

Project Group: 19

April 6, 2025

# **Computer Vision Course Project: ParkVision: A Real-Time Parking Management**

**by**

Aaron James

Jacob Rempel

## **Abstract:**

Efficient parking management is a growing challenge in many urban environments, leading to issues such as wasted time, driver frustration, and increased fuel consumption. By design, ParkVision offers a computer vision-based solution that rapidly detects vacant parking spots from an aerial perspective. Using OpenCV-based image processing, combined with a YOLO deep learning model, trained on two Kaggle datasets containing approximately 32 000 images. Designed for parking lots with standardized, well-defined markings, ParkVision determines space occupancy within a short to medium range. With a low-cost setup, such as a single camera and a Raspberry Pi, ParkVision enables the monitoring of hundreds of spaces simultaneously. Ultimately, ParkVision aims to enhance parking efficiency, reduce congestion, and provide scalable, real-time guidance for drivers in environments such as airports and urban lots.

# 1 Introduction

Urbanization and population growth have led to a significant increase in the number of vehicles travelling, increasing the demand for efficient parking solutions in many different environments. Drivers often waste valuable time searching for available parking spots, contributing to increased traffic congestion, fuel consumption, and environmental pollution. Traditional methods of parking lot monitoring include physical on-foot monitoring, remote gate control with assistance from a surveillance system, or most prominently, none at all. The methods involving any active monitoring can be expensive, labour intensive and difficult to scale.

Recent advancements in computer vision and deep learning offer a favourable alternative. Leveraging object detection models, and real time image processing, automation of vacancy monitoring in parking lots is achievable with visual data alone. This not only reduces infrastructure and labour costs, but enables an easy, scalable, and adaptable parking management solution.

ParkVision is a computer vision-based solution designed to detect vacant and occupied parking spots from an aerial perspective using real time image processing and deep learning. Our project uses a YOLO-based object detection model in combination with OpenCV to analyse parking lot footage to determine the vacancy status in a parking lot. ParkVision focuses specifically on parking lots with clearly defined parking lines and standardized layouts, making it easily suitable for deployment in most structured urban and commercial environments.

The overall objective of this project is to provide real-time insights into parking lot occupancy, improve traffic flow, and enhance overall efficiency of urban parking systems.

# 2 Methodology

The ParkVision system is designed to detect vacant and occupied parking spots from aerial imagery using computer vision and deep learning. The data pipeline that this system uses is data preparation, model training, and real-time inference. All of these core components will be detailed throughout the following subsections.

## 2.1 Dataset Preparation

We worked with three different public datasets from Kaggle.com, which contained thousands of labelled parking lot images with corresponding annotated occupancy information. All of these datasets were first pre-processed to ensure consistency.

1. All annotations were converted to YOLO format (normalized center-based bounding boxes). Bounding box coordinates were extracted from XML and JSON in some cases.
2. The datasets were split into training, validation, and testing sets based on size. Datasets with over 10 000 images were split into 80% training, 10% validation, and 10% testing. The smaller dataset (under 10 000 images) was split into 70% training, 15% validation, and 15% testing. These ratios were selected based on research to optimize model performance. [3]
3. Bounding boxes were labeled either vacant or occupied based on the dataset annotations.

## 2.2 Model Architecture

We used several models from the Ultralytics YOLO YOLOv11 family, including YOLOv11n, s, m, and x. YOLO (You Only Look Once) models are well suited for real time inference given the ability to process entire images in a single forward pass. This makes them an ideal choice for ParkVision, where quick and accurate results are a highlighted priority in dynamic environments.

## 2.3 Occupancy Detection Logic

Each predicted bounding box was classified as either vacant or occupied. To evaluate accuracy:

- Predictions were compared with ground truth labels using Intersection Over Union (IoU).
- Matches above an IoU threshold of 0.5 and correct class labels are counted as True Positives.
- Unmatched predictions were False Positives.

- Unmatched ground-truth boxes were False Negatives.

