Developing an OEDR Module For Autonomous Vehicle Using Simulators

Authors:
Souvik Mandal and Tatavarthy Venkat Nikhil
B. Tech CSE Final Year
IIT Indore
Gourinath Banda
Associate Professor in CSE
IIT Indore

Problem Definition

- To develop an Object and Event Detection and Recognition module with:
  - Camera Sensor
  - LIDAR Sensor

  using the IPG CarMaker Simulator
- Collect video stream from camera sensors from IPG CarMaker Simulator

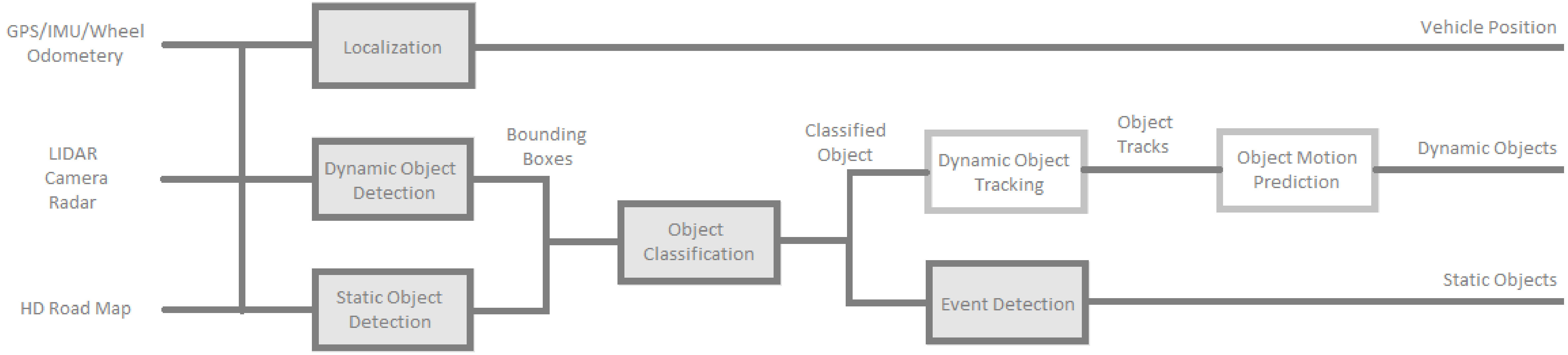
Motivation

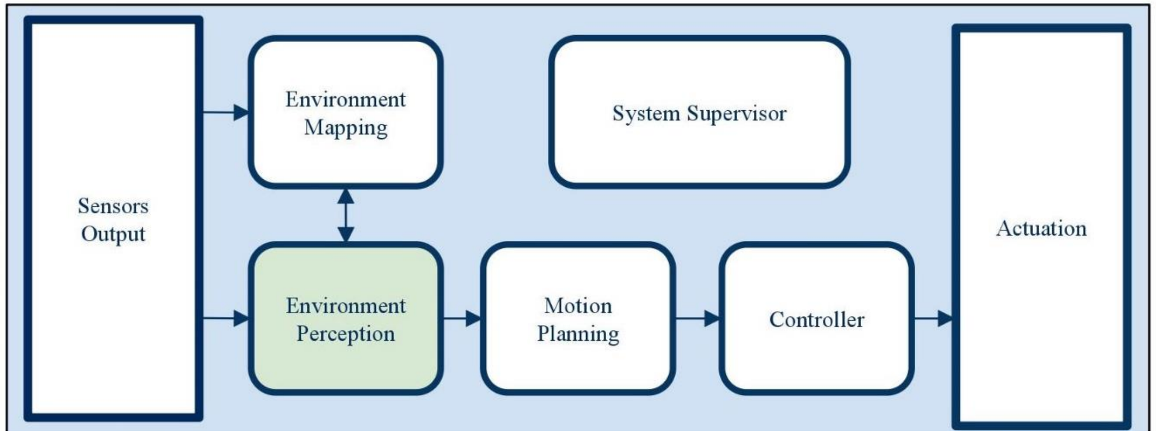
- Getting started in the absence of real time test environment
- OEDR module is important as a perception unit for autonomous cars
- IPG CarMaker simulator lets to define complex scenarios
- Simulators make it possible for virtual training and testing

Advantages with IPG-CarMaker

- Wide variety of environment/scenarios can be configured
- Allows for instantiating different sensor models
- Minimum dependency on hardware

Autonomous Driving





Project details

- ➢ We generated dynamic traffic with various traffic entities
- ➢ We have successfully collected camera sensor data from ego vehicle in real time
- ➢ We have processed collected data from the stereoscopic cameras in real-time
- ➢ We have implemented and integrated modules like
    - Static object identification and classification for traffic sign traffic light, lane, etc.
    - Dynamic object identification and classification for car, motorcycle, etc.
    - Event detection for traffic light and traffic sign state identification
    - Localization for ego vehicle’s relative position

Environment Setup

- Virtual Environment
    1. We have create several virtual scenarios in IPG CarMaker.
    2. We have configured the camera sensor for live processing of video stream.
- Sensor data collection from camera sensors

Traffic Scenarios modelled in IPG-CarMaker

- We created more than 20 virtual test environment with the combination of Car, Bus, Bicycle, Motorcycle, Pedestrian, Truck
- For example, in one virtual test environment we had following configuration

