Vehicle Detection in Parking Lots Using Deep Learning Techniques

Łukasz Miłośnicki
Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdańsk, Poland *s202006@student.pg.edu.pl*

Mateusz Miłośnicki
Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Gdańsk, Poland *s202013@student.pg.edu.pl*

Szymon Zaporowski
Gdańsk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Multimedia Systems, Gdańsk, Poland *szyzapor@pg.edu.pl*

Abstract

Today, deep learning methods are being strongly developed and are used for many different tasks. This paper addresses the task of vehicle detection in parking lots. The focus of this study is to evaluate the performance of several versions of the YOLO (You Only Look Once) object detection algorithm on a self-created dataset, using the models in their default configurations with pre-trained weights from the COCO dataset. The dataset contains various lighting and weather conditions such as sunshine, cloudiness, and the presence of snow. Each YOLO version is evaluated using a range of metrics such as precision, recall, F1 score, and FPS. Methods for optimizing the use of models are then proposed and tested. The results demonstrate the trade-offs between detection accuracy and computational efficiency.

Keywords: Vehicle detection, Deep learning, YOLO, Parking lots

1. Introduction

Object detection is an important part of the field of computer vision, focused on identifying and localizing multiple instances of different objects in images. Detecting vehicles in parking lots is quite particular as the objects are often small and tightly packed, which can make distinguishing between nearby vehicles difficult. With the development of machine learning, neural network-based models such as the R-CNN, Fast R-CNN, and Faster R-CNN [5] (Two Stage Detectors) were introduced, followed by faster One Stage Detectors like YOLO [4], [6, 7, 8], which is well-suited for real-time applications such as traffic or parking monitoring, due to its performance.

Although traditional parking systems relied on entry/exit gates or sensors in each parking space, they are costly and difficult to scale. Vision-based approaches using cameras and deep learning models offer a more flexible and cost-effective alternative. These systems enable advanced features such as determining vehicle size and position, detecting movement direction, and identifying irregular parking behaviors.

2. Dataset

In order to evaluate vehicle detection models in parking lots in an effective manner, it is necessary to select a dataset that is appropriate for the task. In ensuring the model's resilience to environmental variability, it is essential to incorporate a diverse range of weather conditions, including conditions of sunlight, precipitation, and diverse lighting conditions. Furthermore, the inclusion of high-quality annotations and diverse perspectives is crucial in enhancing the reliability of the system's performance in real-world applications. Publicly available datasets, such as PKLot [1] or CNRPark+EXT [2], are generally not suitable for vehicle detection, as their annotations typically focus on parking spaces rather than the vehicles themselves.

2.1. Custom dataset

A custom dataset was created for the purpose of this study. The created dataset contains a variety of photographs of the parking lot situated in front of the University campus, captured under diverse lighting and weather conditions, including instances of sunshine, cloud cover, and snow. The images were collected remotely from two cameras located at the 7th floor inside the building, which caused part of the parking lot to be obscured by window and wall elements. A total of 328 images containing 3772 objects were labeled using a custom data annotation tool, creating a precise outline of the vehicles in the parking area. In the image with a size of 1920×1080 pixels, the average vehicle size is 78×55 pixels. This means that the vehicle occupies around 0.2% of the image. The data was collected in November over the course of six different days from 9 am. to 6 pm., when there is the highest traffic in the parking lot.

3. Evaluation

The purpose of this section is to determine whether the latest models from the YOLO family can be effectively used for vehicle detection in parking lots without additional training, by purely using pre-trained weights from the COCO dataset. For simpler testing and results that will be easier to interpret, only one class - "car" was chosen for detection. All tests in this section were conducted on a machine equipped with AMD Ryzen 5 5600 CPU and RTX 3060Ti GPU.

3.1. Baseline Two Stage Detector and YOLO Models

As a baseline, we included a two-stage detector - Faster R-CNN with MobileNetV3 Large FPN backbone [3], to compare with YOLO family of one-stage detectors. For YOLO, we evaluated the largest models from versions YOLOv8 through YOLOv11. The models were evaluated on 170 images collected over the course of two days, using weights obtained from COCO dataset for all models. The comparison was carried out using several key metrics: precision, recall, and F1 score, which collectively provided a general evaluation of the detection precision and robustness of the models. Since the systems used to detect cars in parking lots require near-real-time operation, the frames per second (FPS) metric was calculated. All metrics were calculated using the same IoU and confidence thresholds.

Table 1. Comparison of Object Detection Models

Model name	Precision	Recall	F1 score	FPS
Faster R-CNN MobileNetV3 Large FPN	0.7229	0.4618	0.5636	39.2
YOLOv8x	0.9432	0.5693	0.6862	**79**
YOLOv9-E	0.9420	0.5601	0.6826	47
YOLOv10x	0.9585	0.6449	0.7527	52.4
YOLOv11x	**0.9742**	**0.6464**	**0.7601**	47.4

As shown in Table 1, each of tested YOLO models outperformed Faster R-CNN with a MobileNetV3 Large FPN backbone in both accuracy and inference speed. This shows the advantages of One Stage Detectors which are capable of keeping a balance between inference time and high precision and recall - unlike Two Stage Detectors which often trade speed for accuracy especially when using bigger backbones.

