BATIN TOPBAŞOĞLU 276718

CANER OLCAY 276715

MYRA SHAH 276834

25/06/2025

Project Final Report

Table Of Content

1. Introduction

- Project Purpose

In this project, we developed a visual system that helps understand and analyze road scenes using images. The main goal was to identify vehicles, understand their positions, estimate how far they are from the camera, and create a simple 3D representation of the scene. To do this, we used a combination of modern image analysis tools and artificial intelligence models.

By analyzing a single photo taken from a roadside camera, the system can detect different types of vehicles like cars, trucks, and buses. It can also estimate how far each object is and rebuild a basic 3D view of the area using this information. The project also includes features like visualizing the road layout and comparing the results with reference 3D models to check accuracy.

This solution can be used in areas like traffic monitoring, smart city systems, or autonomous driving research.

2. Project Pipeline

- Tools and Frameworks Used

os	File path operations
cv2 (OpenCV)	Image processing and lane detection
numpy	Running deep learning models (PyTorch backend)
torch	Numerical computations
open3d	3D point cloud, mesh and visualization operations
PIL.image	Image loading and preprocessing

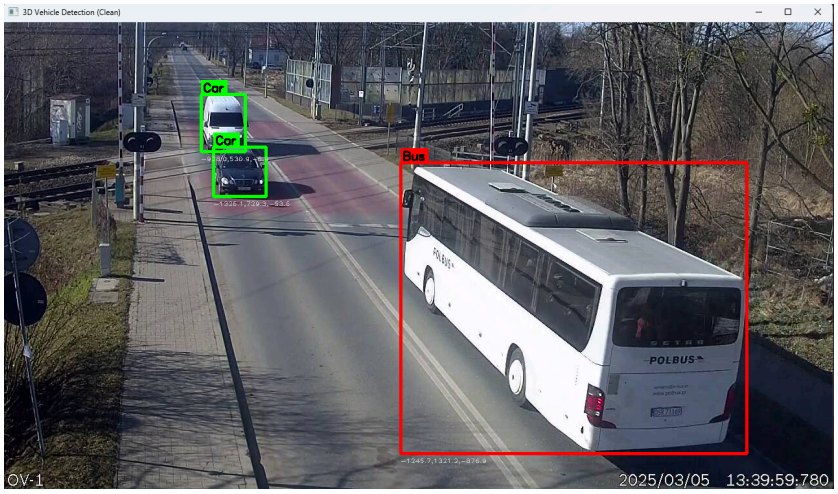
ultralytics.YOLO	Object detection (YOLOv8 model)
transformers.DPT	Depth estimation using Intel DPT model
matplotlib.pyplot	Plotting and saving visualizations
scipy.spatial.transform.Rotation	3D rotation and transformation utilities

3. Scene Processing Pipeline

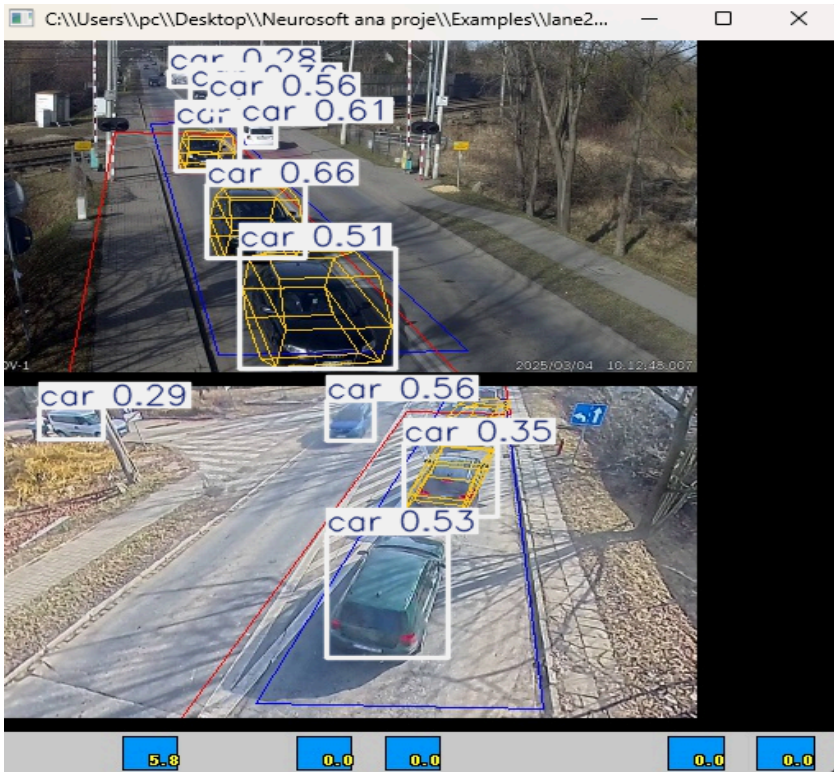
3.1. Input Data Preparation

- In this project, we utilized a diverse set of traffic camera images collected from various urban locations across the city. These images include scenes from parking areas, inner-city streets, and highways, providing a wide range of environmental and vehicle configurations. The input data was selected to reflect real-world variability in lighting conditions, object scales, and scene complexity. Each image serves as the starting point for our processing pipeline, which involves depth estimation, object detection, 3D reconstruction, and comparison with reference vehicle models in GLB format. This diverse dataset helps ensure robustness and generalizability of the 3D vehicle analysis system.

3.2. Object Detection (YOLOv8-Large)

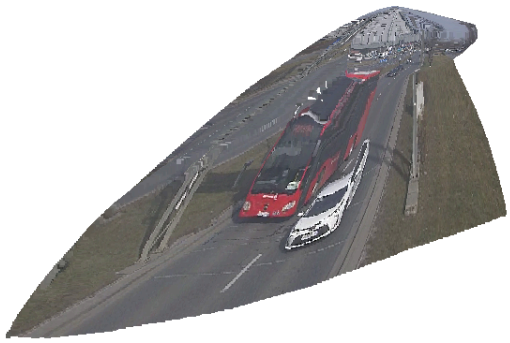


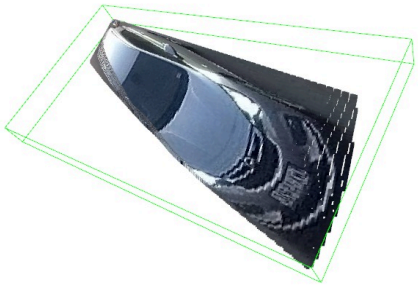
0



3.3. Depth Estimation (MiDaS)

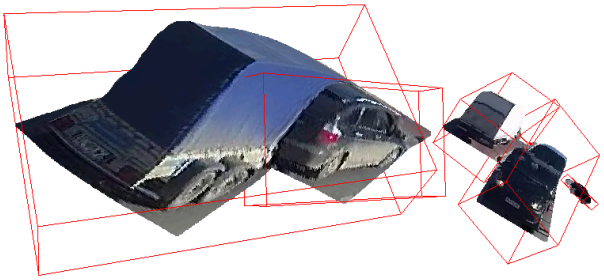
3.4. Point Cloud Generation



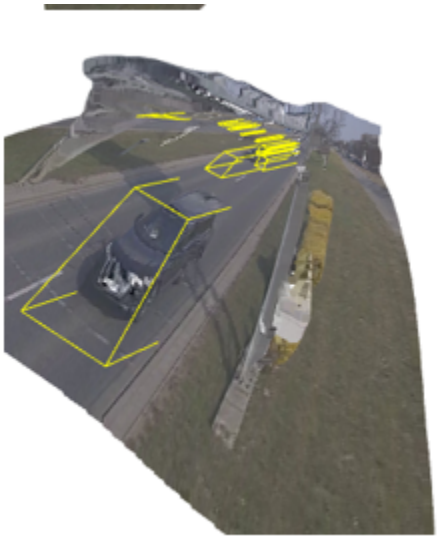


4. Bounding Box Construction

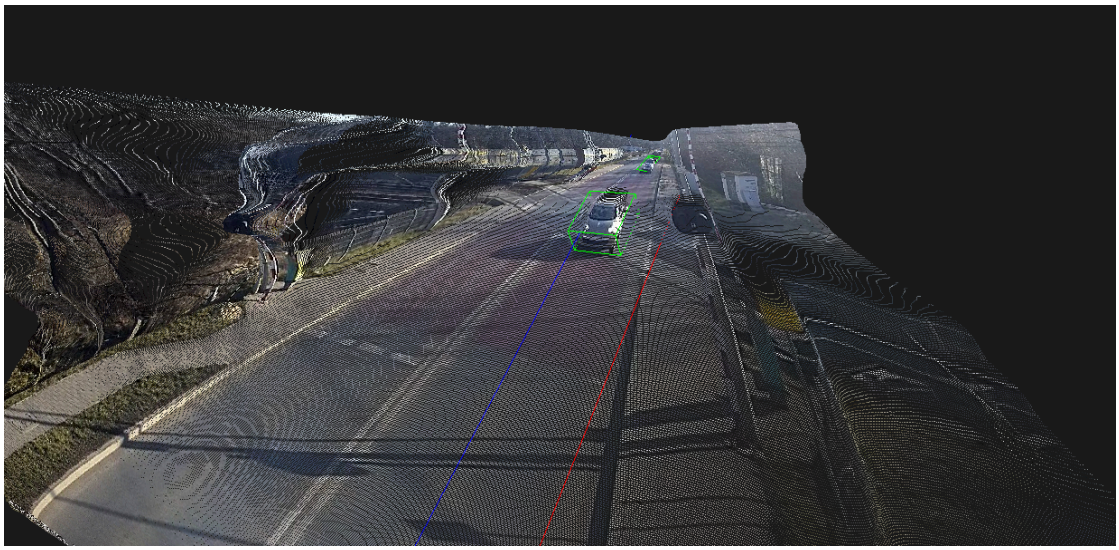
- First Outputs That We Got

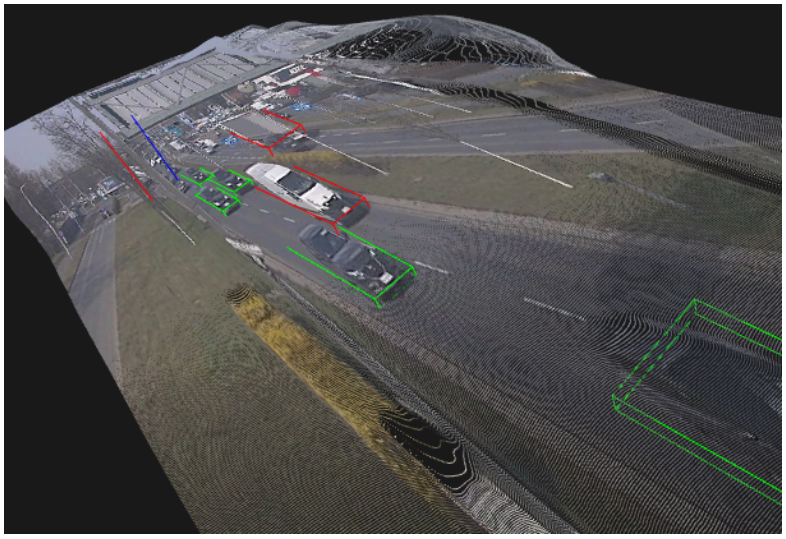


- The process was hard for us but boxes became more smooth week by week



- Final Version





5. Scene Segmentation and Area Marking

Classification of Regions:

- **Streets/Highways:** Primary road areas where vehicles are in motion or aligned. These regions serve as the basis for determining object orientation and placement.

