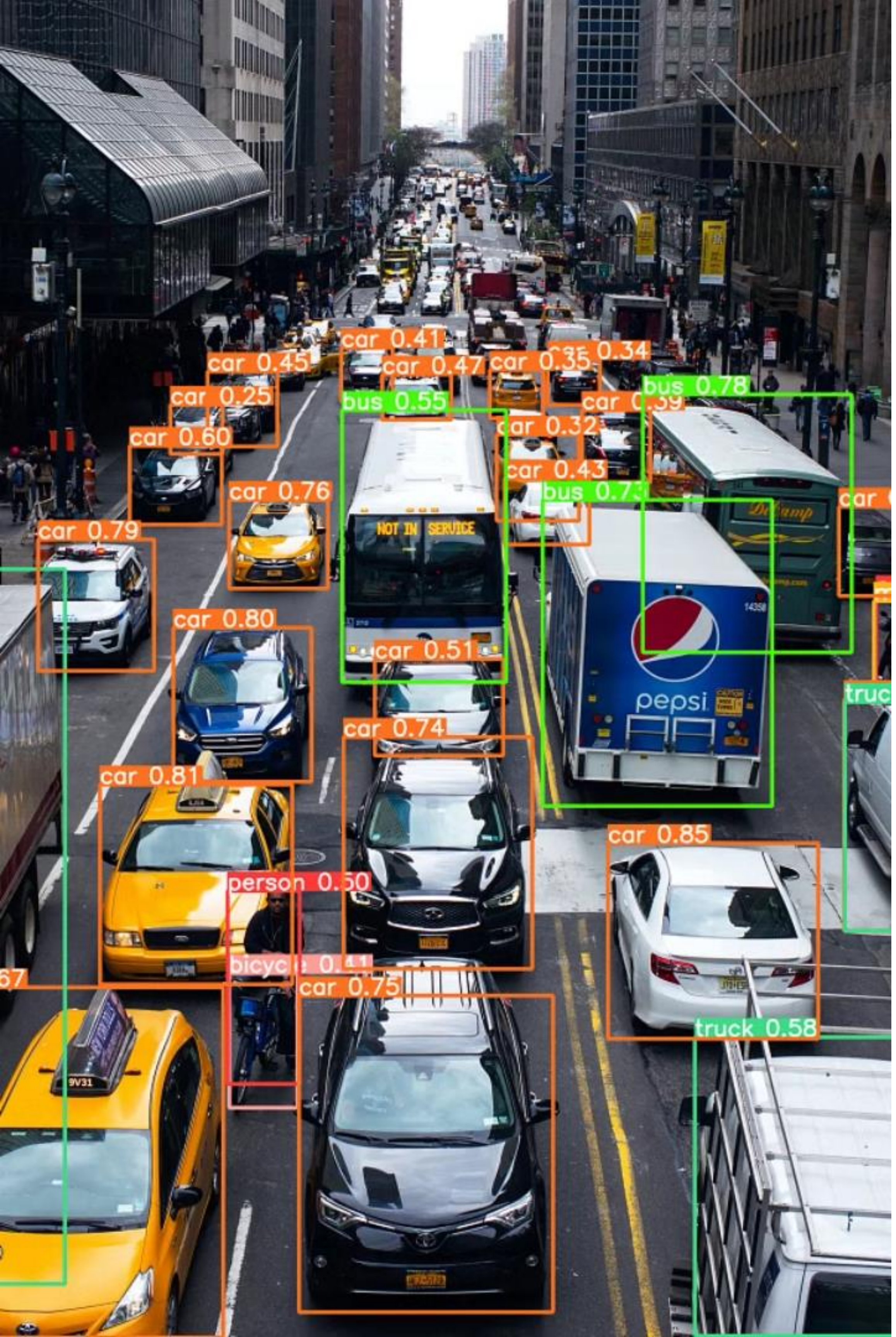


Automated Vehicle-Level Feature Annotation Tool: Methodology & Results

Name: Fareeda Saad

Major: Computer Science and Engineering

Supervised By: Dr Catherine Malak Elias

Project Objectives

Literature Review

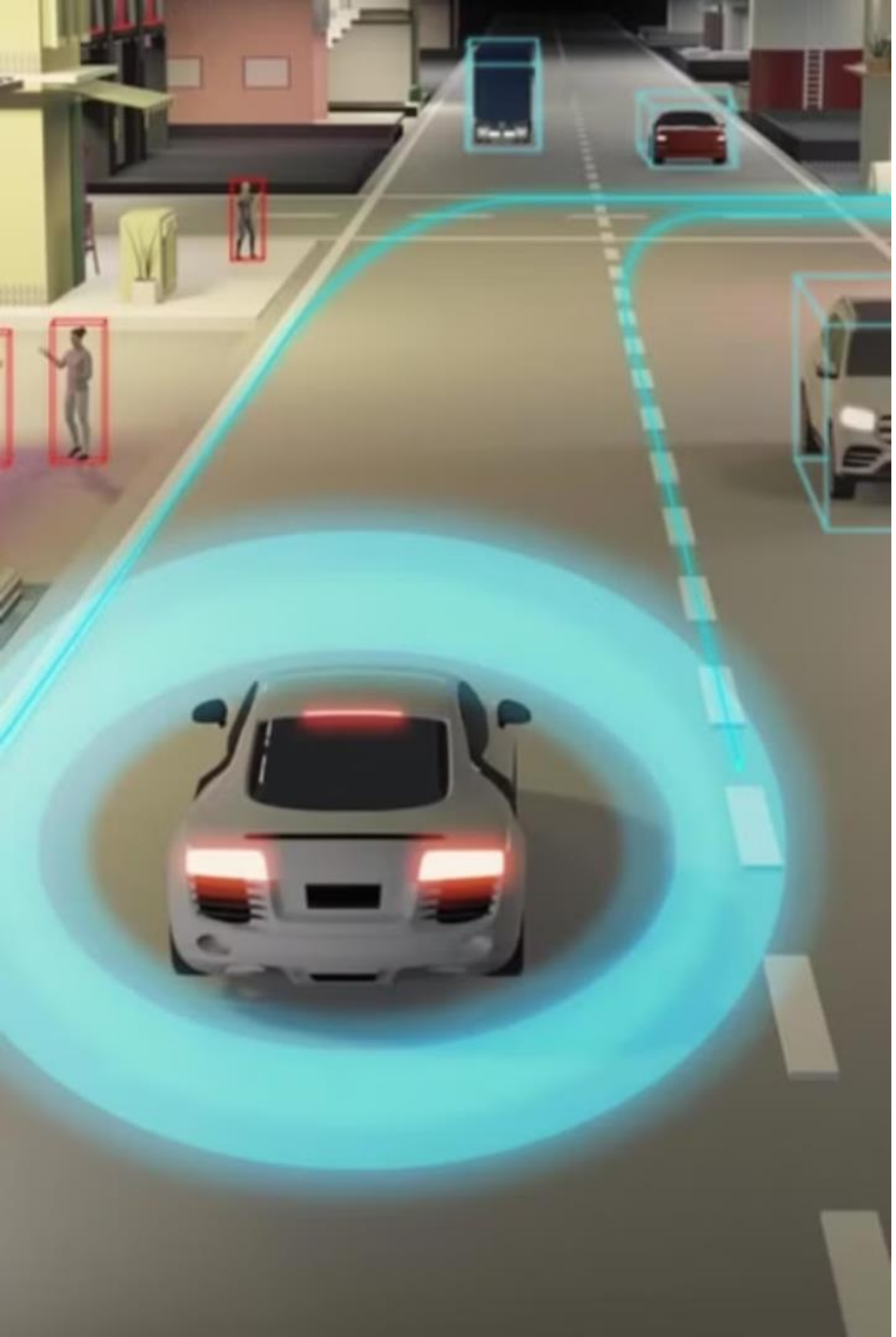
Summarize insights from *20 research papers* to identify key vehicle *features* and *detection techniques.*

Methodology Design & Implementation

- Explore ***2D automatic annotation*** using YOLOv3,YOLOv11(combined with depth anything v2) and ***3D manual annotation*** using OpenCV, CVAT, and Mindkosh.
- Demos

Visual Results

Present samples from YOLOv3, CVAT/Mindkosh, and annotated video frames.



Literature Review Summary

Most Commonly Detected Vehicle Features

- Vehicle Type
- Trajectory & Speed
- 3D Bounding Box (Position, Orientation, Dimensions)
- License Plate / ID
- Object Motion Status (Moving / Stationary)

Popular Detection Methods Methods

- YOLO / CNN-based Object Detection Detection
- LiDAR Point Cloud Segmentation Segmentation
- Sensor Fusion (Camera + LiDAR + Radar)
- 3D Object Detection Models (PointNet, VoxelNet)
- Manual Annotation via CVAT / LabelImg

