Vehicle Detection Project

---

1. Introduction

Vehicle detection underpins countless intelligent-transportation applications—from automated traffic analyses to autonomous driving. In this project, we train and validate a YOLOv8 detector to recognize and localize ten vehicle and pedestrian categories in still images. We leverage Ultralytics' PyTorch-based implementation, preparing a bespoke dataset of annotated road scenes and fine-tuning on our classes.

---

2. Dataset Preparation

- **Source images** were gathered from public traffic-camera feeds and open benchmarks.
- **Annotations** were maintained in a CSV (labels.csv), each row:

scss

CopyEdit

(image_id, class_name, x_min, y_min, x_max, y_max)

- **Class mapping**:

  { "car": 0, "pickup_truck": 1, "articulated_truck": 2, "single_unit_truck": 3, "work_van": 4, "bus": 5, "motorcycle": 6, "bicycle": 7, "motorized_vehicle": 8, "non-motorized_vehicle": 9, "pedestrian": 10 }

- **Train/Val split**: 80/20 by image ID to ensure no leakage.

Our preprocessing script:

1. Reads and groups CSV by image_id.
2. Converts each box to YOLO format—normalized center + width/height.
3. Copies images into runs/detect/train2/images/{train,val} and writes labels to runs/detect/train2/labels/{train,val}.

---

3. Model & Training

- **Framework**: Ultralytics YOLOv8 (ultralytics Python package).
- **Base model**: yolov8n.pt (nano) for fast iteration, later scalable to yolov8m/l.

- **Hardware**: Tesla T4 GPU (15 GB).

---

4. Model Summary

After training, the final best.pt was validated:

Ultralytics 8.3.141 🚀 Python-3.11.12 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)

Model summary (fused): 72 layers, 3,007,793 parameters, 0 gradients, 8.1 GFLOPs

---

5. Evaluation Results (on 1,126 images / 3,806 instances)

Class	Images	Instances	Precision	Recall	mAP@50	mAP@50–95
all	1,126	3,806	0.624	0.589	0.615	0.451
pickup_truck	311	434	0.795	0.786	0.844	0.688
car	944	2,544	0.835	0.882	0.912	0.667
articulated_truck	70	81	0.655	0.751	0.725	0.542
bus	105	121	0.931	0.897	0.938	0.830
motorized_vehicle	232	303	0.570	0.363	0.408	0.245
work_van	99	104	0.629	0.654	0.615	0.500
single_unit_truck	57	60	0.558	0.550	0.518	0.357
pedestrian	45	92	0.457	0.266	0.331	0.166
bicycle	22	26	0.481	0.577	0.640	0.408
non-motorized_vehicle	19	19	0.218	0.105	0.0913	0.0615
motorcycle	20	22	0.739	0.645	0.742	0.497

- **Overall mAP@50**: 0.615
- **Overall mAP@50–95**: 0.451

Inference speed (per image):

- Preprocess: 0.2 ms
- Inference: 1.7 ms

- Postprocess: 1.8 ms

---

6. Qualitative Insights

- **Cars** and **pickup trucks** achieve highest accuracy (mAP@50 > 0.84).
- **Buses** are nearly perfect (mAP@50 = 0.94).
- **Non-motorized vehicles** and **pedestrians** lag behind—small, irregular shapes and occlusion reduce recall.
- **Motorized_vehicle** class (a mixed category) shows lower recall, suggesting a need for more diverse training examples.

---

7. Conclusion

Our YOLOv8-based detector, with just 3 M parameters and 8 GFLOPs, achieves a solid mAP@50 of 0.615 across ten classes, running at sub-5 ms per image. With targeted data augmentation and model scaling, it's well positioned for real-time traffic analysis and autonomous-vehicle pipelines.