

Real-Time Object Detection

FOR AUTONOMOUS VEHICLES

DEPI Graduation Project

Presenter: Nizar Hussien (Team Lead)

December 2025

9/12/2025

DEPI Graduation Project | Real-Time Object Detection Slide 1

eyouth

1

Project Concept



The Problem

Autonomous systems lack reliable environmental perception in adverse conditions (low light, fog), creating safety risks.

Project Concept



The Problem

Autonomous systems lack reliable environmental perception in adverse conditions (low light, fog), creating safety risks.



The Solution

A real-time YOLO detection model optimized for edge deployment, to identify pedestrians & vehicles with low latency.

Project Concept



The Problem

Autonomous systems lack reliable environmental perception in adverse conditions (low light, fog), creating safety risks.



The Solution

A real-time YOLO detection model optimized for edge deployment, to identify pedestrians & vehicles with low latency.



Unique Value

High-speed detection optimized for real-time driving environments
Engineered for < 200ms latency,

Model Architecture & Data Flow

Inference Pipeline (Core Focus)

- ✓ **Input:** Live Video Feed from vehicle cameras.
- ✓ **Preprocessing:** Frame normalization and resizing (640x640).
- ✓ **Detection:** YOLOv8 model runs inference on the frame.
- ✓ **Output:** Structured data (Bounding Boxes and Class Labels).
- ✓ **Performance:** Achieving fast inference for real-time operation.



Target Users & Features



Primary User

The Autonomous Vehicle System:

Requires instantaneous, machine-readable perception data to make life-critical decisions without human intervention.

Target Users & Features



Primary User

The Autonomous Vehicle System:

Requires instantaneous, machine-readable perception data to make life-critical decisions without human intervention.

Key Features

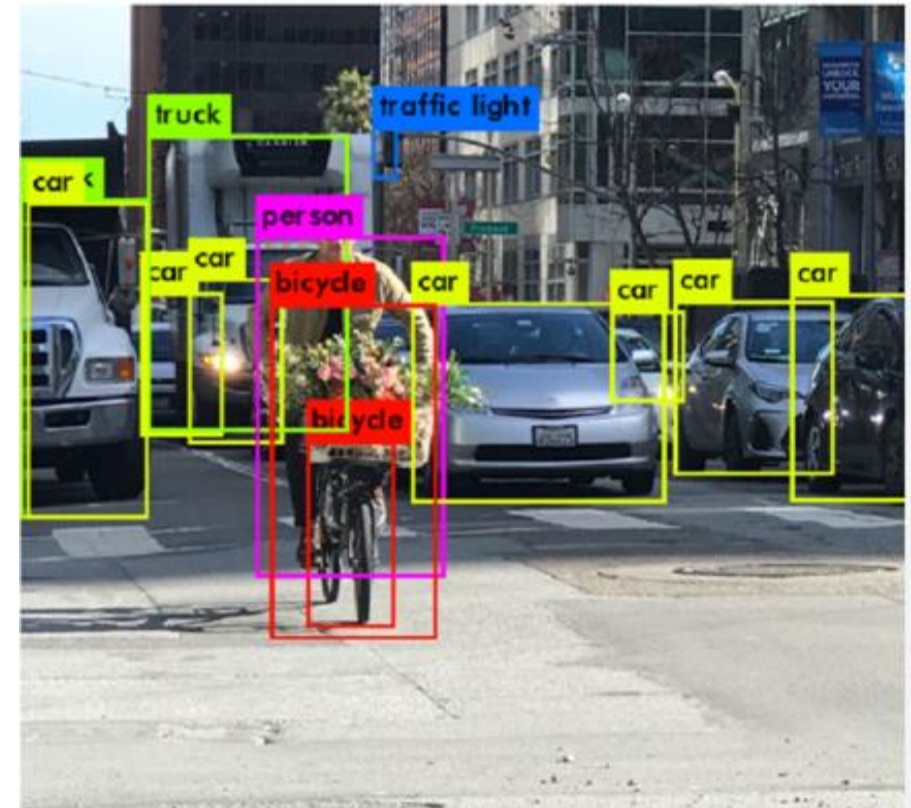
- ✓ **Low Latency:** Optimized to respond in $< 200\text{ms}$.
- ✓ **10-Class ID:** Pedestrians, Cars, Traffic Lights, etc.
- ✓ **High mAP:** Maximizing Mean Average Precision ($>70\%$).
- ✓ **Robustness:** High accuracy in diverse conditions.

Data Strategy & Processing

Dataset: COCO-2017 Subset

We filtered the massive COCO dataset to focus on
10 driving-critical classes.

Out of 80 Total Classes



Data Strategy & Processing

Dataset: COCO-2017 Subset

We filtered the massive COCO dataset to focus on
10 driving-critical classes.