YOLOv11 achieved the best results of all models tested while keeping good performance, and was therefore selected for further evaluation. To further evaluate YOLOv11, it was decided that every variant of the model would be tested. Each variant differs in a number of parameters, from 2.6M in the nano variant to 56.9M in the xlarge one.

3.2. Impact of input image resolution

This subsection aims to examine the impact of adjusting the resolution of the input image on the performance of YOLOv11 models, taking into account different model variants. The tests were carried out on the entire dataset that contained 328 images. The purpose of the experiments is to show the trade-off between inference time and detection accuracy. Since the YOLO model by default rescales the images to 640×640 resolution, it was assumed that using full-size images might give better results.

Table 2. Adjusted size of an input image - 960×540

Model name	Precision	Recall	F1 score	FPS
YOLOv11n	0.9215	0.7085	0.7585	40.49
YOLOv11s	**0.9360**	0.8546	0.8841	**40.98**
YOLOv11m	0.9093	0.8031	0.8292	33
YOLOv11l	0.9311	**0.9228**	**0.9269**	25.51
YOLOv11x	0.8768	0.8201	0.8475	18.21

Table 3. Adjusted size of an input image - 1440×810

Model name	Precision	Recall	F1 score	FPS
YOLOv11n	0.8760	0.9476	0.9032	**30.67**
YOLOv11s	0.8857	**0.9843**	0.9288	29.67
YOLOv11m	0.9025	0.9330	0.9110	19.72
YOLOv11l	**0.9169**	0.9760	**0.9439**	16.50
YOLOv11x	0.8576	0.9624	0.9016	10.05

Table 4. Adjusted size of an input image - 1920×1080

Model name	Precision	Recall	F1 score	FPS
YOLOv11n	0.8110	0.9962	0.8905	**18.80**
YOLOv11s	0.8767	**1**	0.9321	16.61
YOLOv11m	0.9344	0.9911	0.9600	11.21
YOLOv11l	**0.9425**	0.9983	**0.9682**	9.68
YOLOv11x	0.8710	**1**	0.9268	6

Tables 2, 3, 4 show the metrics for YOLOv11 variants across increasing input image resolutions. Based on the results, it can be observed that with increasing image resolution, up to the full resolution of the original images, the metrics improve significantly while at the same time requiring more time to process a single image frame. At the highest resolution, all models were able to reach nearly perfect recall, with the differences in precision. Among the tested models, YOLOv11l consistently achieved the highest F1 scores across all input resolutions, indicating great balance between precision and recall. These trends of increasing metrics as resolution increases might reflect the characteristics of the problem where objects are tightly packed next to each other leading to small, closely spaced bounding boxes. Higher input resolutions help differentiate these objects, improving overall detection accuracy.

4. Conclusions and future directions

This article compares recent YOLO versions for vehicle detection in parking lots using a custom dataset captured under diverse weather and lighting conditions. Results show that higher image resolution improves detection accuracy but increases inference time, which is critical for real-time applications. A promising sign was that smaller models while offering much faster inference could still achieve nearly perfect recall. With fine-tuning those models could become ideal for scenarios such as in our problem where real-time inference time is necessary.

Future work will explore how external conditions affect detection, using generative models or style transfer to enrich the dataset with challenging yet controlled examples. Additional plans include expanding the dataset, testing model robustness to partial occlusion, and fine-tuning smaller YOLO variants to optimize accuracy while preserving real-time performance for practical deployment.

5. Acknowledgements

This work was partly supported by the Polish National Centre for Research and Development (NCBR) through the European Regional Development Fund titled “INFOLIGHT-Cloud-Based Lighting System for Smart Cities” under Grant POIR.04.01.04/2019

References

[1] Almeida, P., Soares de Oliveira, L., Jr, A., Jr, E., Koerich, A.: Pklot - a robust dataset for parking lot classification. Expert Systems with Applications 42 (02 2015)

[2] CNRPark Dataset Team: Cnrpark+ext dataset. http://cnrpark.it/, accessed June 30, 2025

[3] Howard, A., Sandler, M., Chu, G., Chen, L., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q.V., Adam, H.: Searching for mobilenetv3. CoRR abs/1905.02244 (2019), http://arxiv.org/abs/1905.02244

[4] Khanam, R., Hussain, M.: Yolov11: An overview of the key architectural enhancements (2024), https://arxiv.org/abs/2410.17725

[5] Ren, S., He, K., Girshick, R.B., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. CoRR abs/1506.01497 (2015), http://arxiv.org/abs/1506.01497

[6] Wang, A., Chen, H., Liu, L., Chen, K., Lin, Z., Han, J., Ding, G.: Yolov10: Real-time end-to-end object detection. In: Advances in Neural Information Processing Systems. vol. 37 (2024)

[7] Wang, C.Y., Yeh, I.H., Liao, H.Y.M.: Yolov9: Learning what you want to learn using programmable gradient information (2024), https://arxiv.org/abs/2402.13616

[8] Yaseen, M.: What is yolov8: An in-depth exploration of the internal features of the next-generation object detector (2024), https://arxiv.org/abs/2408.15857