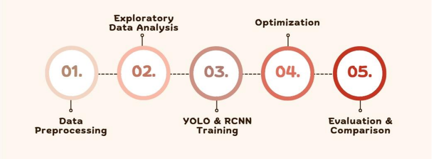
Real-Time Vehicle Detection and Classification on UA-DETRAC Using YOLOv8 and Faster R-CNN

Qinghao Yang / Joyce Lynn / Tianjie Wang
Northeastern University, Seattle Campus - CS6220

ABSTRACT

This project addresses real-time vehicle detection and classification based on the UA-DETRAC dataset. Our workflow consists of data preprocessing, exploratory data analysis (EDA), model training using YOLOv8 and Faster R-CNN, followed by performance optimization and comparison. Both models are evaluated on detection accuracy and inference speed. Results demonstrate that YOLOv8 achieves higher mAP and faster performance, making it more suitable for real-world deployment in intelligent traffic systems.



01. Data Preprocessing
02. Exploratory Data Analysis
03. YOLO & RCNN Training
04. Optimization
05. Evaluation & Comparison

INTRODUCTION

In recent years, the rapid development of intelligent transportation systems and smart cities has highlighted the importance of computer vision in traffic surveillance. Automatic vehicle detection and classification from roadside cameras enables data-driven traffic analysis, congestion prediction, and urban planning. Deep learning has emerged as a powerful solution to handle the complexity and variability of real-world video streams.

❓**Problem Definition:** Real-world traffic scenes pose multiple challenges for automated vision systems, including:

- Diverse vehicle types with subtle visual differences
- High-density traffic and frequent occlusion
- Lighting variations, camera angles, and motion blur
- The inefficiency of manual video analysis on large-scale surveillance footage

This project focuses on comparing two mainstream object detection models—YOLOv8 and Faster R-CNN—in terms of detection accuracy, inference speed, training cost, and suitability for real-time traffic applications.

METHODOLOGY

🔶 **YOLOv8 Pipeline:**

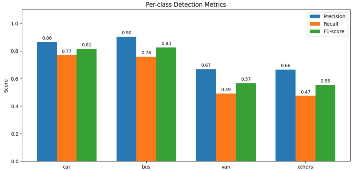
- **Model**: YOLOv8n (Ultralytics), initialized with pretrained weights
- **Dataset:** Full UA-DETRAC dataset used for training and validation, Raw annotations in XML format were converted to YOLO label format, Frames were extracted at **25 fps**, with **resolution 960×540**
- **Training:** Trained for **20 epochs** using image size = 640, batch size = 32, workers = 8, Approximate training time: **~7 hours**
- **Hyperparameter Tuning**: Grid Search on 8 combinations, Parameters:
  - lr0 ∈ {0.001, 0.003, 0.005, 0.01}
  - optimizer ∈ {SGD, Adam}
  - momentum ∈ {0.9, 0.937}
  - weight_decay ∈ {0.0005, 0.001, 0.002}
- **Best Configuration**: SGD + lr0=0.001, selected via mAP + F1

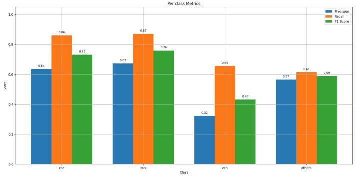
🔷 **Faster R-CNN Pipeline:**

- **Model**: fasterrcnn_resnet50_fpn from PyTorch with 5 output classes
- **Input Transform**: resized to **(min=800, max=1333)**, normalized using ImageNet mean/std
- **Training Strategy**: Used pretrained weights and fine-tuned on UA-DETRAC. **In each epoch**, 5000 images were **randomly sampled** from the full training set. Trained for **10 epochs total** to reduce training time. Approximate training time: **~4 hours**
- **Score Threshold Adjustment:** Instead of using a global threshold, per-class thresholds were applied:
  - ○ **Car: 0.5**
  - ○ **Bus/Van: 0.4**
  - ○ **Others: 0.3**

  This improves recall for harder-to-detect classes while maintaining high precision for easy classes.
- **NMS Threshold Tuning:** nms_thresh lowered to 0.4 (default is 0.5) for stricter merging of overlapping boxes, especially helpful in dense scenes.
- **Proposal Count Tuning:** Increased post_nms_top_n_test from 300 → 500 to retain more candidate regions in crowded frames.

RESULTS





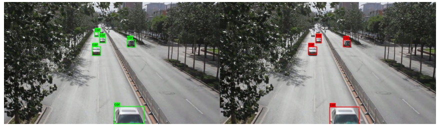
YOLO outperformed Faster R-CNN in all categories:

- Car: 0.81 vs 0.73
- Bus: 0.83 vs 0.76
- Van: 0.57 vs 0.43
- Others: 0.55 vs 0.59

CONCLUSION

YOLOv8 outperforms Faster R-CNN in both accuracy and efficiency. Its one-stage architecture enables real-time inference, while built-in data augmentation and full-dataset training enhance generalization. Compared to RCNN's two-stage pipeline and manual tuning, YOLO is faster, easier to use, and more accurate—especially for smaller or occluded vehicles. With stronger loss functions and automated optimization, YOLOv8 proves more suitable for traffic surveillance tasks.

RECOMMENDATIONS



Left—YOLOv8 detection; Right—Ground truth annotations from UA-DETRAC

🔍 **Model Suggestion:**
Given the results of our evaluation, we recommend deploying **YOLOv8** for real-time traffic detection tasks. It offers a better trade-off between **accuracy**, **speed**, and **ease of use**, making it ideal for intelligent transportation systems.

📸 **Visual Comparison:**
The image below compares a YOLOv8 detection frame (left) with the ground-truth XML annotations (right). YOLO accurately identifies all cars in the scene with high confidence, closely aligning with manual labels.

This project demonstrates that YOLOv8 achieves superior accuracy and efficiency, making it a more practical solution for real-time vehicle detection in intelligent traffic systems.

ACKNOWLEDGEMENTS

We thank the **UA-DETRAC team** for providing the benchmark dataset used in this project: **"UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking"**, Longyin Wen, Dawei Du, Zhaowei Cai, et al. (2015).We also extend our sincere gratitude to **Professor Yifan Hu** from Northeastern University for his insightful guidance throughout the CS6220 course project.