  Cars: 8, Bicycle: 12, Bike: 8, Pedestrian: 9

Object Detection Algorithms

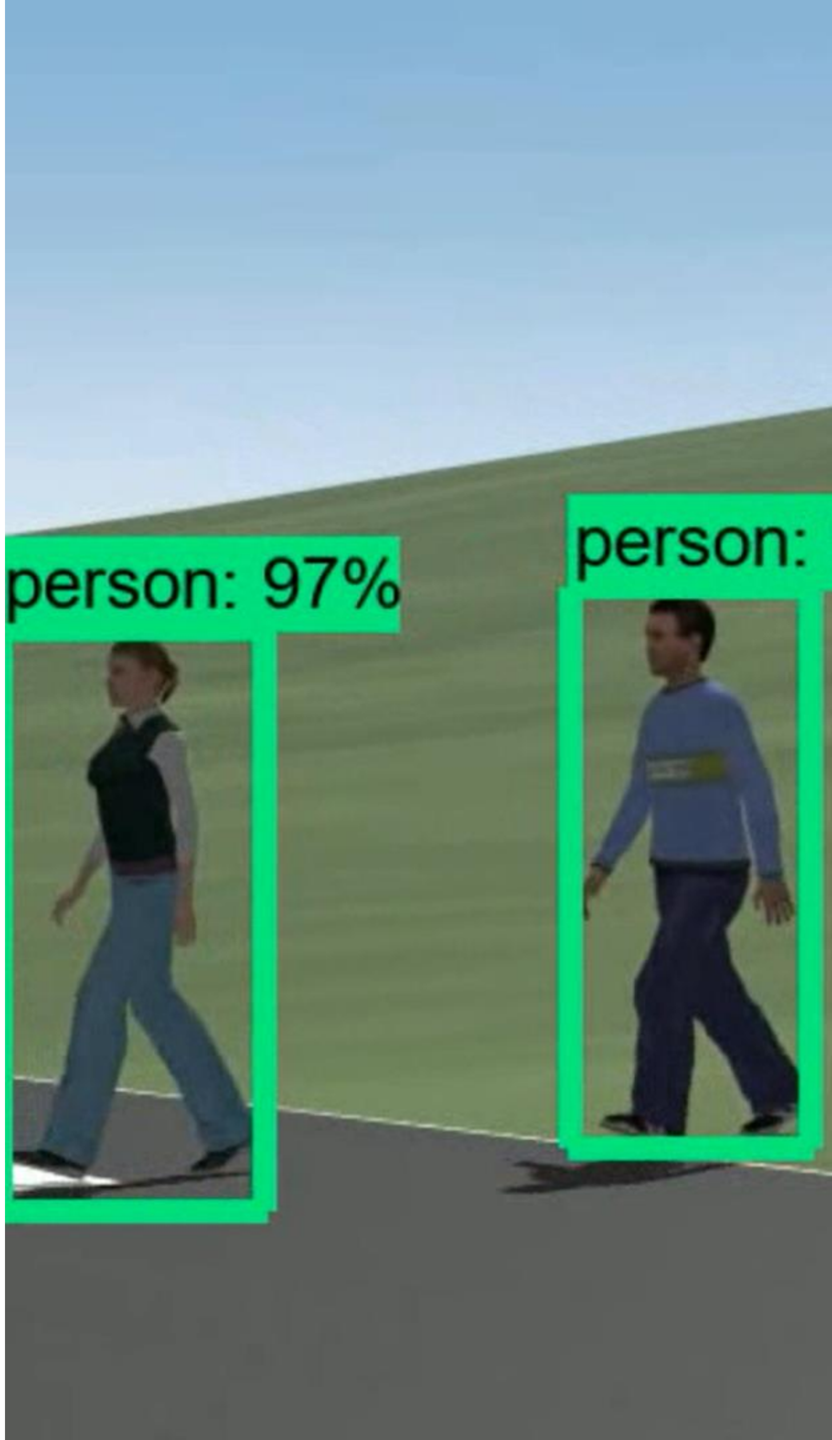
➢ Several real time object detection algorithms considered:

1. Mask R-CNN

   It is a new state of the art in field of instance segmentation. It extends faster R-CNN by adding a branch for predicting an object mask.

2. Single Shot Detector

   It takes one single shot to detect multiple objects within the image. It's much faster than Mask R-CNN but Mask R-CNN has much higher precision.