- **Parking Spaces:** Zones with stationary vehicles; used to analyze spacing and verify the alignment of 3D bounding boxes.
  .

Methods Used:

This project employs several mathematical methods across the 3D object detection and analysis pipeline:

Camera Projection and Intrinsic Transformations

- **Pinhole Camera Model**:
  Used to back-project 2D image coordinates and estimated depth values into 3D space with the formula:

$$X = \frac{(x - c_x) \cdot Z}{f_x}, \quad Y = \frac{(y - c_y) \cdot Z}{f_y}$$

where (fx,fy) are focal lengths and (cx,cy) are principal points.

Homogeneous Coordinate Transformation

- **4x4 Extrinsic Matrix Multiplication**:
  Used to convert 3D points between camera and world coordinate frames.

Point Cloud Processing and Filtering

- Depth thresholding, percentile based clipping , and brightness masking are used to clean noisy depth data.

Principal Component Analysis (PCA)

- **Eigen Decomposition of Covariance Matrix**:
  Applied to 3D point clusters to determine object orientation by calculating eigenvectors of the covariance matrix of centered points.

Oriented Bounding Box (OBB) Estimation

- After PCA, the bounding box dimensions are obtained using:

$$\text{Extent} = \max_{local} - \min_{local}$$

Singular Value Decomposition (SVD)

- Used to orthonormalize rotation matrices for world coordinate conversion:

$$R = U \cdot V^T$$

Distance and Size Error Calculation

- **Euclidean Norm**:
  To evaluate the center offset and dimensional mismatch between predicted boxes and GLB references:
- It will show on the table after each calculation

2D Lane Detection using Edge-Based Hough Transform

- Mathematical line fitting via Hough Transform is used to identify lane markings based on Canny edge detection and slope filtering.

6. Real-Time Performance & FPS

6.1 Use of AI Accelerators

The system leverages GPU acceleration for computationally intensive tasks such as YOLOv8x object detection and DPT-Large monocular depth estimation. When a CUDA-enabled GPU is available, inference times are significantly reduced, allowing the pipeline to benefit from hardware-level parallelism.

- **YOLOv8x** provides efficient multi-class object detection.

- **DPT-Large** uses a transformer-based architecture optimized for depth prediction on GPU.

- Mathematical operations like PCA, covariance analysis, and coordinate transformations are vectorized using NumPy and PyTorch.

6.2 Computational Requirements and FPS

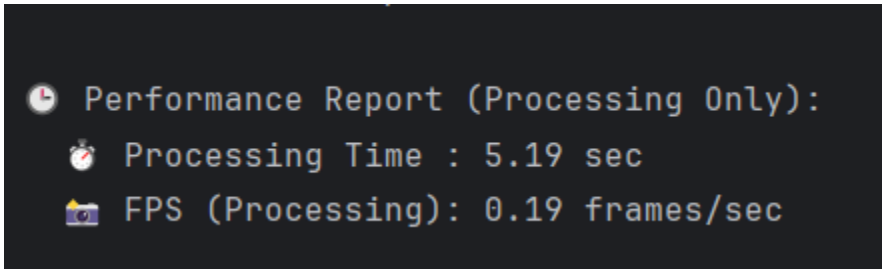
Initial performance evaluation was conducted on a **some high-resolution images**, resulting in the average following processing times:

🕑 **Total Pipeline Time (1 image):** 5-6 seconds

📸 **Approximate FPS:** 0.15-0.20 frames/second (Which means perfect for batch analysis

This performance ensures that the system can operate in near real-time scenarios, especially for batch image analysis or slow-moving environments like urban surveillance.

OUTPUT



7. Output Evaluation

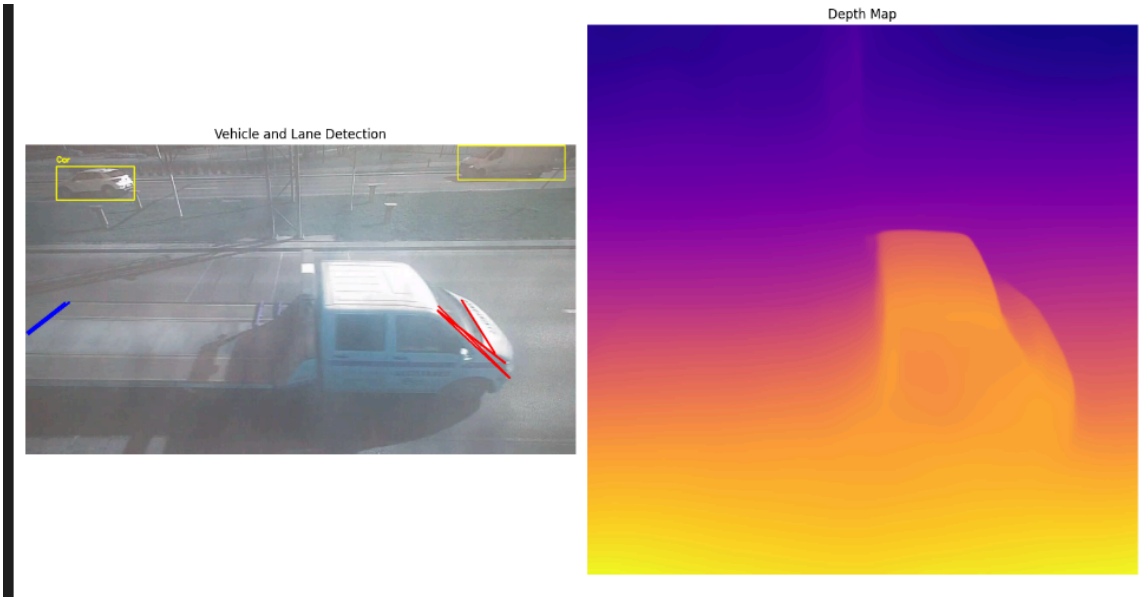
7.1. Highways and Streets

- Dimensions and Glbee’s

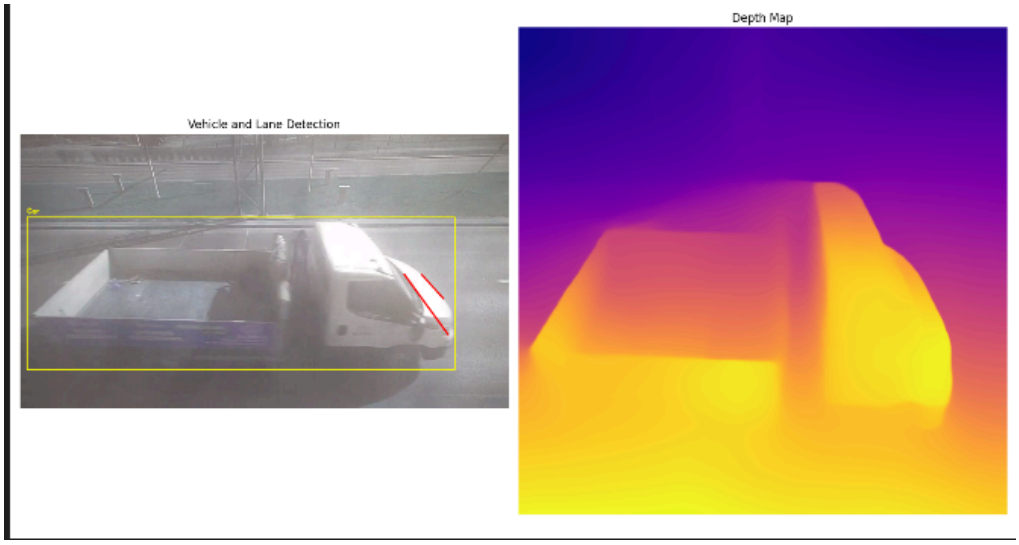
```
Box 1:
  Predicted size: [       8.81        3.32        0.82]
  GLB reference : [      10.21        2.37        1.78]
  Size error    : [        1.4        0.95        0.96]
```

```
Box 0:
  Predicted size: [      16.39        2.16        1.24]
  GLB reference : [      11.93        3.71        3.14]
  Size error    : [       4.46        1.56         1.9]
Showing 3D scene with GLB comparison
```

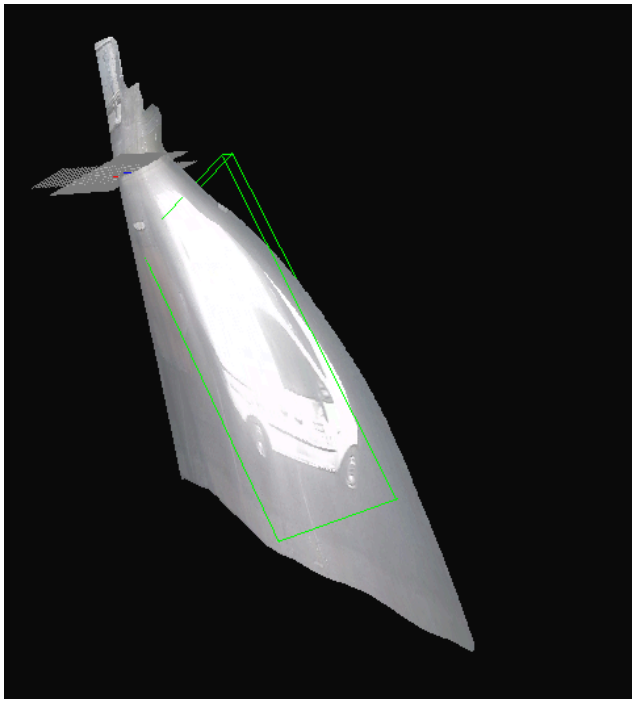
WRONG ONE



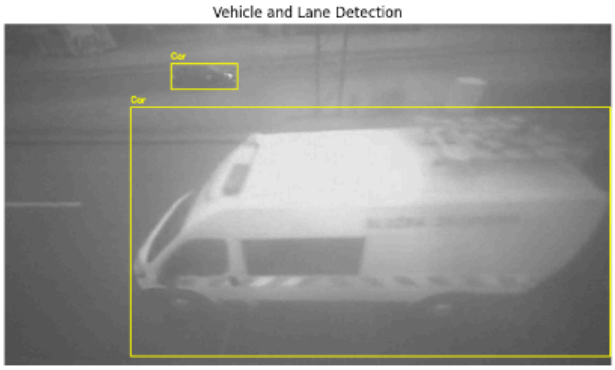

```
Box 0:
  Predicted size: [      31.11       19.15        2.28]
  GLB reference : [       8.45        2.59        2.32]
  Size error    : [      22.66       16.56        0.04]

Box 1:
  Predicted size: [      39.92        9.49         2.5]
  GLB reference : [       8.45        2.59        2.32]
  Size error    : [      31.47         6.9        0.18]
Showing 3D scene with GLB comparison...
```