Methodology & Implementation

1. Began with *2D automated* detection using ***YOLOV3***.

2. Developed *manual 3D* annotation using ***OpenCV***:

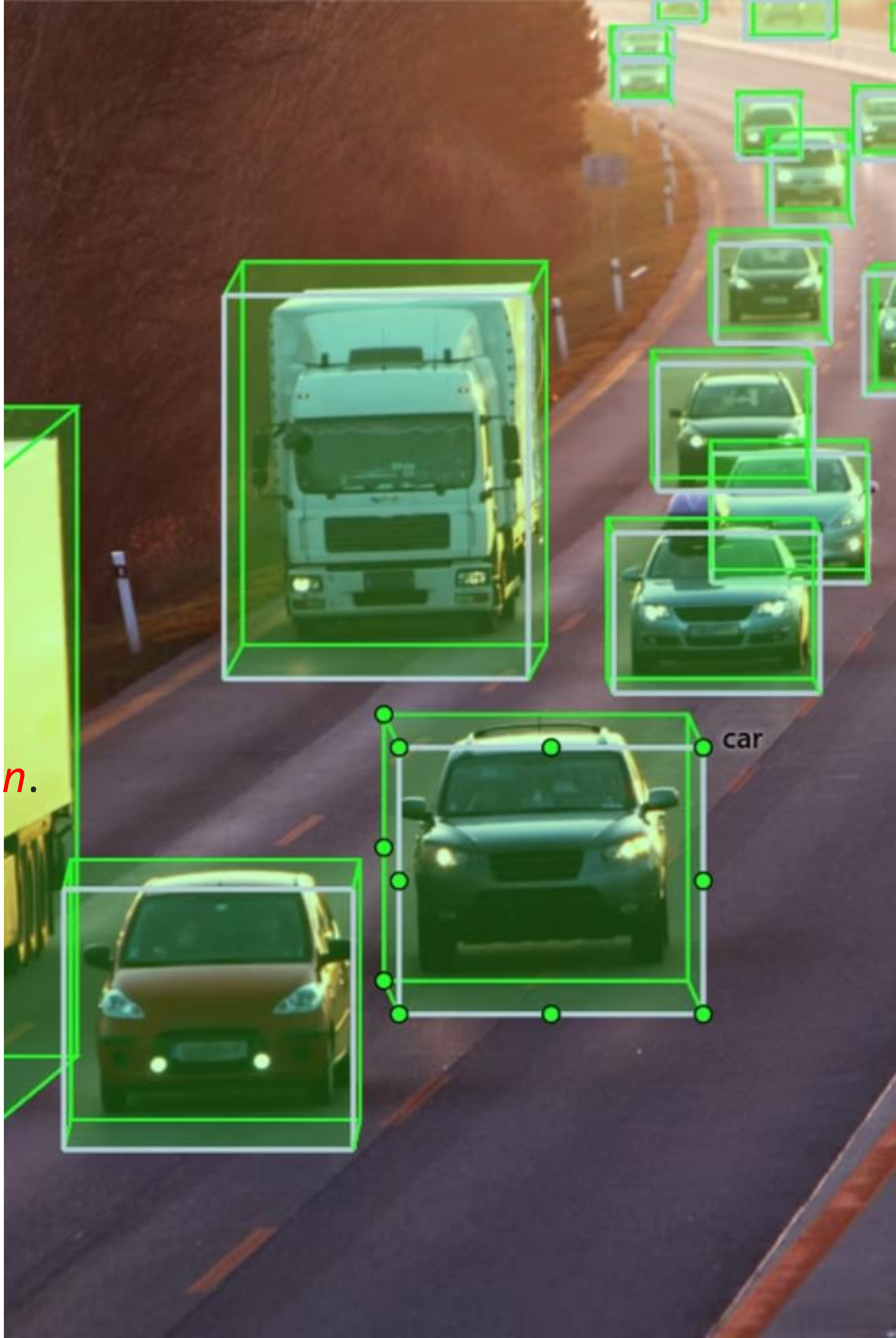
- Code 1: Draw cuboid by *entering coordinates*.
- Code 2: Draw Cuboid by *mouse-clicks* on image.

3. Switched to ***CVAT*** and ***Mindkosh*** for more efficient *3D annotation*.

4. Annotated **videos** *manually* using *3D bounding boxes*.

5. Annotated *videos automatically* using ***YOLOV11*** and ***Depth Anything v2***..

Our methodology utilized a hybrid approach, combining automation and manual processes for efficient annotation.

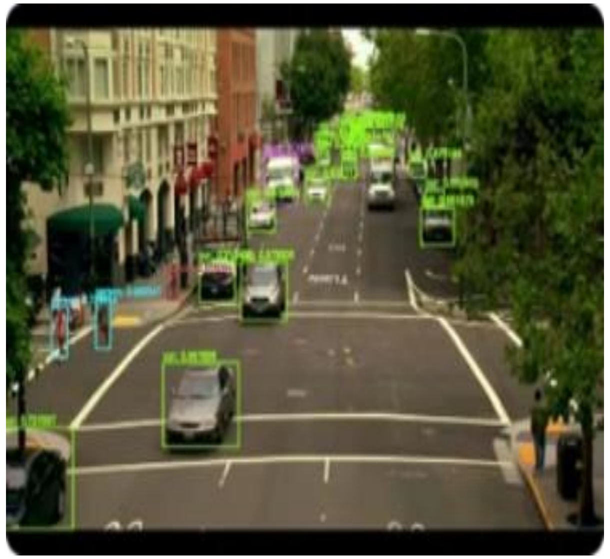


Demos

These are my five main demos to showcase functionality.

