

Vehicle Detection and Counting in Secure Valley Parking Using CNN and YOLO

Ms. Samruddhi Tushar Ogale and Ms. Asha Mane

Sadhu Vasvani Institute of Management Studies For Girls, Pune

Savitribai Phule Pune University

ogalesamruddhi2@gmail.com and ashamane90@gmail.com

Abstract: *When trying to manage vehicles in a car park or on a roadside, there can often be issues with keeping the number of vehicles consistent. This project uses computer vision to try to detect and count these vehicles. A YOLOv8 model was trained to detect vehicles. A unique dataset containing 5 classes (Ambulance, Bus, Car, Motorcycle and Truck) was used to train the model. The model was trained using Google Collab with a Tesla T4 GPU. The model was tested with images and videos. Object tracking was added for video detection. An easy-to-use interface was designed. The results show that the model is able to recognise vehicles of various types. However, there are some false positives and issues with the classification. This project demonstrates how deep learning can be used for vehicle monitoring. The groundwork has been laid to increase automated parking and traffic management.*

Keywords: *manage vehicles*

I. INTRODUCTION

The number of cars on roads and in car parks is increasing, so having automatic vehicle monitoring can help manage traffic and car parks. In a typical scenario, a member of security or an operator would need to watch cameras and estimate how many vehicles there are and what type. This is not the most efficient method, especially if there are many cars.

Computer vision can be used to detect objects. Deep learning models can be trained using labelled images to learn patterns from images. This can then be used to detect objects in new images/video frames. In this project, a YOLOv8 model will be used. This has been built for real-time object detection and can detect the position and type of objects. This system will detect 5 types of vehicles (Ambulance, Bus, Car, Motorcycle and Truck). The trained model will be tested using images and videos. For video detection, object tracking will be used. The aim of this project is to demonstrate how a deep learning detection system can be used for vehicle monitoring in a car park or controlled road.

II. PROBLEM STATEMENT

It can be very time-consuming to manually count vehicles in car parks or monitoring areas. There needs to be a system that can automatically detect vehicles in images/videos and tell what type of vehicle there are. This system needs to track the vehicles across multiple frames.

III. OBJECTIVES

The main objectives for this project are:

- Create a deep learning vehicle detection system using YOLOv8.
- Create and use a custom dataset containing 5 vehicle types.
- Train a detection model using Google Colab with a GPU.
- Detect vehicles using uploaded images.
- Process vehicle videos and use object tracking.



- Estimate the number of vehicle instances that have been detected.
- Have an easy-to-use interface to test the trained model.
- Consider the limitations and possible improvements to the proposed method.

IV. LITERATURE REVIEW

Previous research has shown how Deep Learning-based object detection can be used for traffic monitoring, vehicle classification and road scene analysis. Previous YOLO models have shown how objects can be detected and classified using just one detection. Improvements have been made to increase detection speed and accuracy and make it easier to use.

Recent research has focused on using YOLOv8 for vehicle detection and counting due to its real-time detection and easy training. Research has been carried out on detecting vehicles in road traffic, showing that camera systems can be used to automatically detect vehicles. Other research has focused on improving the different YOLO architectures.

Overall, vehicle detection has been widely researched. This is important as this project is using a custom dataset and test videos.

V. RESEARCH GAP

Many papers show good results using standard or well-prepared datasets. When deploying the model to monitor vehicles in a car park or road there may be different angles of the camera, different sized vehicles, different backgrounds and levels of occlusion. It is important to understand the full detection process from dataset, detection, tracking and counting.

This project focuses on implementing the full detection process using a custom vehicle dataset containing 5 classes. This project highlights the differences in detection performance and accurate counting.

VI. PROPOSED METHODOLOGY

The labelled vehicle dataset will be prepared and formatted correctly for YOLO. The YOLOv8 model can then be trained using the dataset. The best model weights will be saved to Google Drive.

For image detection, the model will detect where the vehicles are and what class they are. For video detection, the model will be run frame by frame. Object tracking will be used to link the detected objects. The final video will be saved for the user to view.