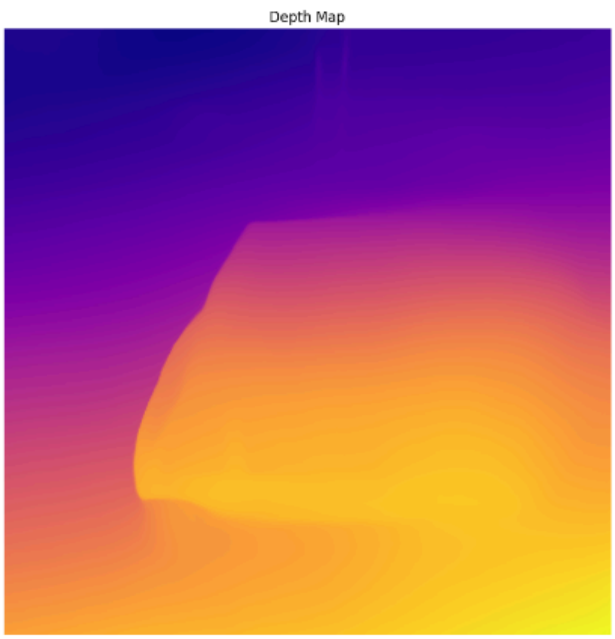



```
Box 0:
  Predicted size: [      6.43        3.49        2.71]
  GLB reference : [      6.42        2.49        2.58]
  Size error    : [      0.01        1.00        0.13]
```

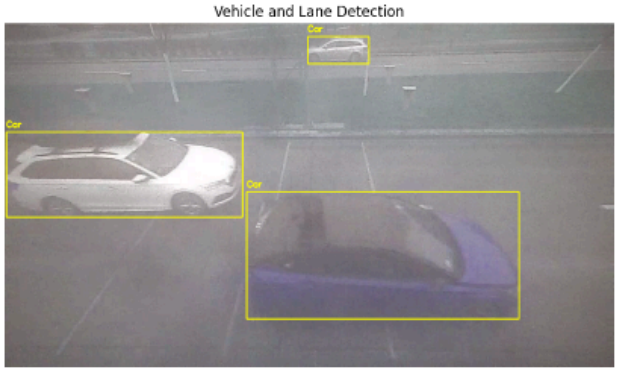


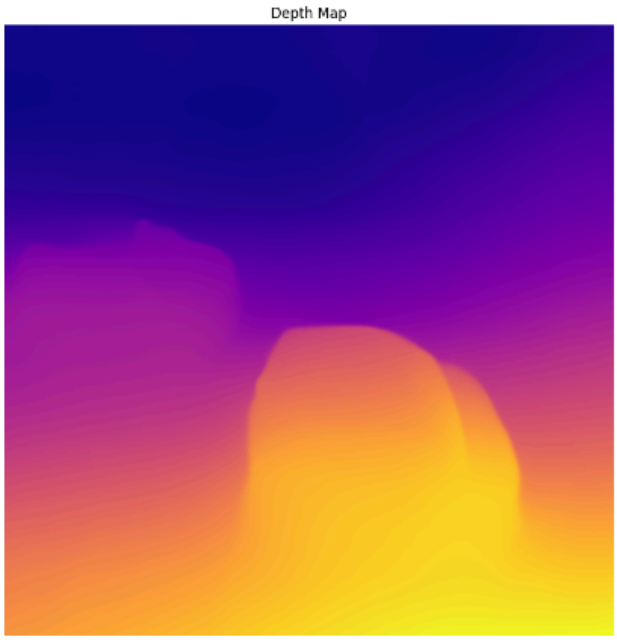

```
Box 0:
  Predicted size: [      87.54       44.51        3.45]
  GLB reference : [      10.85        2.14        1.94]
  Size error    : [      76.69       42.37        1.51]
```





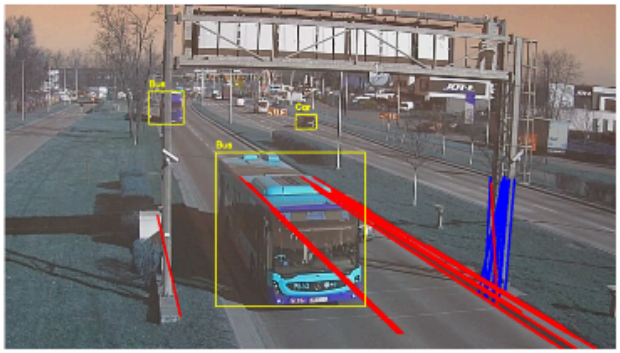

```
GLB object dimensions (W x H x L): [      11.84        2.83        2.62] meters
GLB object dimensions (W x H x L): [     11.843      2.8272      2.6207] meters

Box 0:
  Predicted size: [       2.66        1.24        0.81]
  GLB reference : [      11.84        2.83        2.62]
  Size error    : [       9.19        1.59        1.81]
```

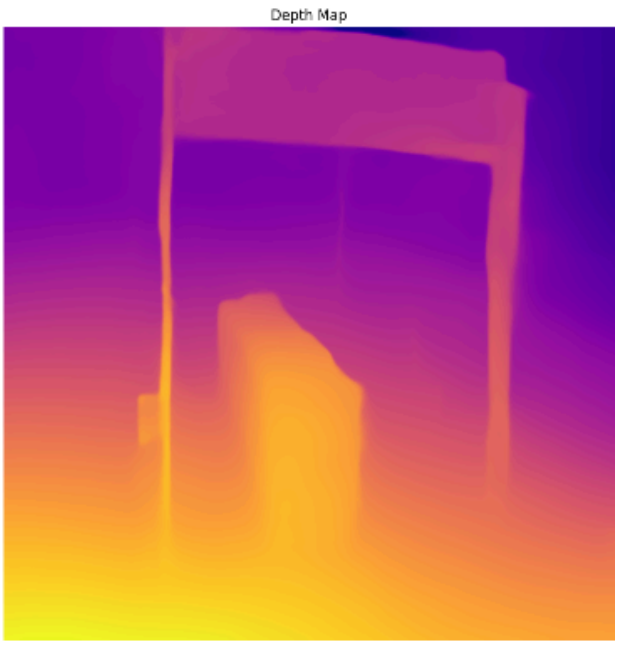




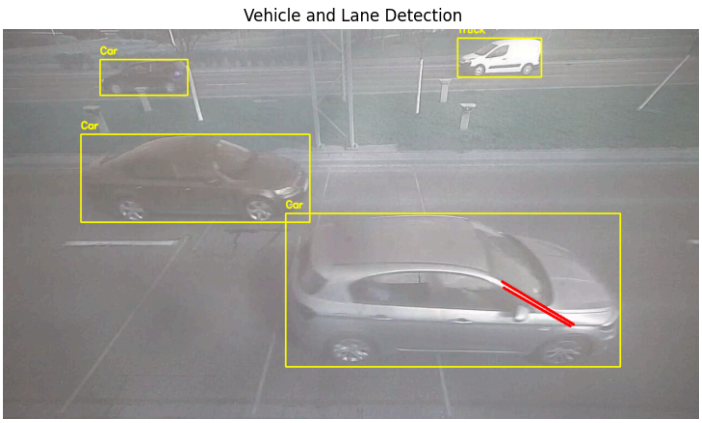
```
Box discarded due to extreme size: [      30.48       26.48        2.53]
[Camera] Box 0 center: [       8.04      -12.75        4.83]
Box discarded due to extreme size: [     107.25       58.86        4.35]
[Camera] Box 0 center: [       8.04      -12.75        4.83]
1 box(es) will be visualized.
```

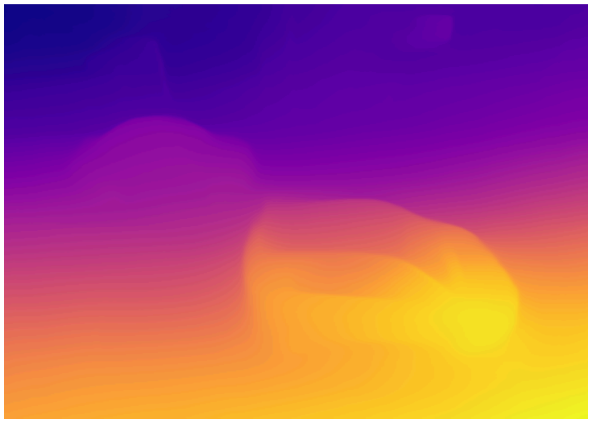
```
Box 0:
  Predicted size: [       4.32        3.77        0.13]
  GLB reference : [       9.76        1.84        1.49]
  Size error    : [       5.44        1.94        1.36]
```





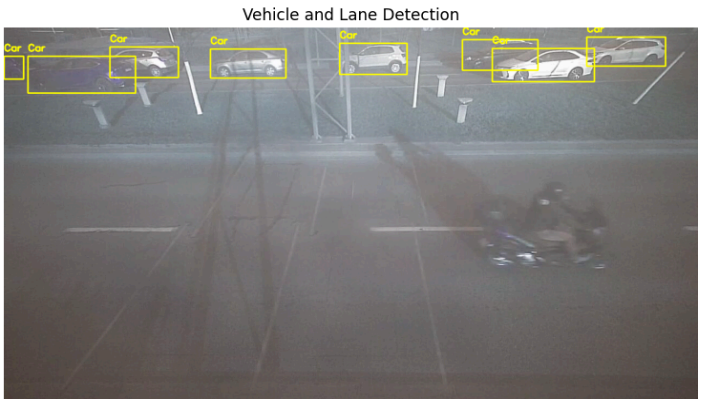

```
================================================================================
VEHICLE DIMENSIONS SUMMARY
================================================================================
#   Type         Length    Width     Height    Volume      Points
--------------------------------------------------------------------------------
1   Bus          19.10     10.87     2.62      544.85      4850
2   Bus          9.08      2.62      2.62      62.57       1209
3   Car          4.20      1.89      1.47      11.67       288
================================================================================
All dimensions in meters, volume in cubic meters
```