YOLOv3 2D Vehicle Detection

Demonstrates real-time 2D vehicle detection using YOLOv3.

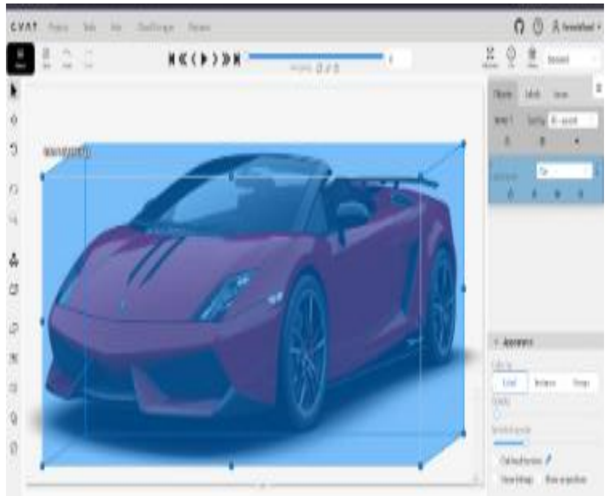


3D Manual Bounding Box Box Annotation using OpenCV

Shows the process of manually creating 3D bounding box annotations using OpenCV.

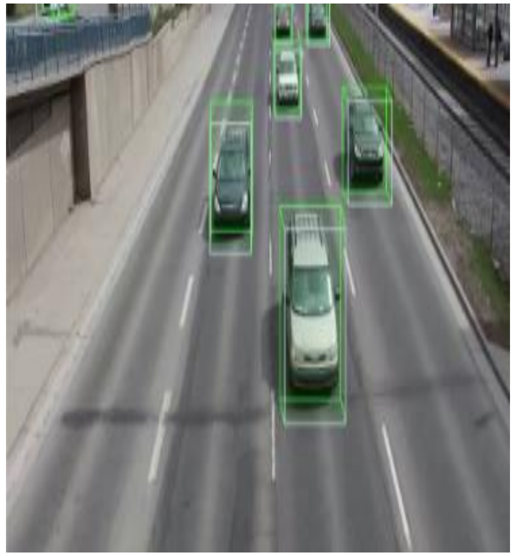
CVAT/Mindkosh Annotation Workflow

Illustrates the workflow within the CVAT/Mindkosh annotation platform.



60-frame video

Demonstrates results of *3D manual annotation* across 60 60 video frames using *CVAT.*