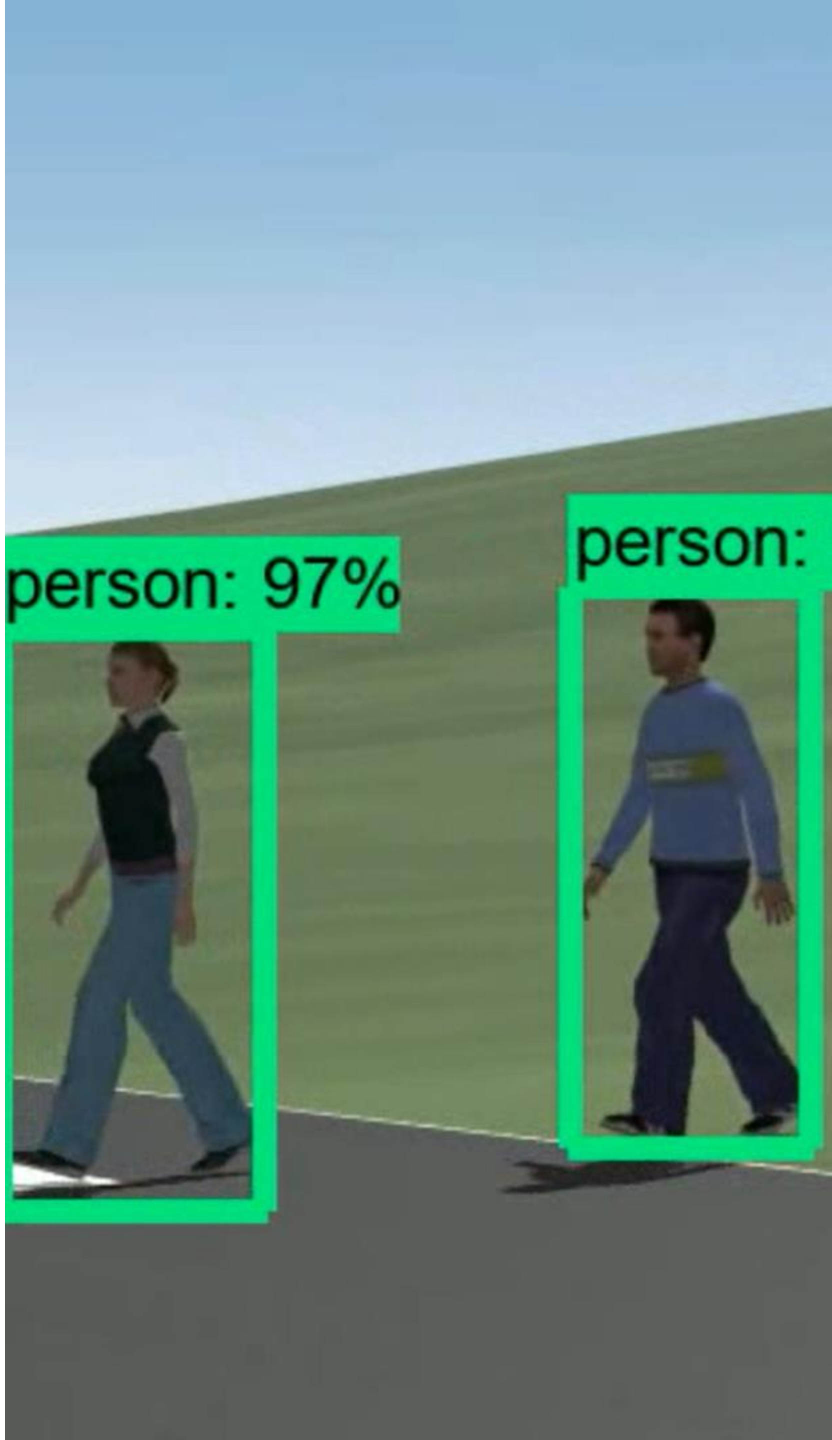


Object Detection Algorithms

- Several real time object detection algorithms considered:

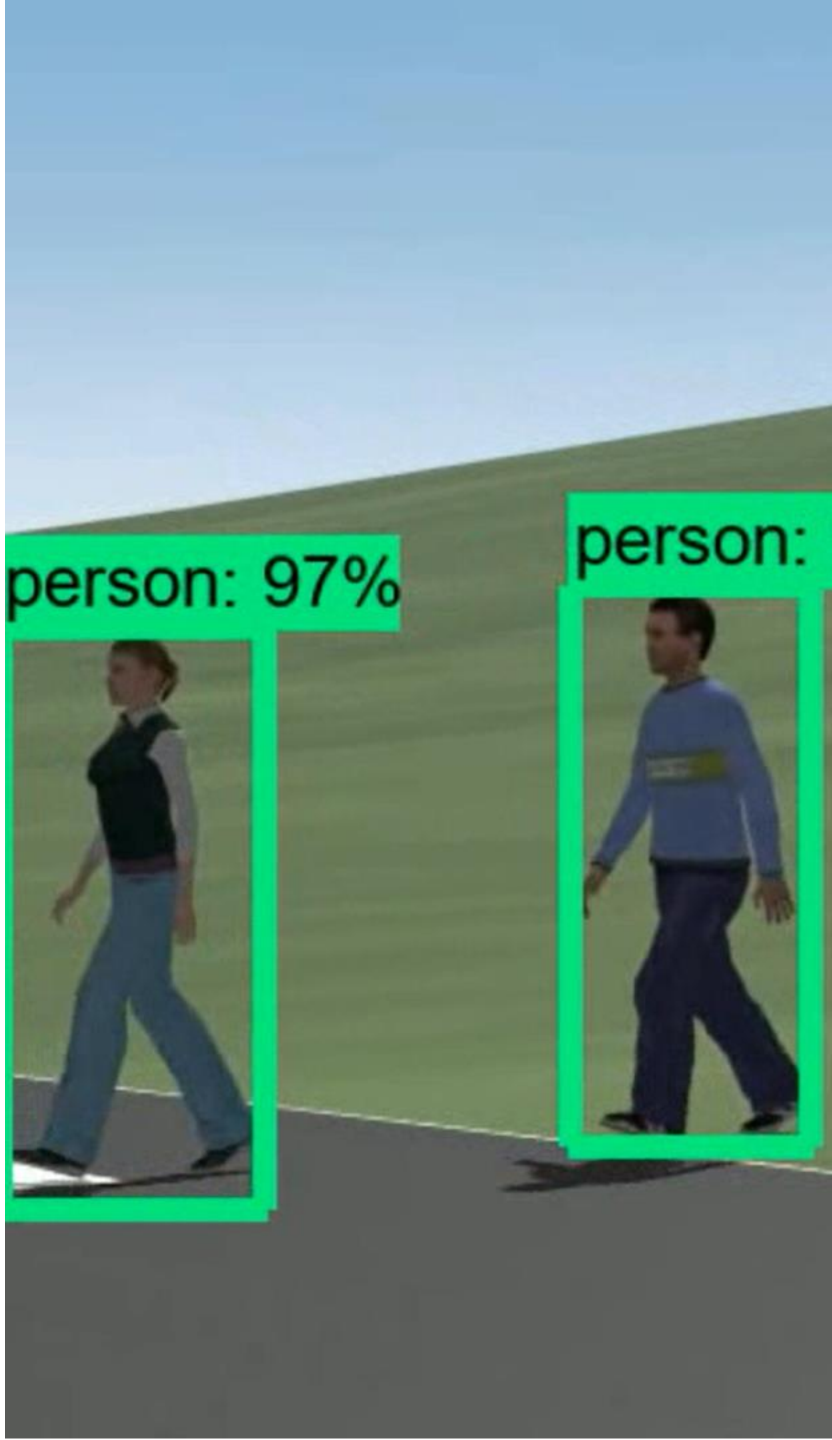
3. YOLO
   - YOLO is extremely fast and reasonably accurate.
   - Its accuracy is similar to Mask R-CNN and also has better detection time than single shot detector.