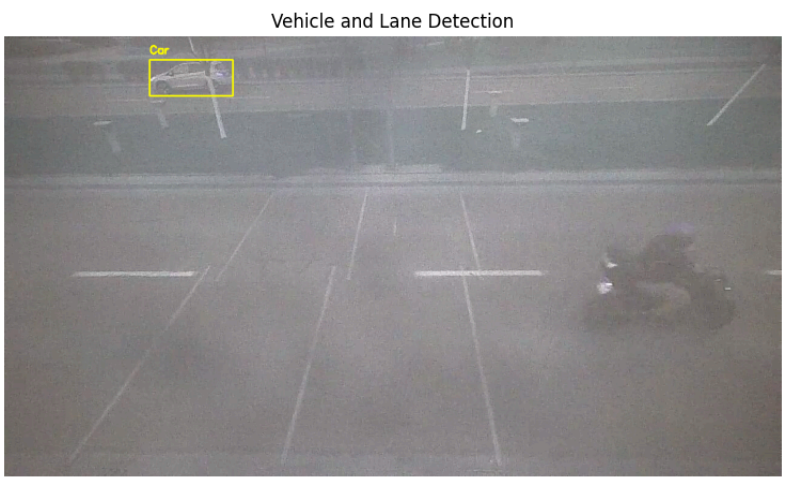


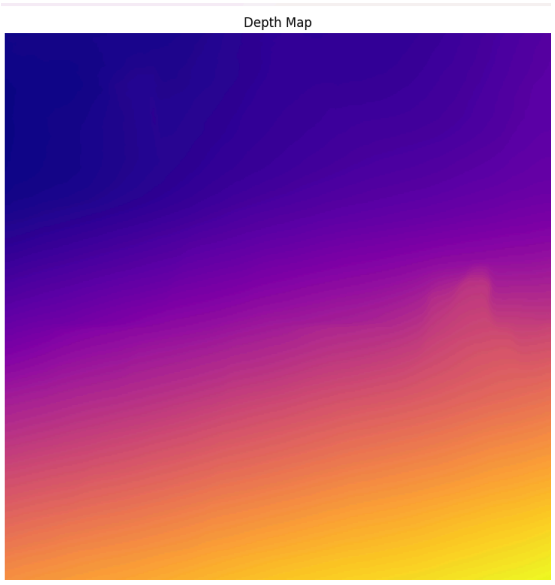
```
Loading GLB object: C:\Users\caner\Desktop\man\design\data\glb\glb2\7\VehicleTrace-20250325-123644-863-pl-wrozmigrodzka242-cen-l1-vehicle.glb
GLB object dimensions (W x H x L): [       5.36        1.88        1.56] meters
GLB object dimensions (W x H x L): [     5.3642      1.8765      1.5567] meters

Box 0:
  Predicted size: [       6.33        6.83        0.55]
  GLB reference : [       5.36        1.88        1.56]
  Size error    : [       0.97        4.95        1.01]

Box 1:
  Predicted size: [       4.92        2.37        0.78]
  GLB reference : [       5.36        1.88        1.56]
  Size error    : [       0.45        0.49        0.78]
Showing 3D scene with GLB comparison...
Visualizing 3D detection...
Creating ground plane...
```

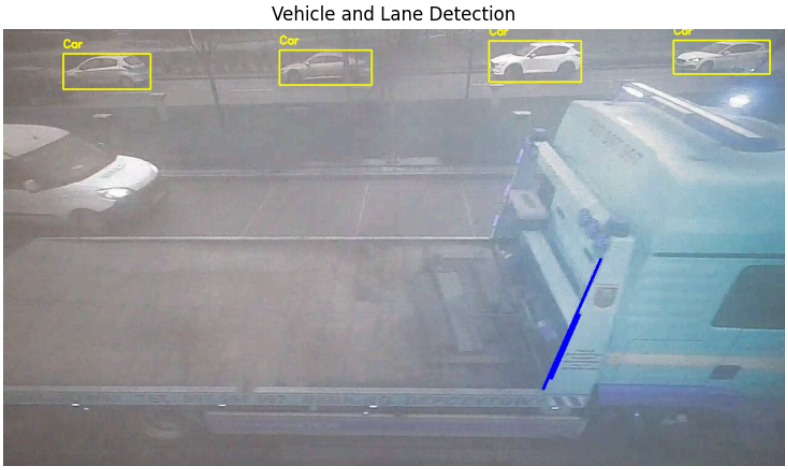
For motorcycle (Wrong one)

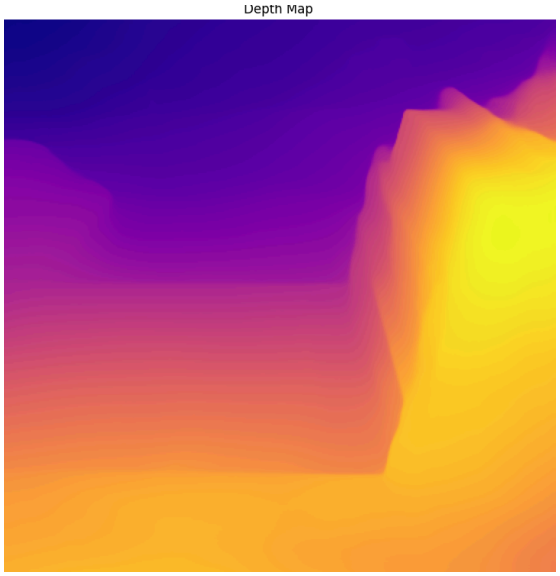



```
Box 1:
  Predicted size: [       4.03        3.15        0.86]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [       1.81        1.88        0.01]

Box 2:
  Predicted size: [       2.31        1.45        0.67]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [       0.09        0.18         0.2]

Box 3:
  Predicted size: [       8.57        3.09        0.73]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [       6.36        1.81        0.14]
```



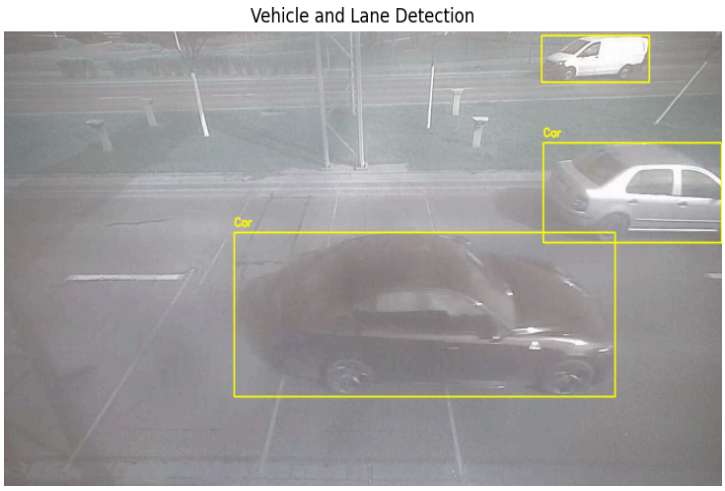


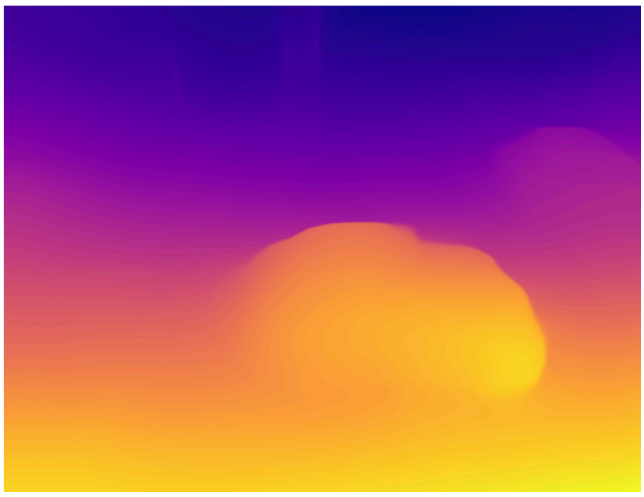




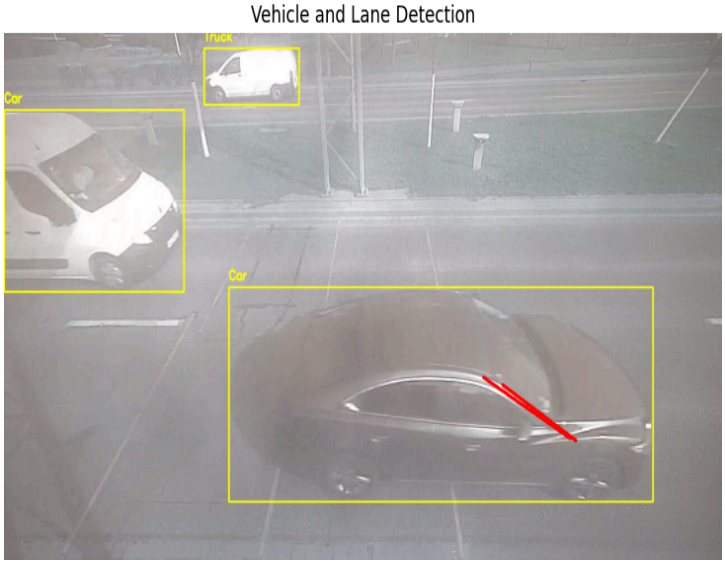

```
Creating 3D bounding boxes...
[Camera] Box 0 center: [    14.27    -15.51      6.04]
[Camera] Box 1 center: [    24.84    -18.64      6.76]
[Camera] Box 2 center: [      4.2    -12.25      4.84]
[Camera] Box 3 center: [    -2.51     -8.78       3.7]
[Camera] Box 0 center: [    14.27    -15.51      6.04]
[Camera] Box 1 center: [    24.84    -18.64      6.76]
[Camera] Box 2 center: [      4.2    -12.25      4.84]
[Camera] Box 3 center: [    -2.51     -8.78       3.7]
√ 4 box(es) will be visualized.
```

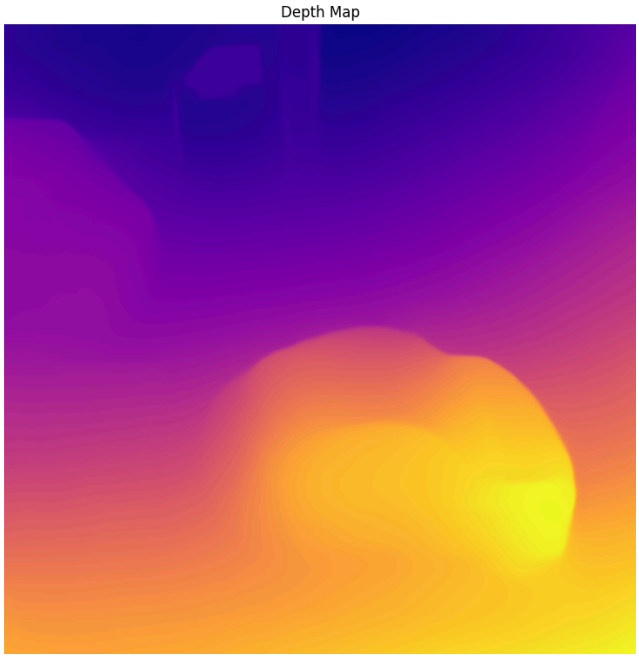
```
Box 0:
  Predicted size: [      6.05      3.78       0.4]
  GLB reference : [     31.26      4.14      3.42]
  Size error    : [      25.2      0.36      3.02]

Box 1:
  Predicted size: [      9.08      3.42      0.38]
  GLB reference : [     31.26      4.14      3.42]
  Size error    : [     22.17      0.72      3.04]

Box 2:
  Predicted size: [      4.19      2.76      0.18]
  GLB reference : [     31.26      4.14      3.42]
  Size error    : [     27.07      1.38      3.24]

Box 3:
  Predicted size: [      3.14       1.3      0.36]
  GLB reference : [     31.26      4.14      3.42]
  Size error    : [     28.12      2.84      3.06]
```



