, which generates sub-segments of the input image and passes those proposals through a feature extractor. However, R-CNN can be considered slow due to its training pipeline that consists of 3 separate models. Fast R-CNN attempts to speed this up utilizing Region of Interest Pooling which attempts to reduce repeating computations for overlapping region proposals by sharing a single feature map. This way, rather than passing every single region proposal through the CNN we can pass the entire image through. Faster R-CNN attempts to mitigate the final bottleneck of the selection search for region proposals, which can be costly. Faster R-CNN diminishes the need for prior region proposal by using a CNN that performs both region proposals and classification [15]. The systems mentioned above all required some form of region proposals, which requires us to look at the same image thousands of times. YOLO [12], You Only Look Once, diminishes the requirement of region proposal by splitting the image into a grid and combining both bounding box regression and classification into a single model [16]

While each system has its own strengths and weaknesses, object detection for autonomous vehicles requires a careful balance between accuracy and speed. A highly accurate model with slow run time would be futile in accelerated environments while an inaccurate model with fast speeds could prove fatal. In a study that compared several object detection models for Autonomous vehicle applications [17], it was found that SDD and YOLO provide effective results under real-time latency requirements. Ideally, we would compare datasets against several different models, but for the scope of this paper, we will compare YOLOv4 [12] as a baseline to compare metrics of the datasets and determine if we can localize vacant parking spaces with reasonable precision.

V. Experiments

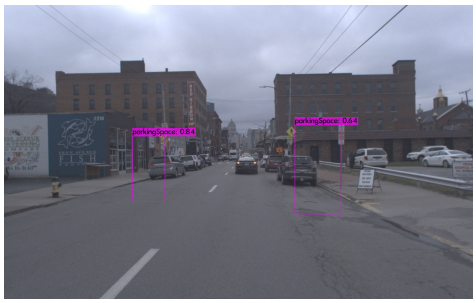
Each dataset was manually annotated using following the guidelines outlined previously and 2 experiment sets were run against a standard YOLO v4 model with starting network size of 608. To further improve precision, all models were trained with the random flag on. Rather than having a fixed dimension, this improves precision by randomly changes the network image size by factors of 32 every 10 iterations to expose the model to more variation.

The first experiment tested the performance of the trained models against their own test data, with the exception of the combined model whose test set comprised of data from all three datasets. The second experiment tested the trained combined model on each individual dataset.

Additional metadata that came with the datasets, such as localization for other object classes were not included within the training process but would be favorable to include for future work, which is further discussed in the conclusion.

The metrics utilized for evaluation are from the PascalVOC 2007 metrics - mean average precision (mAP), average precision, precision, recall and F1 score, where precision = $\frac{TP}{TP+FP}$, recall = $\frac{TP}{TP+FN}$ and $F1 = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$. mAP here is measured at a threshold of 50% (iou = 50%) where the detected bounding box and groud truth box have an overlap of at least 50%

The performances of each dataset and its corresponding model were evaluated utilizing the best weights from each model based on mean average precision. Tables 1 and 2 portray the outcomes of the following experiments after 3,000 iterations. Experiments were also done on the combined dataset after 6,000 iterations. The combined dataset consists of 4,500 images along with equal amounts of positive and negative annotations that span across the three different camera angles.



(a) 2 parking spaces detected with 64% and 84% confidence

(b) 1 parking spaces detected with 52% confidence

(c) 1 parking space detected with confidence of 86%

Fig. 2: Example of results

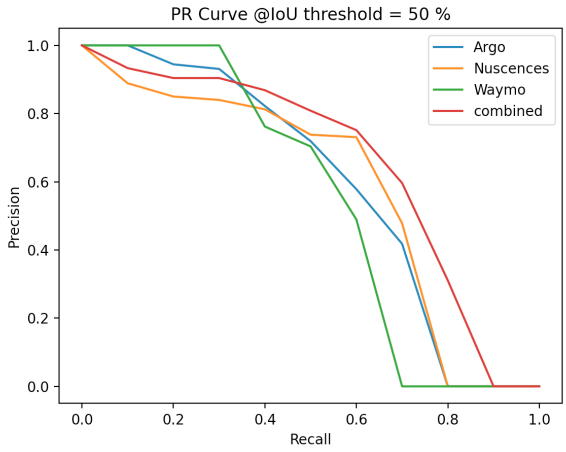


Fig. 3: PR Curve

	Argo	Nuscences	Waymo
mAP	68%	64%	45%
Precision	77%	85%	77%
Recall	61%	55%	37%
F1	68%	67%	50%

TABLE III: Experiment II: Combined Model Weights on Individual Test Datasets

	Argo	Waymo	NuScences	Combined
mAP	58%	42%	58%	64%
Precision	83%	81%	81%	79%
Recall	39%	28%	34%	56%
F1	52%	42%	48%	66%

TABLE II: Experiment I; YOLOv4 Model Metrics

VI. Results

A. Experiment I

Looking at Table II we see that the outputs of the YOLO model on the three separate datasets have somewhat similar performances. The datasets were first measured separately to gauge their performance individually and then combined the three train and test datasets with their annotations to see if performance would impair or improve.