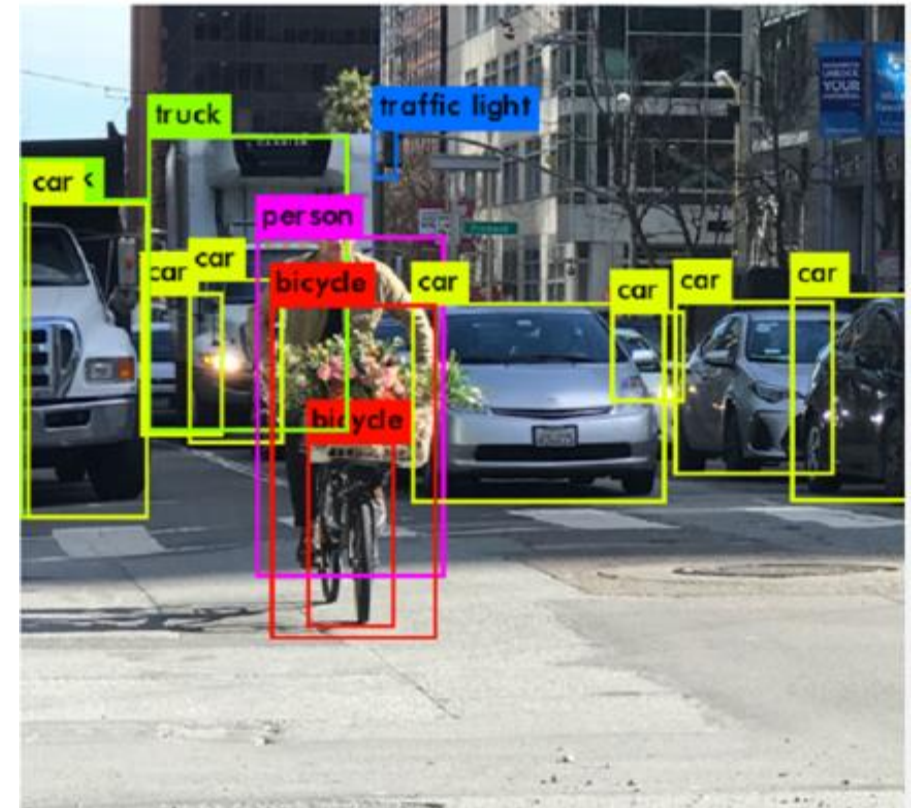
Pipeline Implementation

Filtering & Selection: Python script extracts only relevant images.

77K Out of 200K+

Normalization: Converts COCO [x,y,w,h] to YOLO format.

Config: Automatically generates data.yaml for training.



Technology Stack



Core & ML

Python (Main Language)

YOLOv8 (Ultralytics Framework)

PyTorch (Engine)



Data & Tools

Pandas & NumPy (Data Handling)

OpenCV (Image Processing)

Git (Version Control)

Performance KPIs

The model is optimized to meet strict real-time safety metrics.

Metrics achieved on the validation dataset during training.

< 200ms

Target Latency

> 70%

mAP
(Accuracy)

*Training parameters: Image size **640x640**,
Batch size 16.*

Evaluation, Success & Next Steps

The model is optimized to meet strict real-time safety metrics.

Metrics achieved on the validation dataset during training.

78.5ms

Inference Latency

> 70%

mAP
(Accuracy)

*Training parameters: Image size **640x640**,
Batch size 16.*

Evaluation, Success & Next Steps

The model is optimized to meet strict real-time safety metrics.

Metrics achieved on the validation dataset during training.

78.5ms

Inference Latency

71.1%

mAP@50 Achieved

*Training parameters: Image size **640x640**,
Batch size 16.*

Evaluation, Success & Next Steps

78.5ms

Inference Latency

71.1%

mAP@50 Achieved

Performance Summary

Our model is highly efficient and demonstrably fast:

- ✓ **Real-Time Validation:** Achieved **78.5ms** (2.5x faster than 200ms target).
- ✓ **Accuracy Gain:** Increased mAP by **12.7%** via extended training.
- ✓ **Remaining Challenge:** Closing the 9% accuracy gap, especially for small objects (e.g., Traffic Lights).

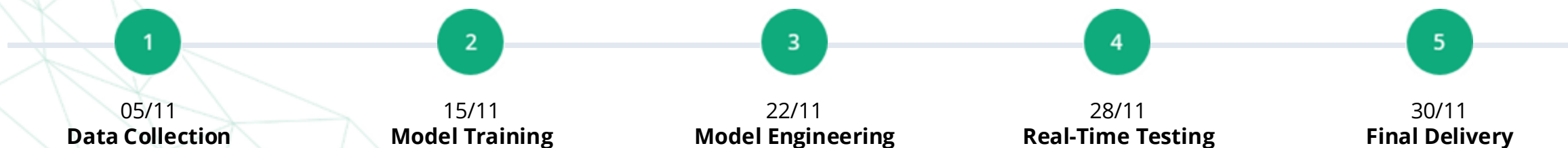
Next Step: Switch to YOLOv8-Medium and implement Targeted Augmentation.

Live Deployment



<https://youtu.be/UQaHwTuYU6g>

Milestones & Deliverables



Final deliverables include the working source code, and comprehensive technical documentation.

Project Team

Member	Role	Key Responsibility
Nizar Hussien	Team Lead	Project Management & Data Acquisition & Presentation
AbdElRahman Ahmed	Data Scientist	Data Conversion & Annotation
Ahmed Ashraf	ML Engineer	Model Architecture & Training
Mohamed Ashraf	ML Engineer	Model Deployment & Validation
Elsayed Aboulila	ML Engineer	Testing & Documentation

Thank You

Questions & Discussion