## 2.4 Evaluation Metrics

The model was evaluated using the following metrics:

- Precision: How many predicted occupied/vacant spots were actually correct
  - High Precision = Few false positives
  - Low Precision = Model guessing too much
- Recall: How many actual occupied/vacant spots the model was able to correctly detect
  - High recall = few false negatives
  - Low recall = missing many real occupied/vacant spots
- F1 Score: Harmonic mean of precision and recall
  - High F1 = good balance of precision and recall
  - Low F1 = poor performance in one or both areas
- Detection Rate: Percentage of ground truth spots the model identified, as well as the models ability to locate all spaces in the image

## 3 Results

### 3.1 The PKLot Dataset

#### 3.1.1 Training

The first dataset we decided to use was PKLot [1] which contains approximately 700k images. The images come from three different cameras, two cameras are from the same parking lot and the third camera is from a completely different parking lot. Every image is taken at a different time of the day throughout the year, this results in a dataset that has extensive data on the different types of cars,

different weather conditions, and different times of day giving a ton of brightness deviations. We converted this data set into a YOLO ready format, and commenced training on a pre-trained YOLOv11 model, with a fine tuning of 25 epochs.

### 3.1.2 Overfitting and Limitations of PKLot

Unfortunately training a YOLO model on a dataset of the same three parking lots results in a model that is **very** good at identifying parking spaces **only** in that parking lot. This is the results of the YOLOv11m model after training for 25 epochs.

```
Summary Metrics Across All Images:

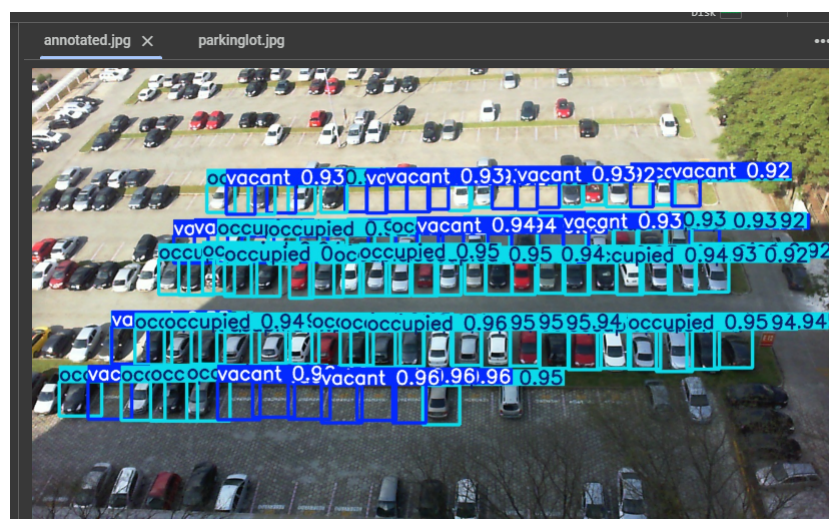
True Positives: 70247 out of 70320 ground truth spots
Detection Rate: 99.90%
Total False Positives (FP): 125
Total False Negatives (FN): 73

Overall Precision: 0.998
Overall Recall: 0.999
Overall F1 Score: 0.999

Average Precision across images: 0.998
Average Recall across images: 0.999
Average F1 Score across images: 0.998

Annotated images saved to 'ModelResults/Result'
Detailed log saved to 'ModelResults/Result/inference_detailed_log.csv'
```

The model is clearly overfitting the data, and running it for 25 epochs only amplifies the problem. Here is the model performing inference on in domain data:



When an out-of-domain image was selected, the model did not identify a single parking space. After reassessing the PKLot dataset, our conclusion was that the dataset lacked diversity, having nearly 700k images of essentially the same location. This model was able to perform inference very well for images of the same parking lots included inside the dataset. However, it failed to generalize to images from domains not represented in the dataset, rendering it ineffective for out-of-domain scenarios.

### **3.1.3 Next Steps**

After the shortcomings using the PKLot dataset, we took a step back and re-evaluated the dataset in use. We realized that diversity in a dataset to properly train a model needs to be significantly higher, so that the model can better generalize images outside of the dataset domain. We found another two datasets on Kaggle, that had a lot more parking lot diversity. The new datasets are the parking-lot-dataset and the carpk\_coco dataset. [2] [4]

### **3.1.4 Combining Datasets**

These datasets include a ton of different images and annotations of different parking lots from various angles. Since the two datasets seem to each offer their own level of diversity, we decided to combine them to achieve better results. The parking-lot-dataset is approximately 30.3k images/labels, and the carpk\_coco dataset is roughly 1 450 images/labels. It is noted that the parking-lot-dataset contains some of the PKLot dataset, so the images and parking lot data there are not completely removed, just reduced to be more reasonable to allow our model to better generalize.