Key elements:

Dataset – Pre-Processing – YOLOv8 Training – Validation – Image Detection – Video Detection – Object Tracking – Counting – UI

VII. DATASET

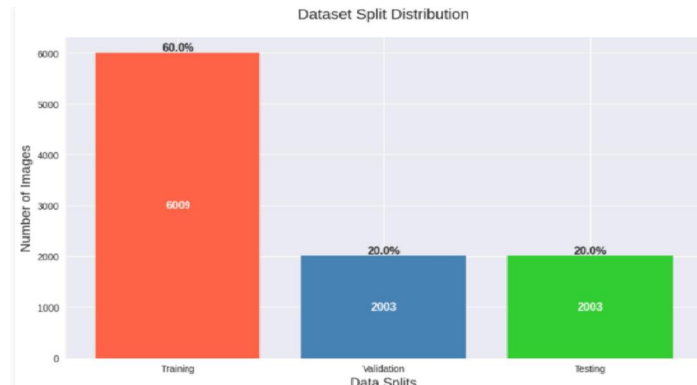
A custom dataset containing 5 vehicle classes (Ambulance, Bus, Car, Motorcycle and Truck) will be used. The dataset will be split into 3 sections:

Training, Validation & Testing.

The dataset contains a config file showing the location of the images, number of classes and names. The dataset was downloaded into Google Collab into the /content/vehicle dataset folder. This project used a dataset structure provided by RoboFlow.

As the dataset is not very large, the model may not generalise as well across different traffic scenarios. This will be taken into consideration when analysing the final results.





VIII. MODEL AND TRAINING

YOLOv8 was chosen to be used for object detection. YOLO is an object detection method. It can detect and classify objects. This can all be done in one detection process, so is ideal for near real-time.

The YOLOv8n model was trained for 30 epochs with an image size of 640px and a batch size of 8. This was done in Google Collab using an NVIDIA Tesla T4 GPU. The weights were saved to Google Drive, so the model can be reloaded after the runtime has been reset.

Main training parameters:

Model: YOLOv8n

Epochs: 30

Image Size: 640x640

Batch Size: 8

Device: NVIDIA Tesla T4 GPU

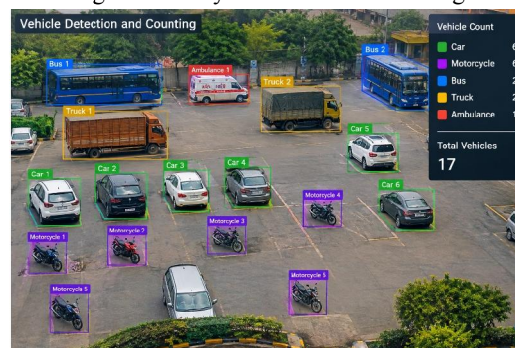
Classes: 5

Environment: Google Collab

IX. IMAGE DETECTION

Once the model has been trained, it can be saved and tested on some example images. The model will provide bounding boxes and class labels for each object detected. A confidence threshold was used to filter predictions.

One of the example images provided outputs for cars, motorbikes and buses. This shows that the model has been trained to detect multiple vehicle categories. However, there is variation in detection quality. This is expected due to the lack of data in the training set or if the image is visually different to the training set.



X. VIDEO DETECTION, TRACKING AND COUNTING

The trained YOLOv8 model was used with the ByteTrack algorithm for video detection. Tracking was used to link the vehicles across the frames.

A simple unique ID system was used to estimate the counts. The model will not increment the count each time a vehicle is detected. Instead it will keep track of the unique IDs of the tracked objects within each class. The count will rely on how many unique IDs there are.

The tracking and detection will greatly affect how accurate the count is. If the detector has false positives, changes the class of a vehicle or loses track of a vehicle and creates a new track this may mean that the final count is not as expected. The count provided should be seen as a prototype.