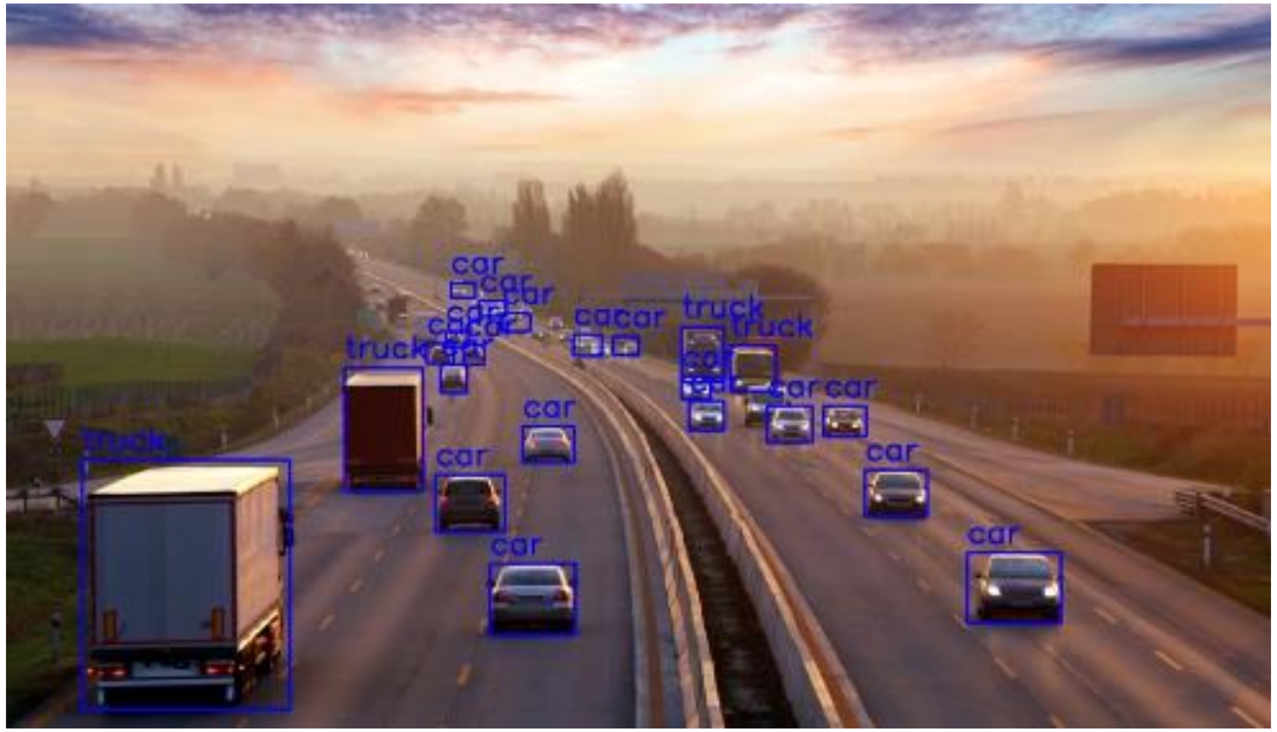
YOLOV11 with Depth Anything v2

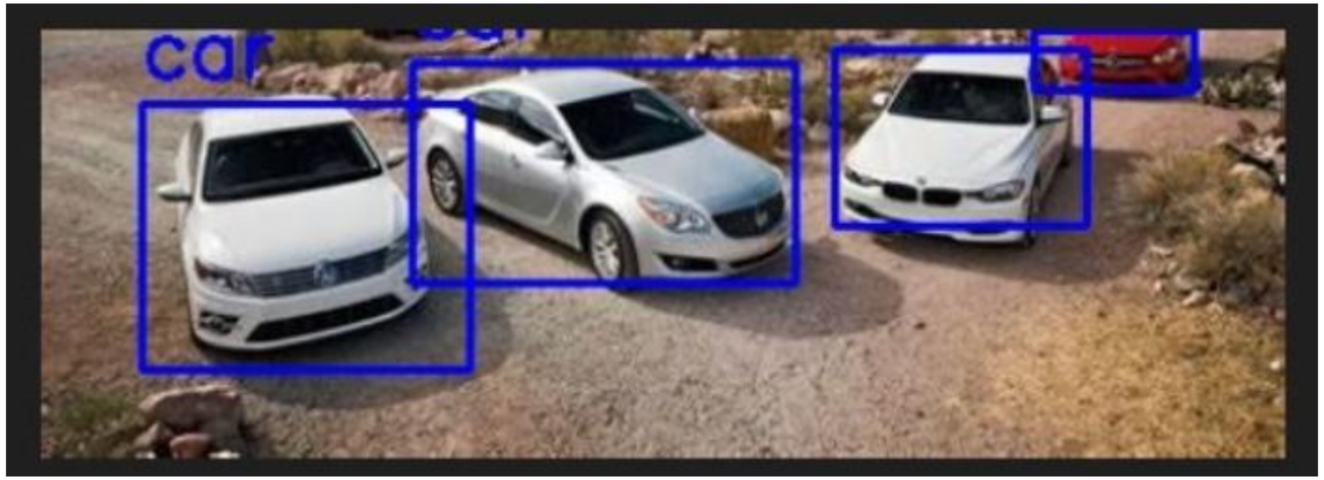
real-time 3D object detection system that combines ***YOLOv11*** for object detection with Depth Anything v2 for depth estimation to create pseudo-3D bounding boxes

Results

Here are some sample outputs showcasing both our automatic and manual annotation methods.