As we wanted our OEDR module to be fast with out loosing accuracy we chose YOLO.



Traffic Datasets

- We wanted to train our machine learning models with reasonable size of dataset but due to high GPU requirements we trained them with small sized dataset.
- We trained our models with real life traffic dataset because we did not have access to IPG CarMaker traffic dataset.



Traffic Datasets

➢ Urban Traffic Datasets Considered

- Lyft:

  Large dataset size (>100 GB Training Data),

  Incompatible image annotation (3D Annotation)

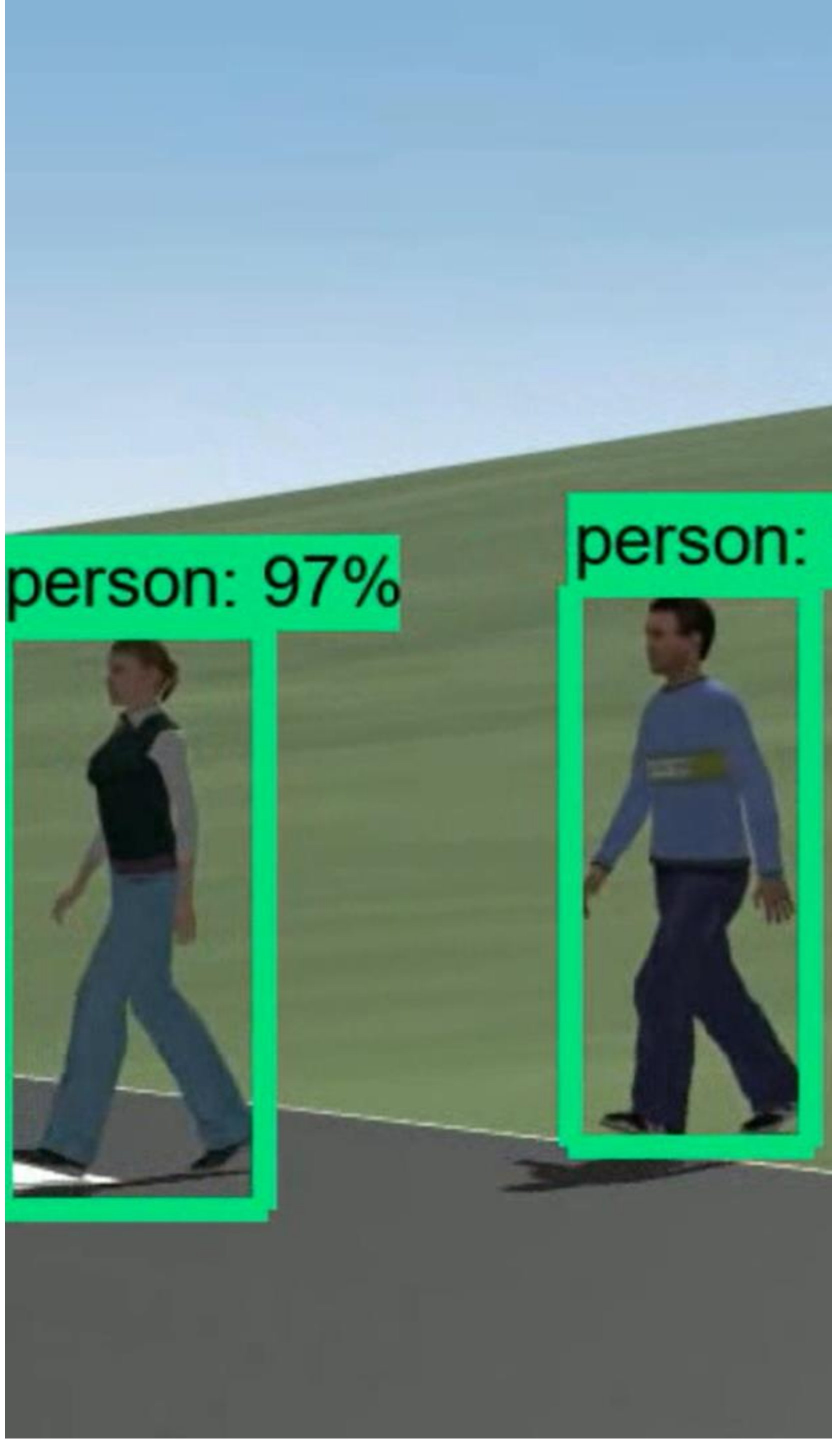
- Waymo:

  Large dataset size (~1 TB)

- Rovit:

  Small dataset size (~23 GB), Compatible image

  annotation (2D Annotation)



YOLOv3

- ➢YOLO stands for You Only Look Once
- ➢It is fully Convolutional layer with skip connection and up-sampling
- ➢We preferred YOLOv3 over YOLOv2 or YOLO

  Because small objects are detected with higher accuracy compared to it’s predecessors for it’s three level detection and classification.

YOLOv3

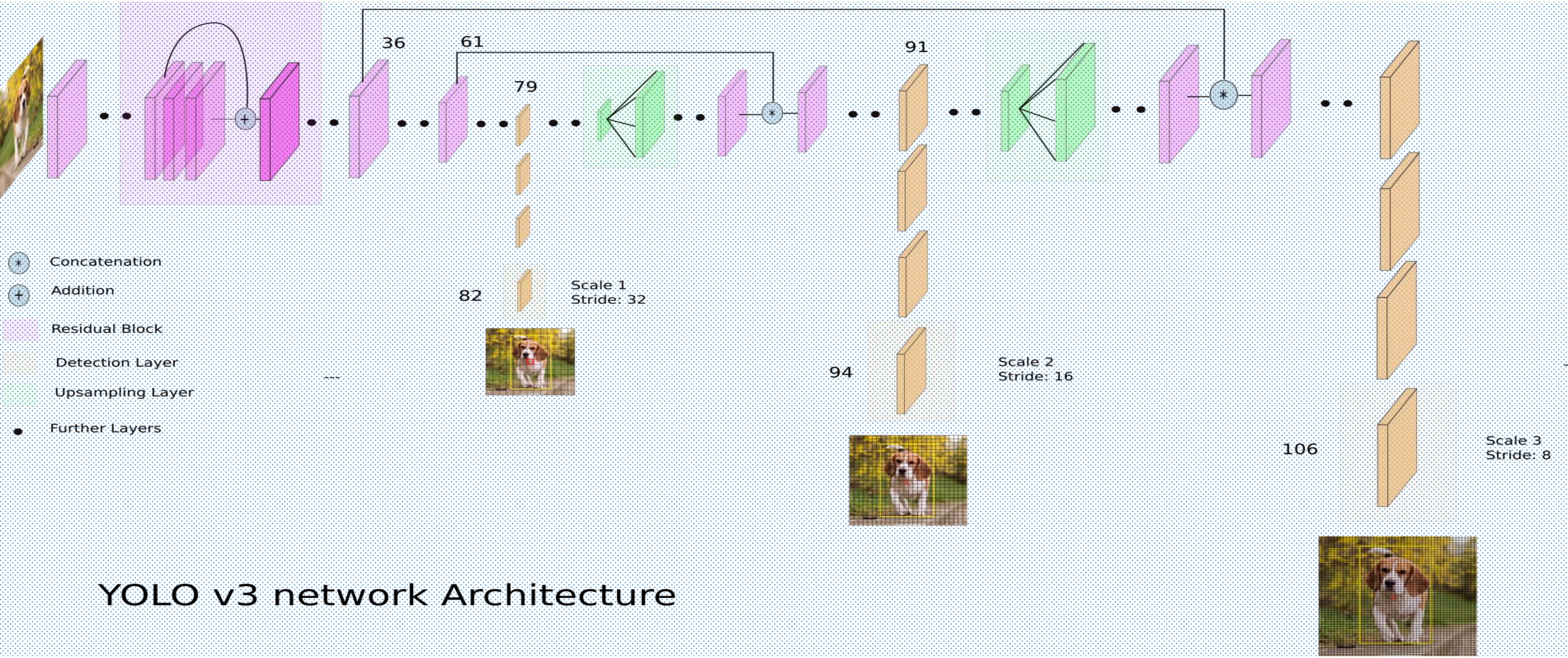
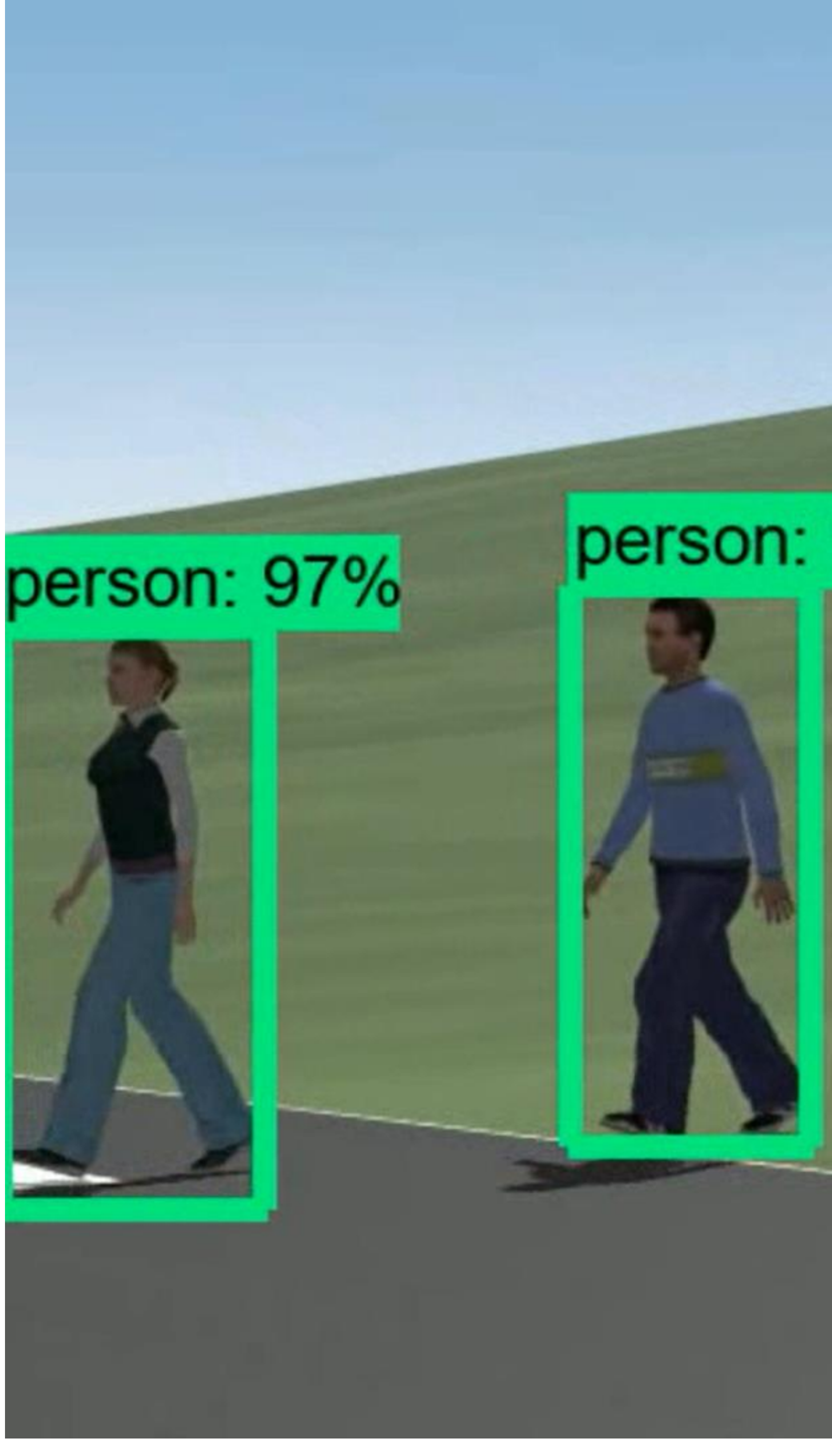