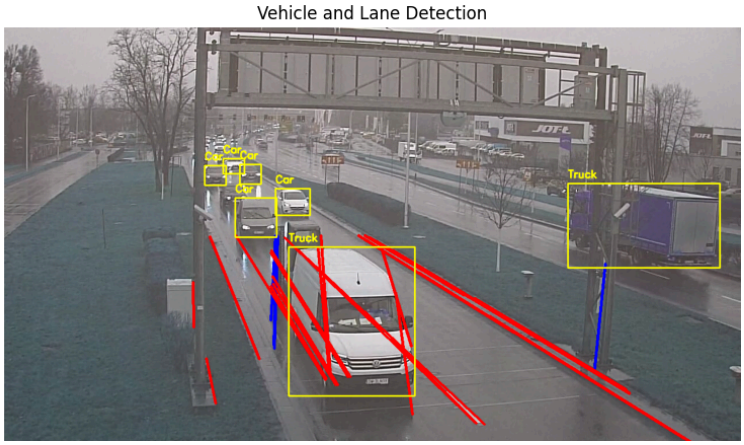


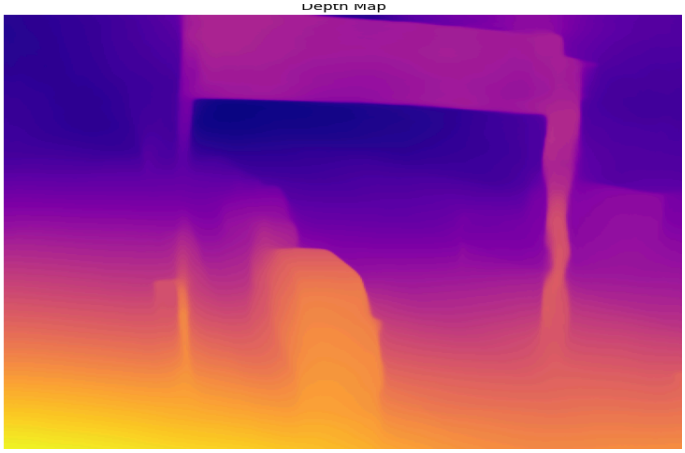
```
Box 0:
  Predicted size: [        6.62          3.73           0.3]
  GLB reference : [        5.65          1.99          1.51]
  Size error    : [        0.97          1.74          1.21]
```





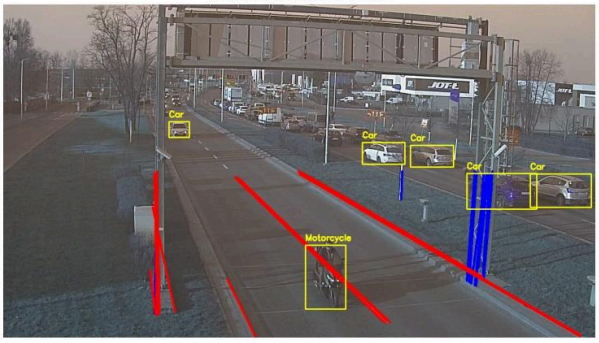

```
Box 0:
  Predicted size: [        6.11          4.56          0.92]
  GLB reference : [         5.7           1.9          1.61]
  Size error    : [        0.41          2.66          0.69]
```



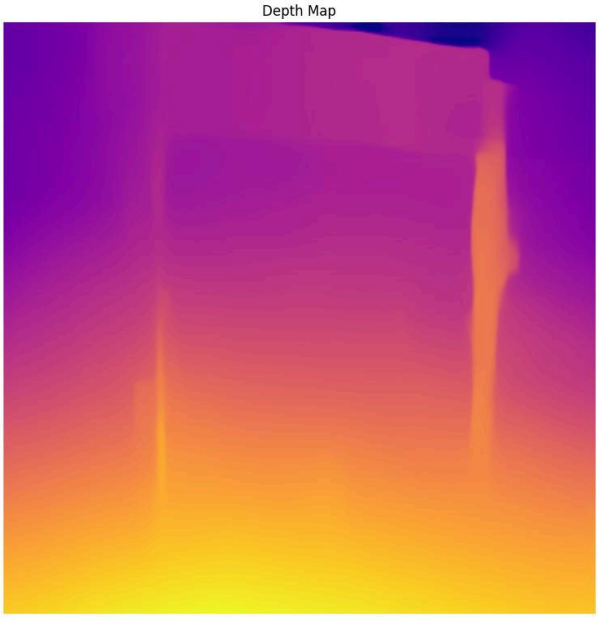



```
Box 0:
  Predicted size: [      8.41        2.78        0.8]
  GLB reference : [     12.21           3        2.2]
  Size error    : [       3.8        0.21        1.4]

Box 1:
  Predicted size: [     11.71        4.34       0.38]
  GLB reference : [     12.21           3        2.2]
  Size error    : [       0.5        1.34       1.81]

Box 2:
  Predicted size: [       3.9        1.35       0.38]
  GLB reference : [     12.21           3        2.2]
  Size error    : [      8.32        1.65       1.82]

Box 3:
  Predicted size: [      3.95        1.53       0.31]
  GLB reference : [     12.21           3        2.2]
  Size error    : [      8.26        1.47       1.89]

Box 4:
  Predicted size: [      3.03        1.45       0.26]
  GLB reference : [     12.21           3        2.2]
  Size error    : [      9.18        1.55       1.94]
```

Vehicle and Lane Detection

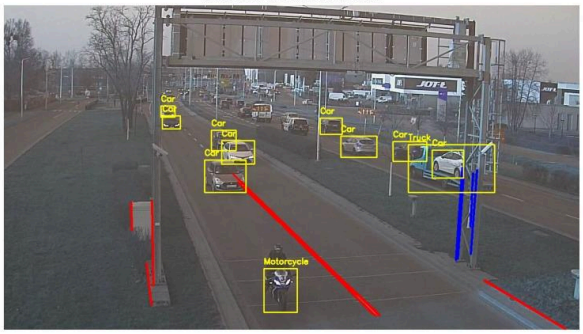


Depth Map

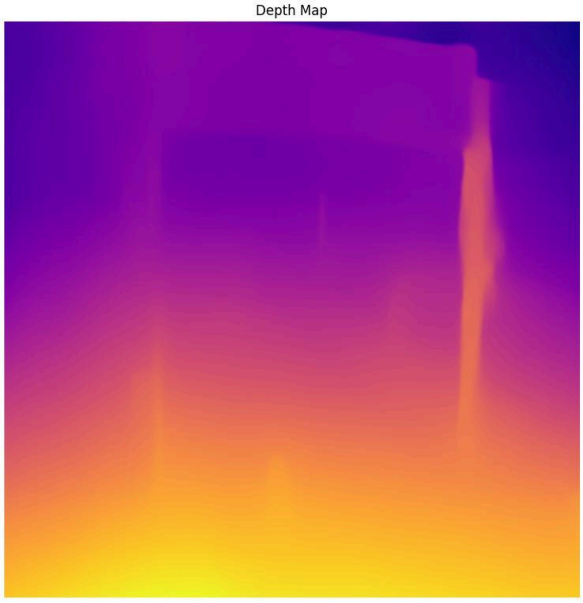


```
Box 0:
  Predicted size: [      13.98        2.96        0.42]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [      11.76        1.69        0.45]

Box 1:
  Predicted size: [       4.32        1.97        0.13]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [       2.11        0.69        0.74]

Box 2:
  Predicted size: [       8.27         3.8        0.14]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [       6.06        2.52        0.73]

Box 3:
  Predicted size: [       8.92        4.72         0.1]
  GLB reference : [       2.22        1.28        0.87]
  Size error    : [       6.71        3.45        0.77]
```

Vehicle and Lane Detection

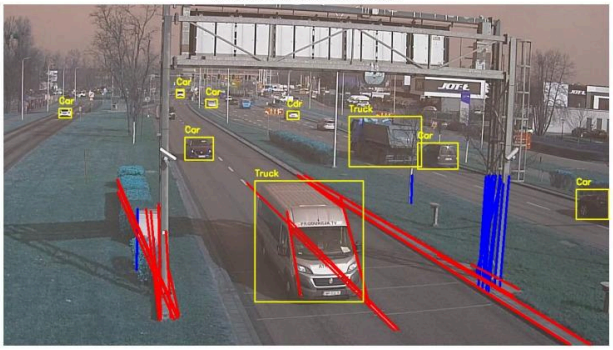


Depth Map

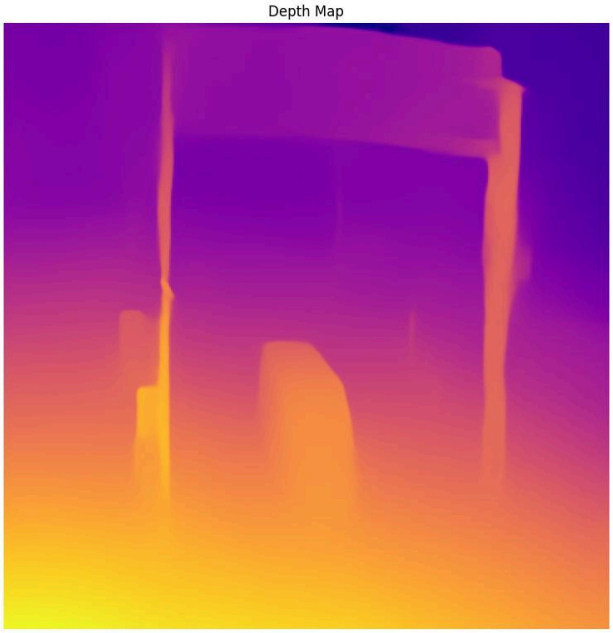


```
Box 0:
  Predicted size: [      12.58        4.64        0.18]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [      11.08        3.26        0.41]

Box 1:
  Predicted size: [       6.44        3.52         0.1]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [       4.94        2.14        0.48]

Box 2:
  Predicted size: [       8.39         3.3        0.11]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [       6.89        1.93        0.47]

Box 3:
  Predicted size: [       3.18        1.59        0.12]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [       1.68        0.22        0.47]

Box 4:
  Predicted size: [        5.6         2.2        0.06]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [        4.1        0.83        0.53]

Box 5:
  Predicted size: [       7.38        3.45        0.31]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [       5.88        2.07        0.27]

Box 6:
  Predicted size: [       2.54        1.03        0.22]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [       1.04        0.34        0.36]

Box 7:
  Predicted size: [      11.93        7.44         0.8]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [      10.44        6.06        0.21]

Box 8:
  Predicted size: [       4.57        1.53        0.51]
  GLB reference : [        1.5        1.37        0.59]
  Size error    : [       3.07        0.16        0.08]
```