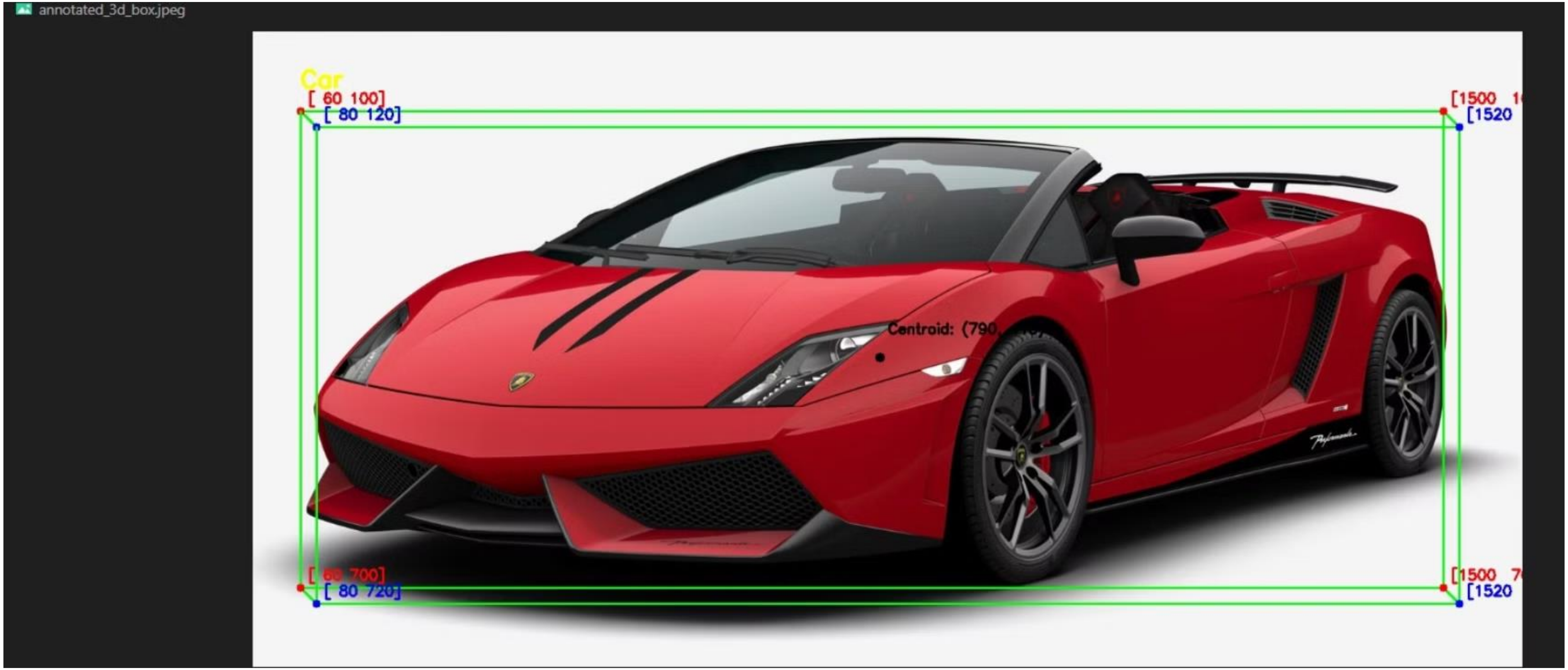
1 Yolo Detection (2D bounding boxes)