XI. SYSTEM DESIGN

The system can be shown using the following:

- Input Image/Video
- Create Dataset and Load Model
- YOLOv8 Vehicle Detection
- Bounding Box and Class Prediction
- ByteTrack Object Tracking (if video)
- Vehicle Category and Count Estimation
- Output with Annotations

This design separates detection and tracking. It makes the process easy and can be improved in the future.

XII. USER INTERFACE

A simple interface was created using Gradio. This allows either an image or video to be passed into the trained model, and the processed results can be viewed. For images, the detected vehicles and their category can be seen. For videos, the processed video will be returned.

This makes it easier to show what the project can do. No commands need to be typed into the model. This can be improved in the future to include confidence thresholds, live camera feeds and more detailed stats.

XIII. EXPERIMENTAL RESULTS

Throughout this project, the model provided results of around 0.31 precision, 0.24 recall, 0.208 mAP@50 and 0.103 mAP@50-95. For each class, the model provided results of:

Ambulance: ~0.190 mAP@50

Bus: ~0.335 mAP@50

Car: ~0.041 mAP@50

Motorcycle: ~0.357 mAP@50

Truck: ~0.117 mAP@50

This shows that useful patterns have been learnt for some of the vehicle classes. However, the model does not perform consistently across all classes. The Motorcycle and Bus performed relatively well. The Car detection was poor however, this can be improved by having more representative training data and training the model further.

XIV. DISCUSSION

Overall, this shows that a custom YOLOv8 model can be trained and used for vehicle detection without needing a large local computing system. Using Google Collab with a Tesla T4 GPU made training the model much easier.

However, the experiments showed that vehicle detection is not the same as vehicle counting. Even if the detector detects an object in each frame,



Improving the dataset and tracking was more important than increasing the confidence threshold. Having better camera positions, more training images per class, balanced classes, data augmentations and strict tracking rules can improve the system.

XV. LIMITATIONS

There are several limitations with the current implementation:

- Relatively small custom dataset.
- Variation in detection across vehicle classes.
- Some scenes in the test videos are visually different to the training dataset.
- Can cause false detections.
- Split tracking can cause incorrect unique vehicle counts.

XVI. FUTURE SCOPE

There are a few ways this can be improved. A larger and more varied dataset can be collected from real car parks. More training data can be added for classes that aren't performing as well. Data augmentation can be used to make the model more robust to changes in lighting and viewpoint.

The tracking can be improved by having strict tracking rules and better handling of lost tracking. A car park-specific version could be created to detect empty spaces, estimate how full the car park is, track entries/exit and create daily/weekly stats. A future version could include a live CCTV feed.

XVII. CONCLUSION

This project demonstrates a practical implementation of vehicle detection and counting using YOLOv8 and computer vision. A custom dataset was used containing Ambulance, Bus, Car, Motorcycle and Truck classes. The model was trained in Google Collab and tested using images and videos. Additional tracking was added using ByteTrack. A Gradio interface was created to make it easier to demonstrate.

The results show that the system can detect vehicles. However, improvements would need to be made to the dataset, detection and tracking to get reliable vehicle counts. Despite the issues, this project provides a useful base for automated vehicle monitoring. Further improvements can be made to create a more reliable smart parking/traffic management system.

REFERENCES

- [1] Terven, J. & Cordova-Esparza, D.-M., 2023. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS.
- [2] Reis, D. et al., 2023. Real-Time Flying Object Detection with YOLOv8 [online]. arXiv.
- [3] A Systematic Review of Drone-Based Road Traffic Monitoring System. [online]. IEEE Access. (2022).
- [4] Vehicle Detection, Counting and Classification Using YOLOv8 Deep Learning Model and OpenCV for Urban Traffic Management System. [online]. ICCES. (2024).
- [5] Guo et al. Research on Vehicle Detection Based on Improved YOLOv8 Network. (2024).
- [6] A Survey on Deep Learning Approaches for Real-Time License Plate, Vehicle Type and Face Recognition. (2023).
- [7] Deep Learning on Traffic Prediction: Methods, Analysis, and Future Directions. [online]. IEEE Transactions on Intelligent Transportation Systems. (202