Vehicle and Lane Detection

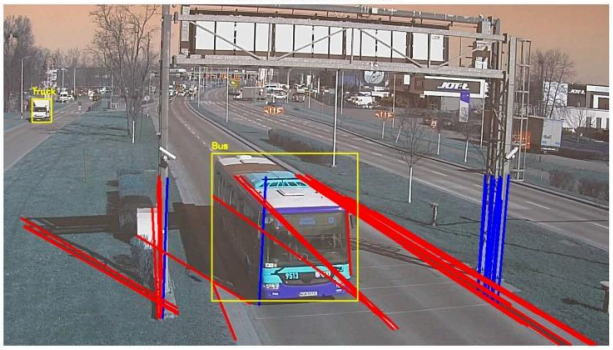


Depth Map

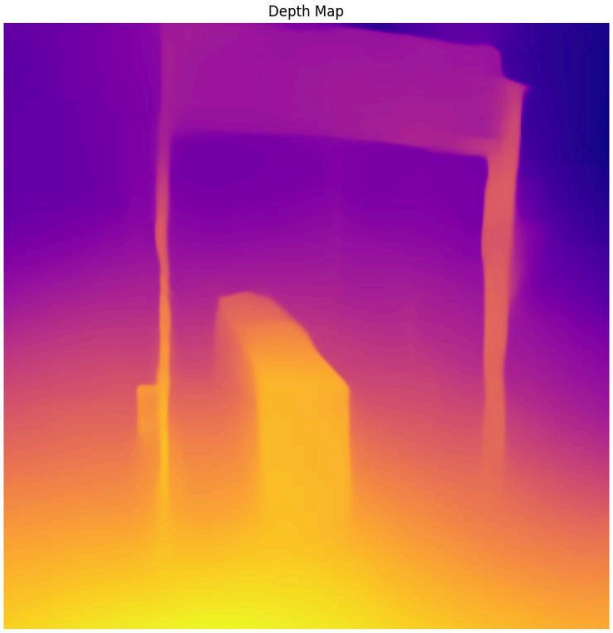


```
Box 0:
  Predicted size: [       1.23        0.69        0.02]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       6.58        2.13        2.36]

Box 1:
  Predicted size: [       1.87        0.84        0.02]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       5.95        1.99        2.36]

Box 2:
  Predicted size: [      13.51        5.07        0.37]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       5.69        2.25        2.01]

Box 3:
  Predicted size: [       6.26        2.36        0.15]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       1.55        0.46        2.23]

Box 4:
  Predicted size: [       8.24        2.62         0.2]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       0.43         0.2        2.19]

Box 5:
  Predicted size: [       1.21        0.43        0.08]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       6.61         2.4        2.31]

Box 6:
  Predicted size: [      10.65        1.46        0.12]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       2.83        1.37        2.26]

Box 7:
  Predicted size: [       1.29        0.69        0.02]
  GLB reference : [       7.81        2.83        2.38]
  Size error    : [       6.53        2.14        2.37]
```

Vehicle and Lane Detection

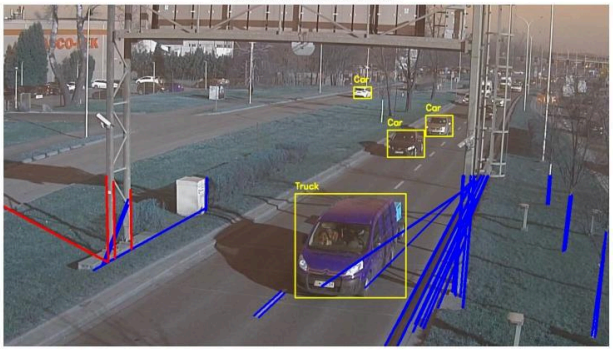


Depth Map

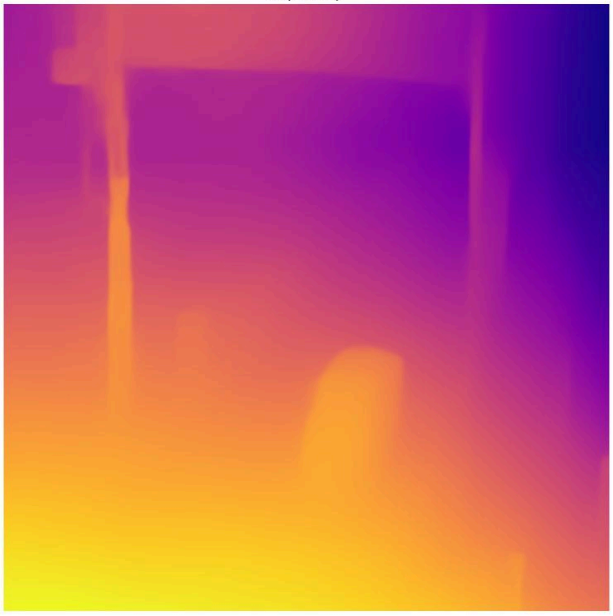


```
Box 0:
  Predicted size: [      3.89        1.06        0.13]
  GLB reference : [      13.4         3.1        2.71]
  Size error    : [      9.51        2.04        2.58]
```