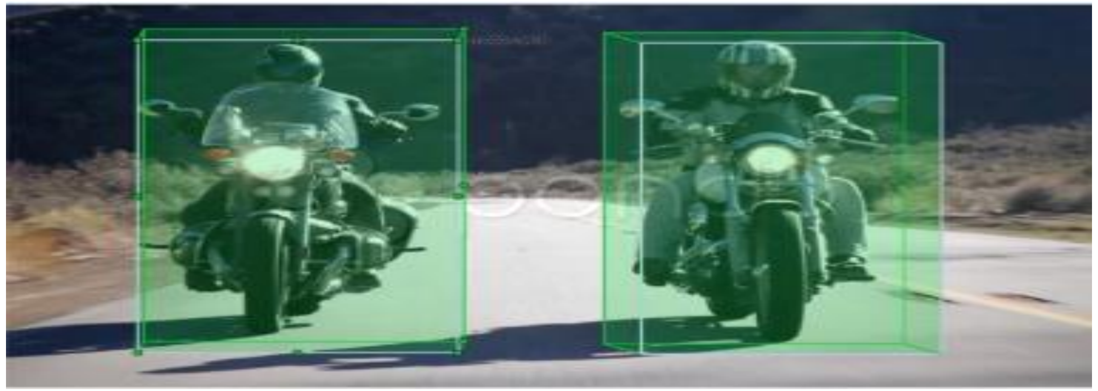


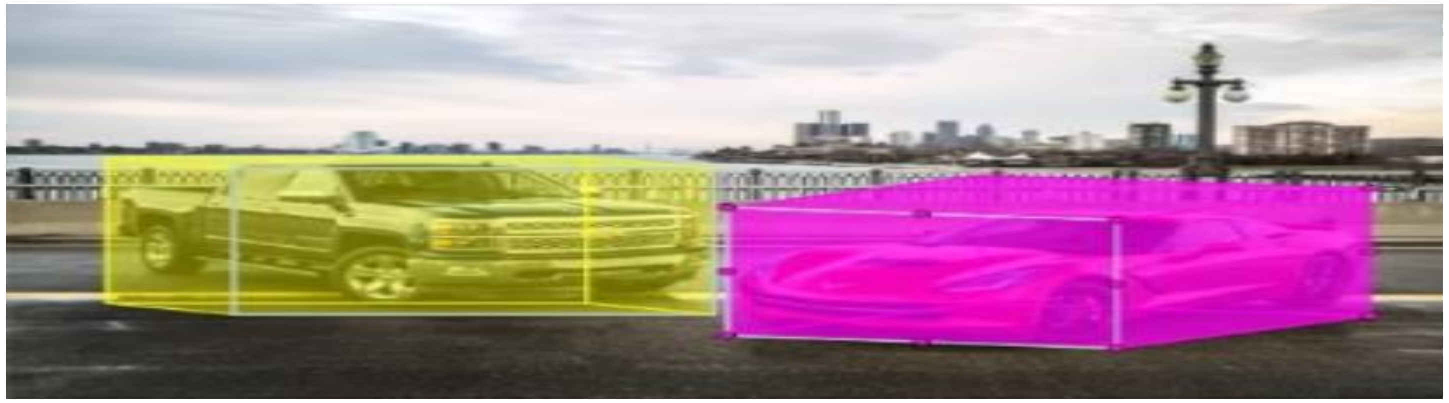
2

3D bounding boxes manual annotation using OpenCV



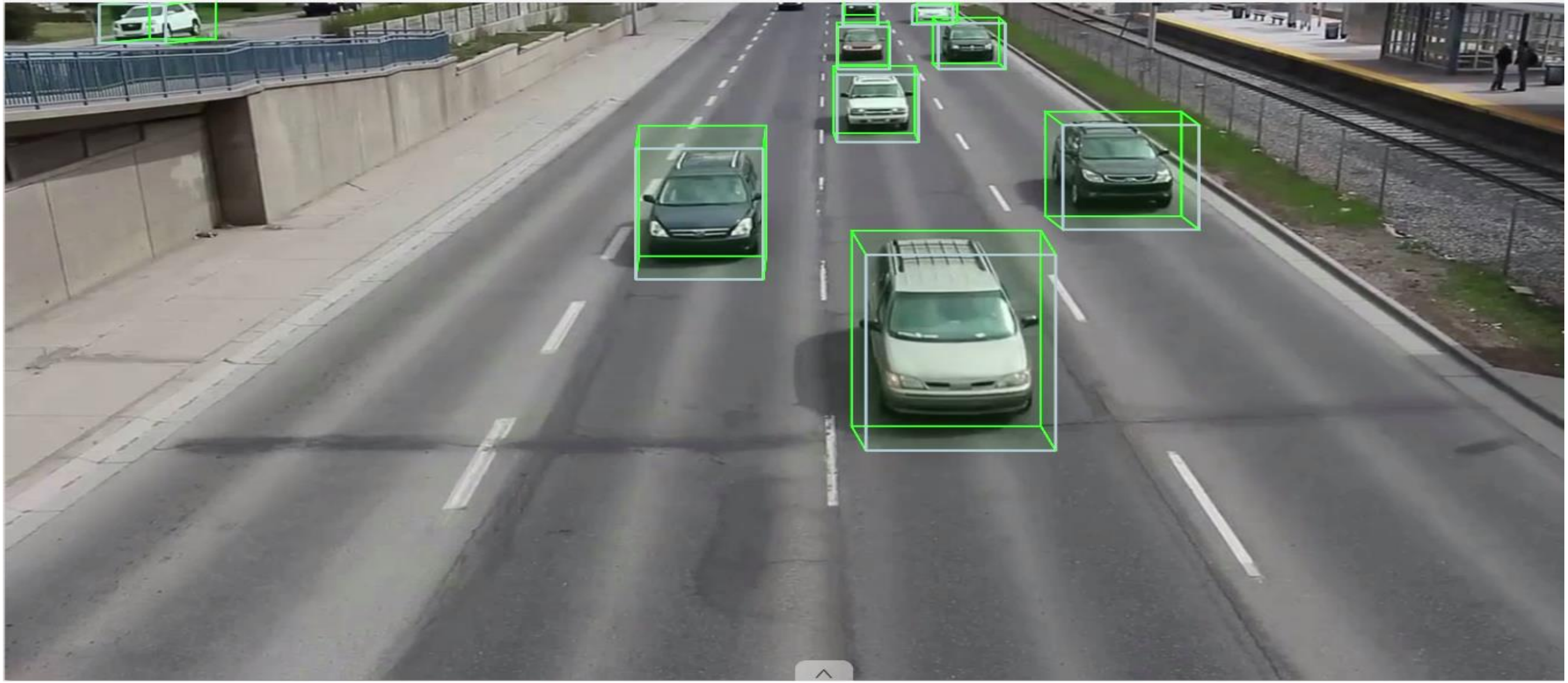
3 Annotated CVAT/Mindkosh Outputs





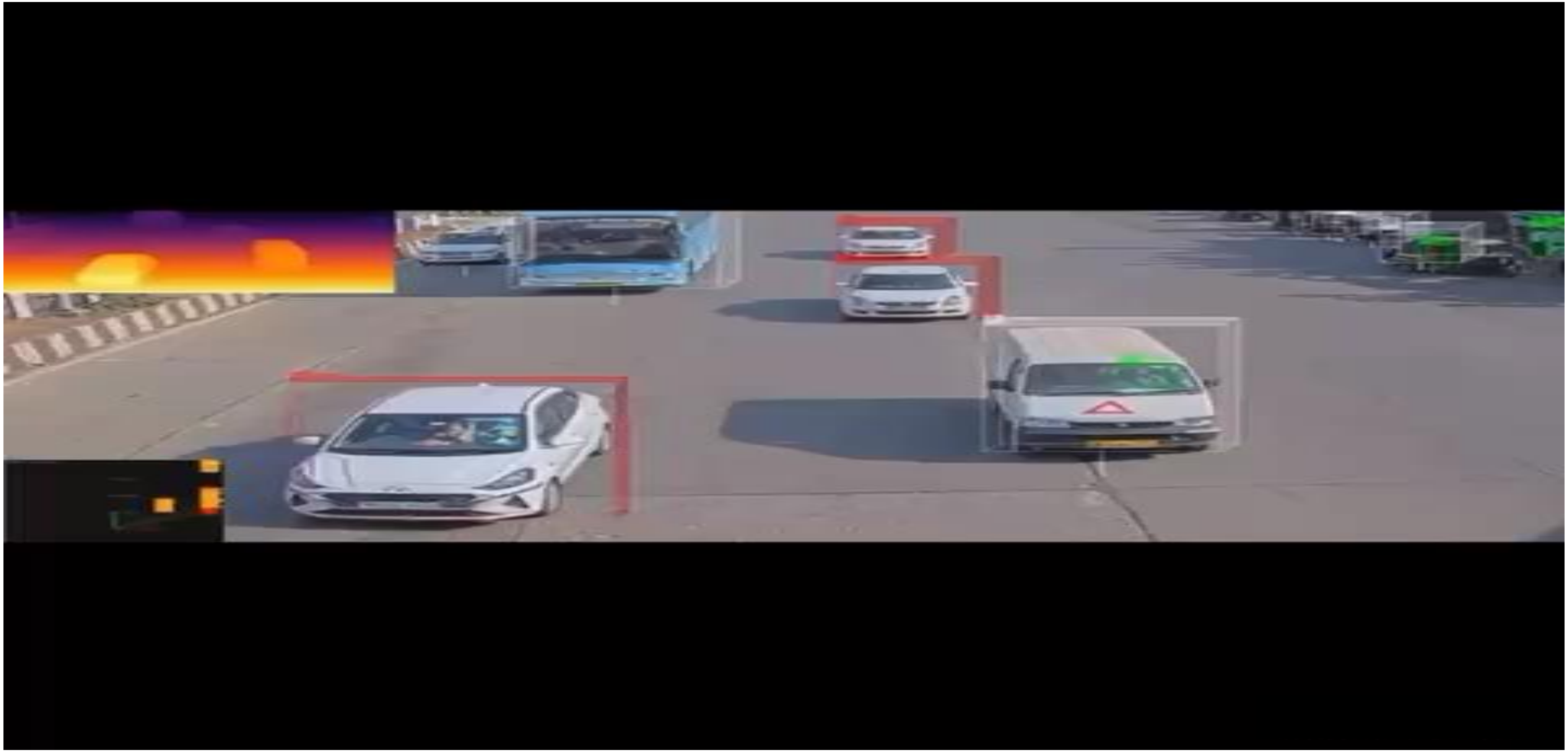
4 Annotated Video Results

Here is the *60-frame video* I annotated. It shows the progress of labeling over 60 frames. Doing this frame-by-frame using the track features helps create high-quality training data for vehicle detection models.



5

YOLOv11 and Depth Anything v2



What's Next?

Future Work:

- 🔍 **Explore *automated 3D annotation* tools**

  Find and test the *best tool* for automatically adding *3D labels* to *vehicles* in images and videos.

- 🎥 **Create more annotated videos**

  Manually label *longer video sequences* with many frames to improve *consistency* and training *data quality*.

- 🧩 **Label more detailed features**

  Start with manual labeling of *specific vehicle details* (like behavior of vehicles, velocity), then work towards automating that process.

- 🧩 **Handling Occlusions in 3D Annotations**

Any Questions?

Thank you for your time. I welcome any questions you may have.