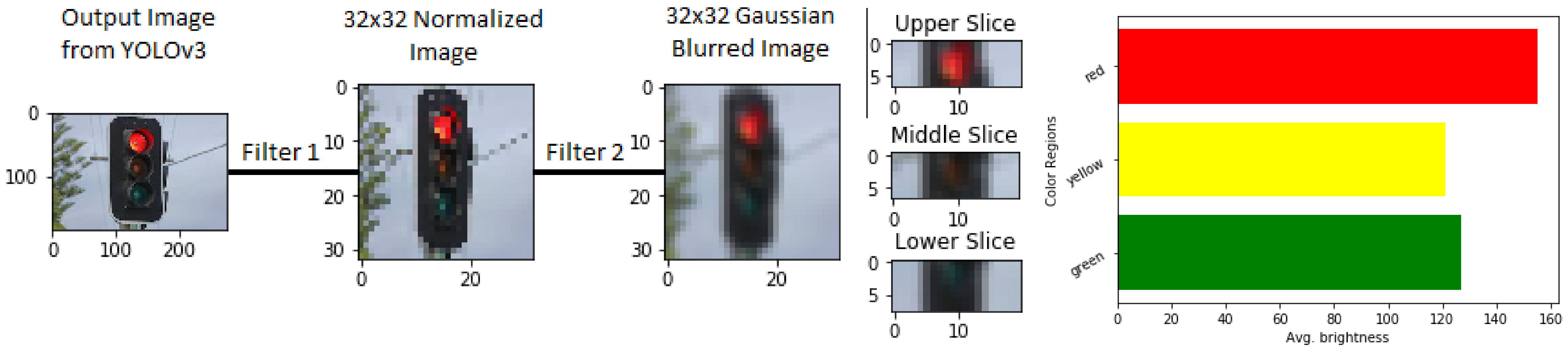
Image Credit: Ayoosh Kathuria

Traffic Light Classification

- ➢We considered machine learning and computer vision models.
- ➢As computer vision models are as accurate as machine learning models in this use case with higher performance we selected computer vision model.