Vehicle and Lane Detection

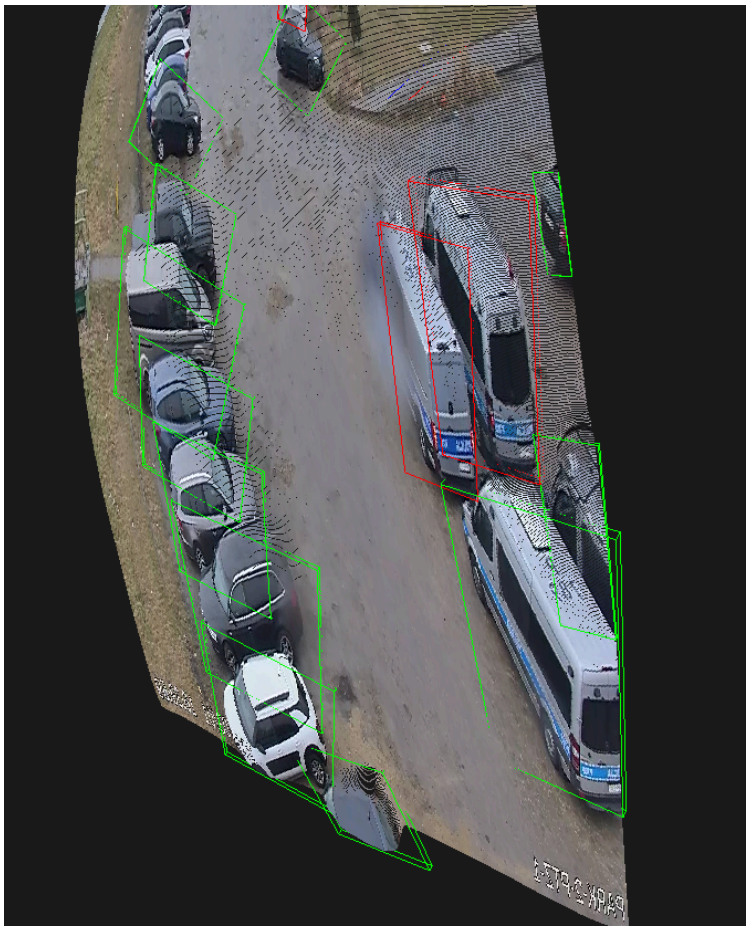


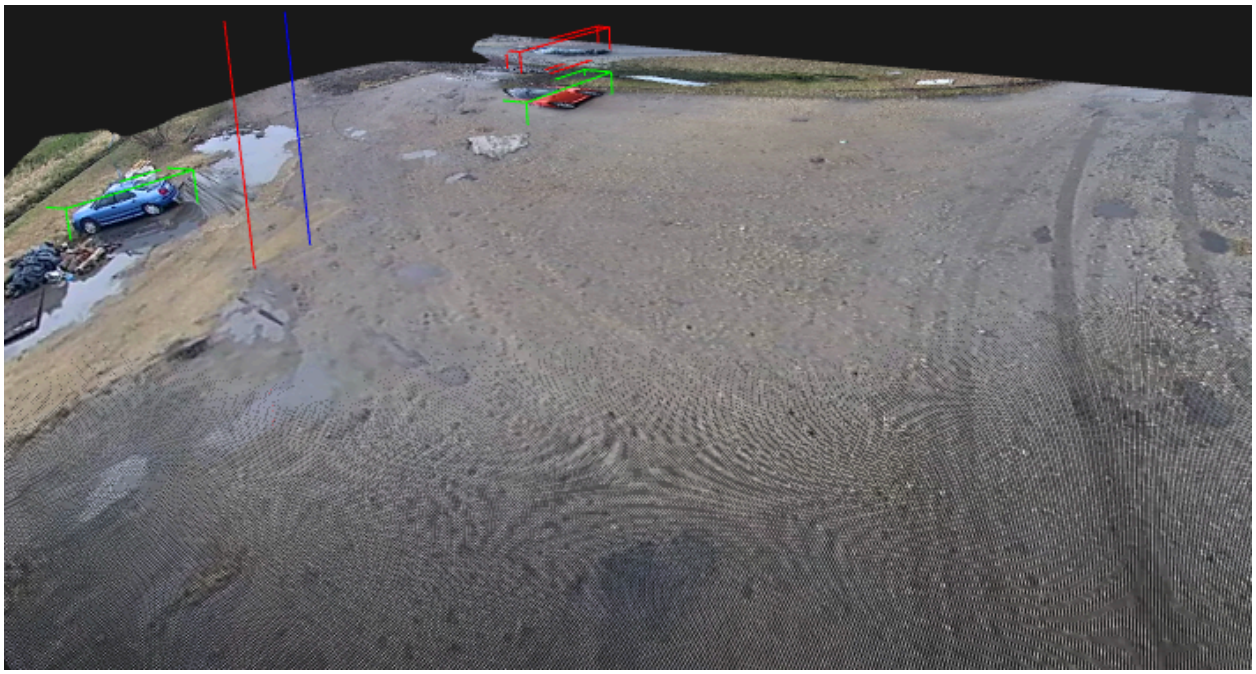
Depth Map



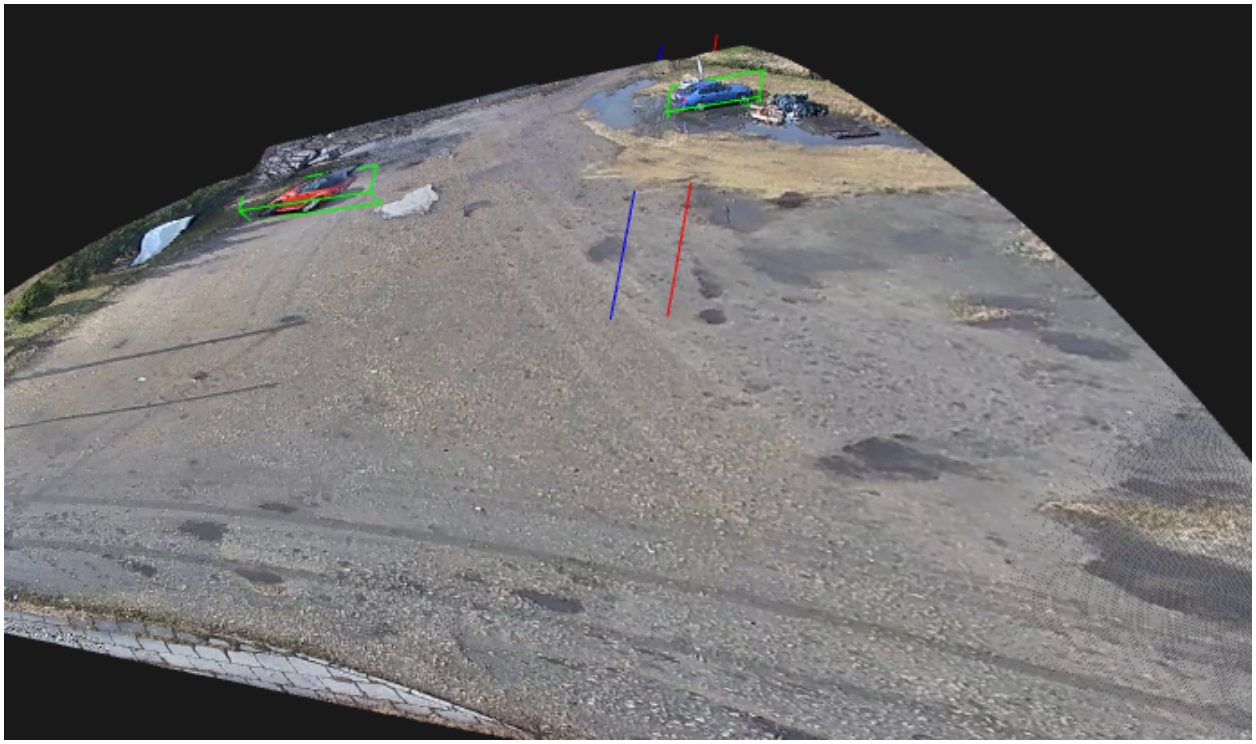
```
Box 0:
  Predicted size: [       9.1       2.29       0.11]
  GLB reference : [      8.42        2.1       2.02]
  Size error    : [      0.68       0.18       1.91]

Box 1:
  Predicted size: [      5.66       1.25        0.1]
  GLB reference : [      8.42        2.1       2.02]
  Size error    : [      2.76       0.85       1.92]

Box 2:
  Predicted size: [         3       1.14       0.07]
  GLB reference : [      8.42        2.1       2.02]
  Size error    : [      5.42       0.96       1.95]
```