## **3.2 Combined Results**

The results from training models off this new dataset are very promising. We trained a medium and an extra large YOLO model. Here are the results for the extra large model.

### 3.2.1 Extra Large Results

```

Summary Metrics Across All Images:

True Positives: 80092 out of 82132 ground truth spots
Detection Rate: 97.52%
Total False Positives (FP): 3285
Total False Negatives (FN): 2040

Overall Precision: 0.961
Overall Recall: 0.975
Overall F1 Score: 0.968

Average Precision across Images: 0.939
Average Recall across Images: 0.957
Average F1 Score across Images: 0.945

Annotated images saved to 'ModelResults/Result'
Detailed log saved to 'ModelResults/Result/inference_detailed_log.csv'

```

Figure 1: Summary Metrics of The Fine-Tuned Extra Large Model

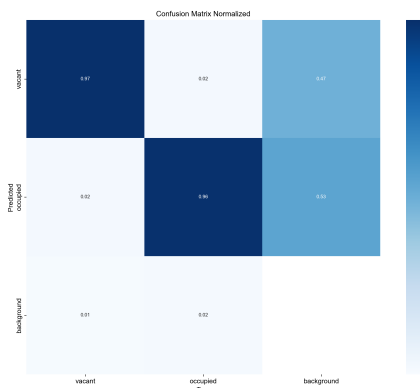


(a) Ontario Tech University's Parking Lot (Founders 2)

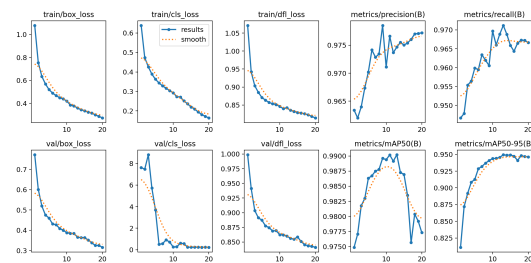


(b) A Generic Parking Lot

Figure 2: Inference from our fine-tuned Extra-Large YOLOv11 model



(a) The Confusion Matrix



(b) Model Performance Curves

Figure 3: YOLO Run Results

The plots summarize the successful training and validation of our object detection model trained over 20 epochs. The training and validation losses, including box, classification and DFL losses show a consistent downward trend, indicating the successful learning. The evaluation metrics such as precision, recall and mean Average Precision (mAP) steadily improve. The mAP@50 and mAP@50-95 reach high values by the end of the training. The smoothed curves highlight that the model is converging well without signs of overfitting.

The normalized confusion matrix also shows strong performance for the vacant and occupied classes, with 97% and 96% accuracy respectively. A noted area of struggle for our model is with background class. It correctly identifies only 53% of the time and often misclassifies it as vacant. Overall the classifier distinguishes well between the parking statuses, but there is some room for improvement. We believe with more powerful hardware, and more training time, we could have greatly improved these results.

## 4 Conclusions

This project presented numerous challenges. However, we have successfully developed a model capable of generalizing parking space detection. The real-world applications of this project are both practical and cost-effective. With just a single camera and a Raspberry Pi, the system can monitor hundreds of parking spaces, offering a low-cost and scalable solution. Such a system could be integrated into guided parking software, directing drivers to available spots in real time. In environments like airports, this approach offers a highly affordable alternative to existing parking management systems.



## References

- [1] P. Almeida, L. S. Oliveira, E. J. Silva, A. S. Britto Jr, and A. Koerich. Pklot: A robust dataset for parking lot classification. <http://web.inf.ufpr.br/vri/parking-lot-database>, 2013. Universidade Federal do Paraná & Pontifícia Universidade Católica do Paraná.
- [2] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu. Drone-based object counting by spatially regularized regional proposal networks. In *The IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017.
- [3] V. R. Joseph. Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 15(4):531–538, 2022.
- [4] Muhwira. Parking lot dataset. <https://www.kaggle.com/datasets/muhwira/parking-lot-dataset>, 10 2024. Dataset. Accessed: 2025-04-06.