Traffic Light Classification



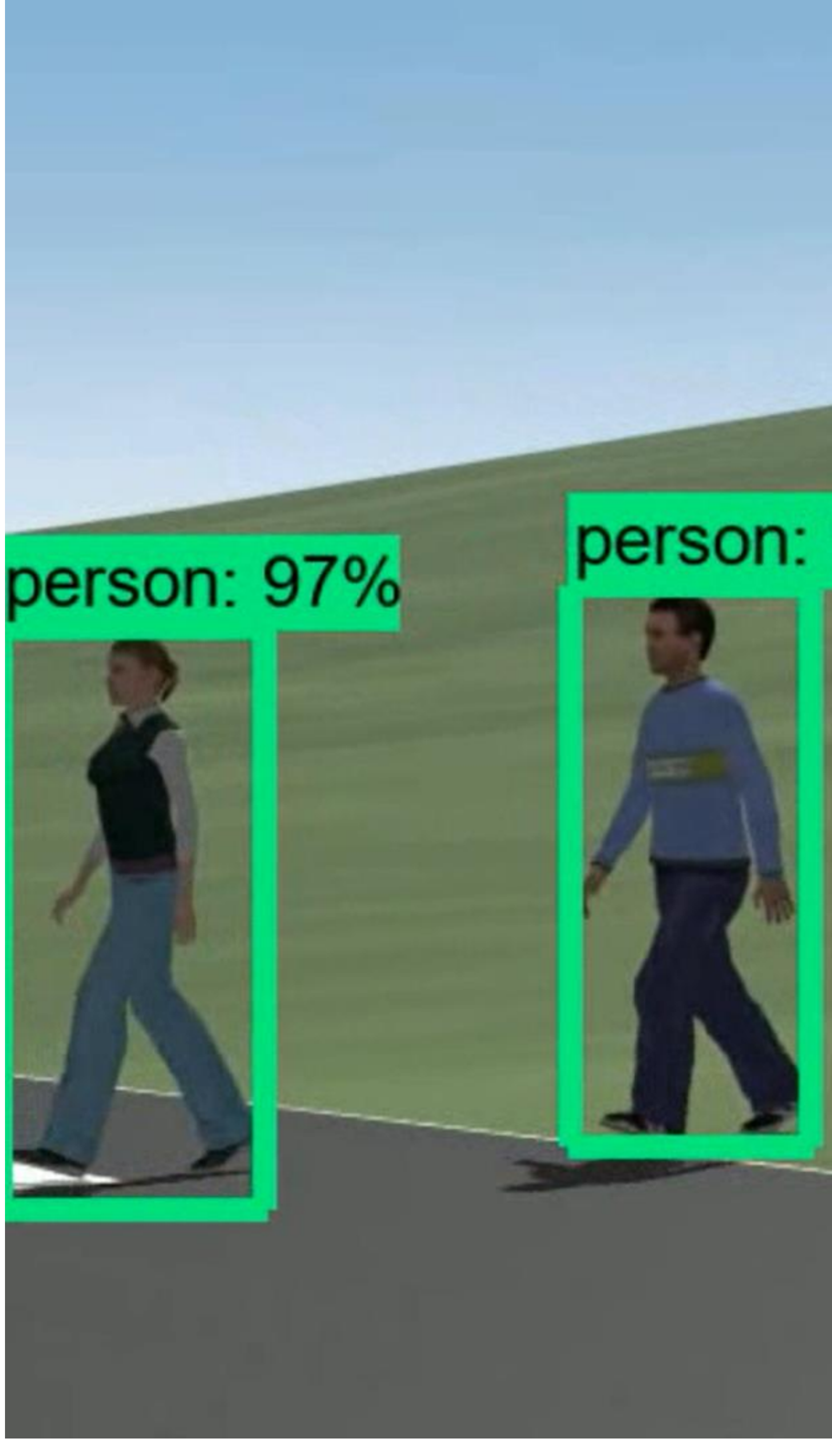
Filter 1 - Size Normalization Filter
Filter 2 - Gaussian Filter