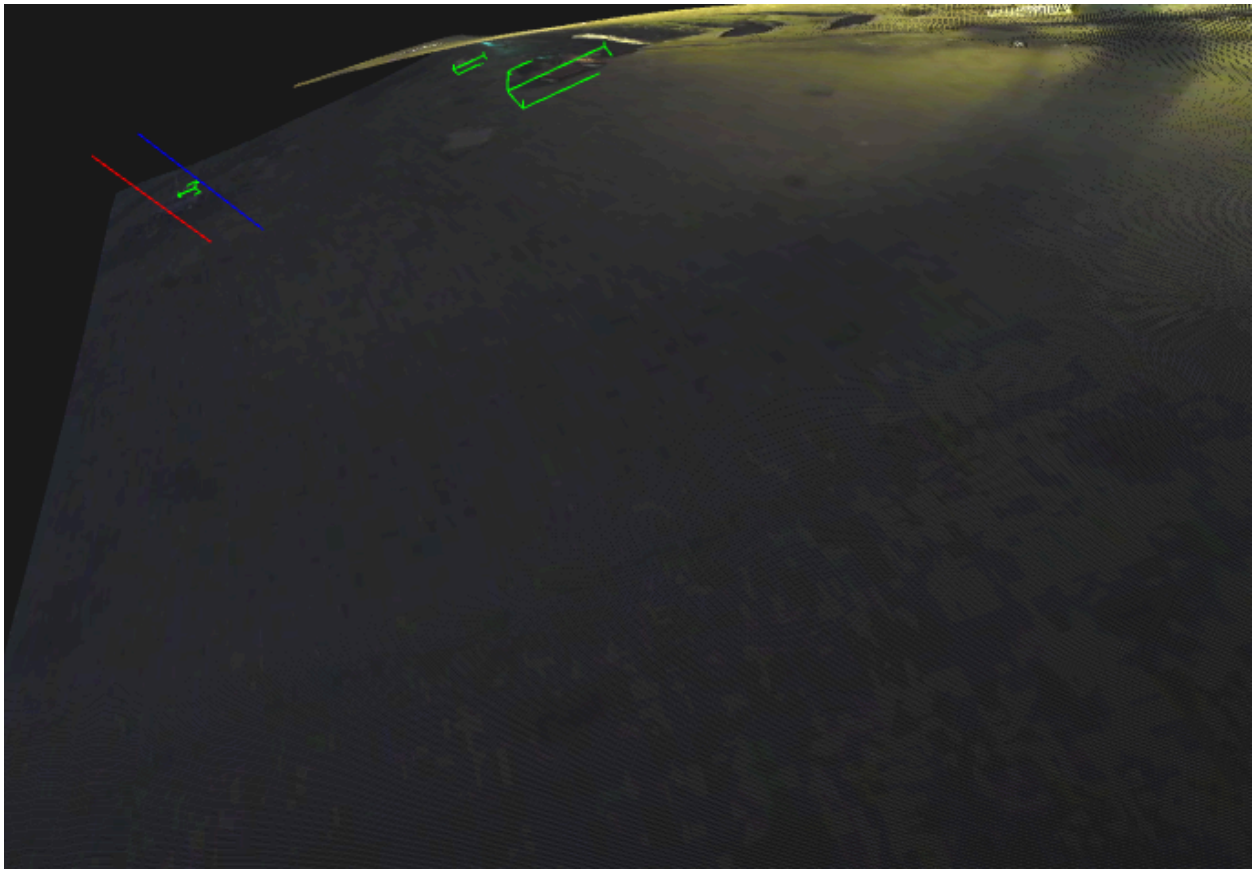
7.2. Parking Areas

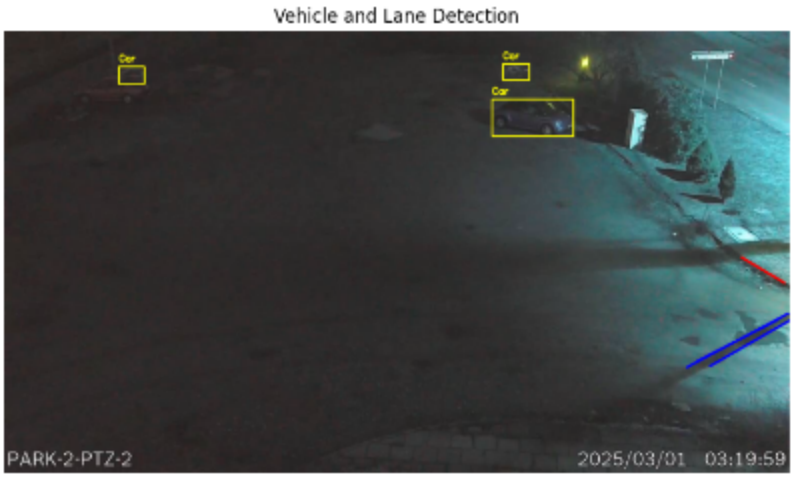




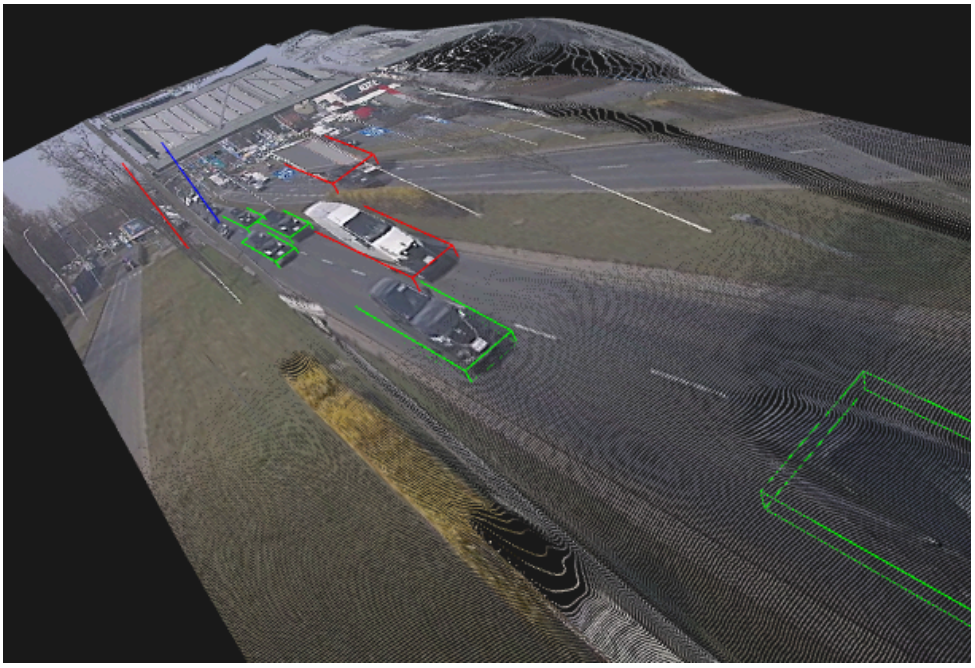
Vehicle and Lane Detection

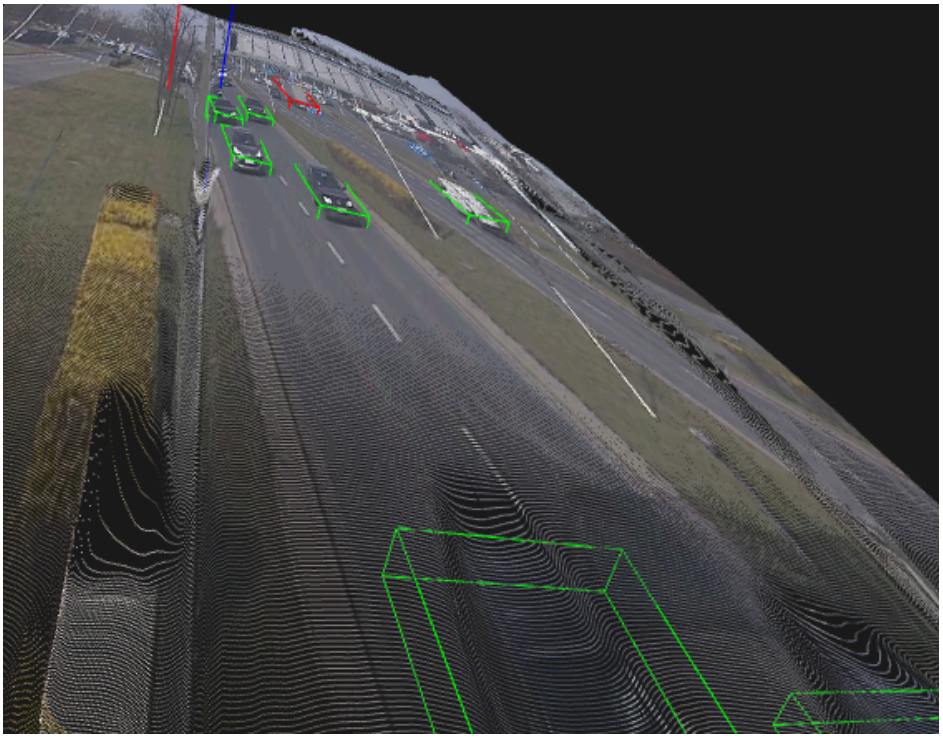


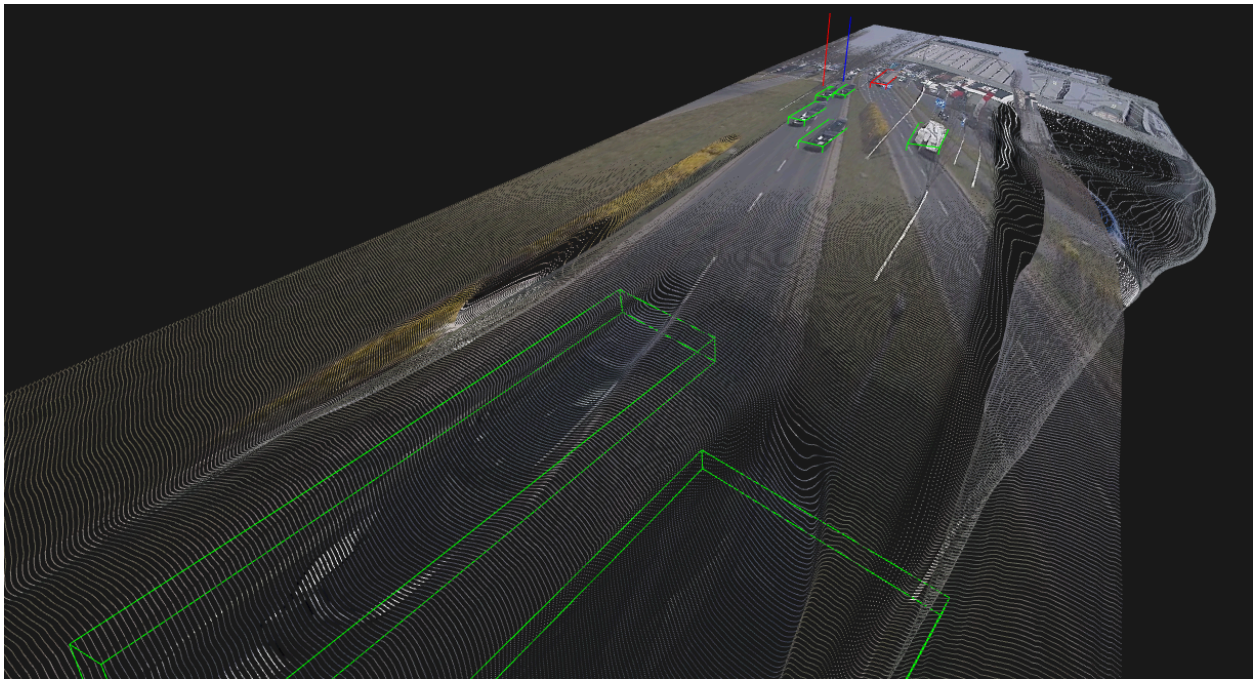


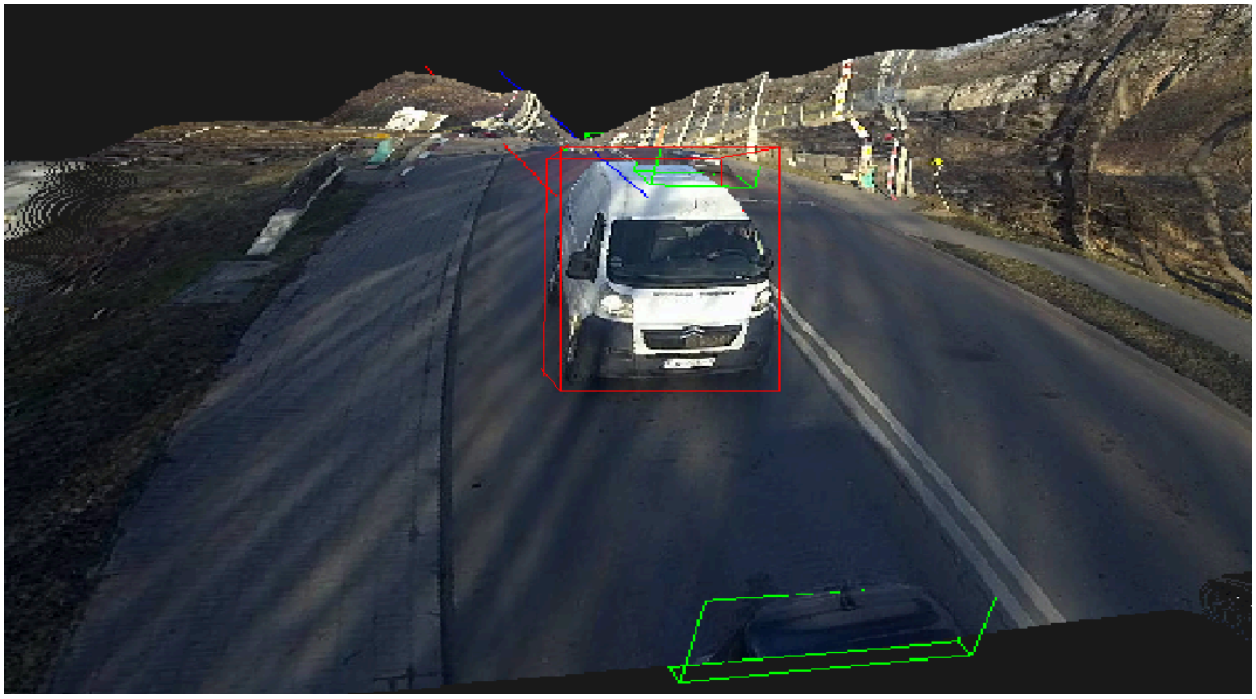


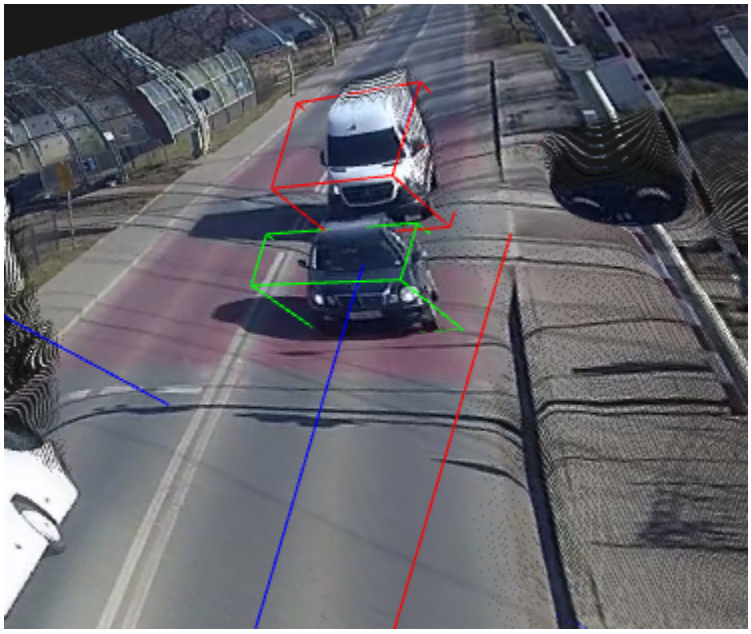
7.3 Highway Scan

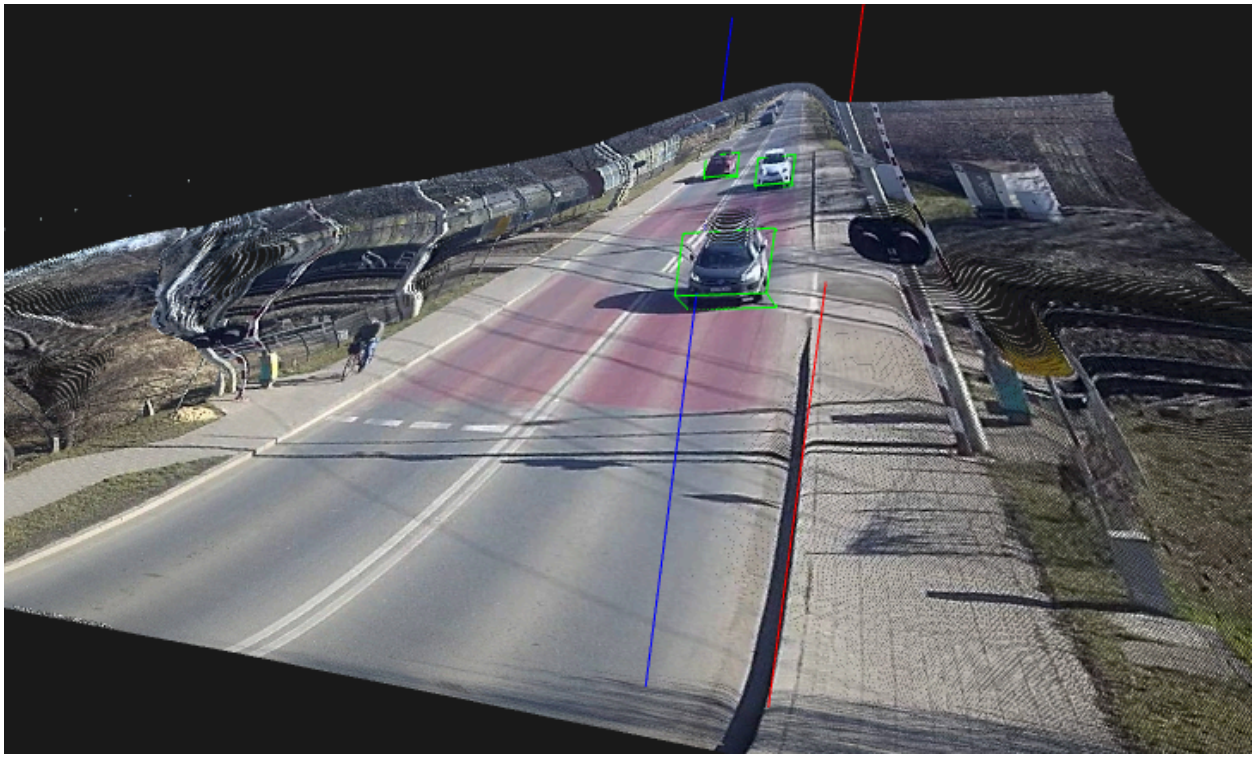












8.CONCLUSION

We started with basic CCTV images, calibration data, and initial detection outputs. Our goal was to achieve real-time 3D vehicle body estimation including position, orientation, and size. Through deep learning and 3D reconstruction techniques, we successfully developed an automated pipeline that converts 2D camera input into accurate 3D scene understanding — paving the way for scalable deployment on low-resource edge devices.