As we detect for parking spaces, reducing both false positives and false negatives is important. Having false negatives imply that the model is passing on potential parking spaces, which is frustrating and not ideal to the user, but also not fatal to the user. On the other hand, focusing on diminishing FP's is more critical as a false positive would result in the car potentially crashing into another car. That being said, we want a model with good precision. All three datasets have similar precision, with Waymo and NuScences giving us 81% and Argo resulting in a slightly higher 83%.

The precision-recall curve is reported as well in Figure III. Waymo outperforms the other datasets at a lower recall but drops more quickly in precision as recall increases. The other datasets require slightly higher recall to reach precision in the 80th percentile. Overall at different thresholds, the combined dataset slightly exceeds the others, this is also reflective in the F1 score as well.

The combined dataset overall has a more stable performance than the 3 individual datasets with a mean average precision of 64% compared to 58%, 42%, and 58% for Argo, Waymo, and NuScences respectively. As the combined dataset had more exposure to different scenes and variations this performance makes sense.

B. Experiment II

Table III outlines the results of the combined trained model with the individual test datasets. The Argo set gives the model the best results overall against all thresholds, with an mAP of 68% . However, the combined model on the Nuscences dataset gives us an 85% precision rate.

While we're able to get reasonable precision for both experiment I and II, the recall for the datasets is the lowest metric. Reasons for the inconsistency might be due to the fact we only trained on 1,500 frames for the individual datasets. The original data from the separate companies had many frames left over to train on so to improve this metric we should continue to add more annotations and expand the train set.

Figure 2 details some examples of resulting frames from the model. Bounding boxes appear on the resulting frames if the model detects a space with over 0.5 probability. Frame a is using the Argo trained model with a scene in Pittsburgh. It correctly detects both available parking spaces behind the cars. Frame b is from the Waymo trained model using a frame in San Francisco and frame c is from the Nuscences model based in Boston. Both correctly detect a valid available spot between 2 cars, one with 52% confidence and 86% respectively.

VII. Conclusion

This paper gives us insight into how we can detect street parking spaces with decent precision from a drivers framework using a simple YOLO model and frames from several autonomous vehicle open datasets.

While we are able to get reasonable results with precision in the 80th percentile, there is still much work to be done and constraints within the paper that should be addressed. This work is limited to 2D object detection techniques and reliance on other vehicles to determine the presence of a valid vacant parking space.

Future work involves improving the accuracy of vacant parking street detection by expanding this to 3D object detection, incorporating sign detection and reading, and utilizing the localization of other objects within the autonomous vehicle datasets to develop more accurate and stricter rules for street parking detection. Additionally, because street parking is highly variable, and requires deeper knowledge of its surroundings, incorporating knowledge graphs to further

improve accuracy would be a vital concept to incorporate moving forward. For example, creating a knowledge graph that explicitly defines the relationship between a fire hydrant and parking space vs parking meter and parking space.

Testing parking detection against more object detection models and comparing results would be desirable and give further insight into how we can improve accuracy for this complex situation.

REFERENCES

[1] P. Oldfield. (2017) Searching for parking costs americans $73 billion a year". INTRIX. [Online]. Available: http://inrix.com/pressreleases/parking-pain-us/

[2] "Nwave Smart Parking Company," NWave. [Online]. Available: https://www.nwave.io/. [Accessed: 13-Mar-2021].

[3] "Smart Parking Limited – Reinventing the Parking Experience," Smart parking, 05-Feb-2021. [Online]. Available: https://www.smartparking.com/. [Accessed: 13-Mar-2021].

[4] Millard-Ball, Adam, 2019. "The Autonomous Vehicle Parking Problem," Transport Policy, Elsevier, vol. 75(C), pages 99-108.

[5] Fagnant, Daniel Kockelman, Kara. (2015). Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice. 77. 10.1016/j.tra.2015.04.003.

[6] Wu, M.-C., Yeh, M.-C. (2019). Early Detection of Vacant Parking Spaces Using Dashcam Videos. Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 9613-9618. https://doi.org/10.1609/aaai.v33i01.33019613

[7] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, James Hays. (2019). Argoverse: 3D Tracking and Forecasting with Rich Maps.

[8] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., others (2020). Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 2446–2454).

[9] "nuScenes: A multimodal dataset for autonomous driving", H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beijbom, In arXiv preprint arXiv:1903.11027.

[10] R. Girshick, "Fast R-CNN", Proc. IEEE Int. Conf. Comput. Vis., pp. 1440-1448, Dec. 2015.

[11] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, Jun. 2017.

[12] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[13] John, A., Meva, D. (2020). A Comparative Study of Various Object Detection Algorithms and Performance Analysis. International Journal of Computer Sciences and Engineering, 8, 158-163.

[14] Albawi, Saad Abed Mohammed, Tareq ALZAWI, Saad. (2017). Understanding of a Convolutional Neural Network. 10.1109/ICEngTechnol.2017.8308186.

[15] U. B. Nikhil Yadav, "Comparative study of object detection algorithms", International Research Journal of Engineering and Technology (IR-JET), pp. 586-591, May 2017.

[16] John, A., Meva, D. (2020). A Comparative Study of Various Object Detection Algorithms and Performance Analysis. International Journal of Computer Sciences and Engineering, 8, 158-163.

[17] M. Masmoudi, H. Ghazzai, M. Frikha, Y. Massoud (2019). Object Detection Learning Techniques for Autonomous Vehicle Applications. In 2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES) (pp. 1-5).