Traffic Light Classification

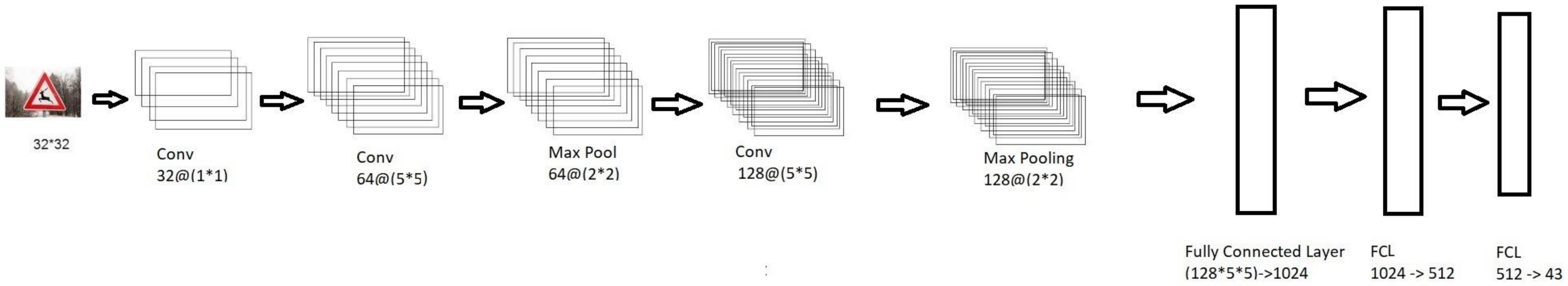
- Average Time for computation is 0.0033 s
- Testing Images for red Traffic light illuminated 723
- Testing Images for green traffic light illuminated 429
- Testing Images for yellow traffic light illuminated 44
- 100% accuracy for Red traffic light illuminated.
- 100% accuracy for Yellow traffic light illuminated.
- 99% accuracy for Green traffic light illuminated.

Traffic Sign Classification

- ➢We first used LeNet-5 architecture.
- ➢We achieved ~91% accuracy.
- ➢We have customized LeNet-5 architecture to improve performance for our use case.



Traffic Sign Classification

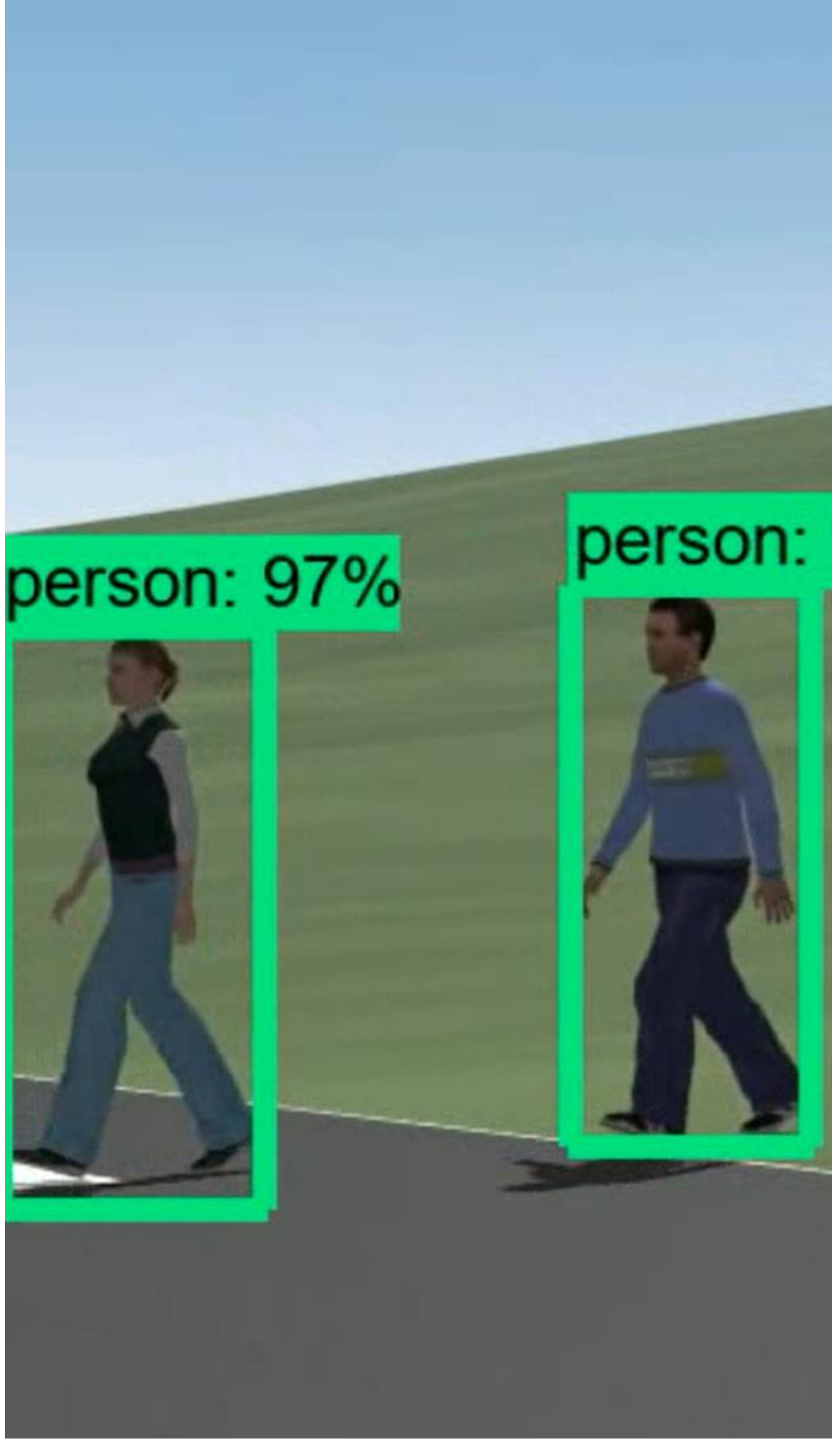


Traffic Sign Classification

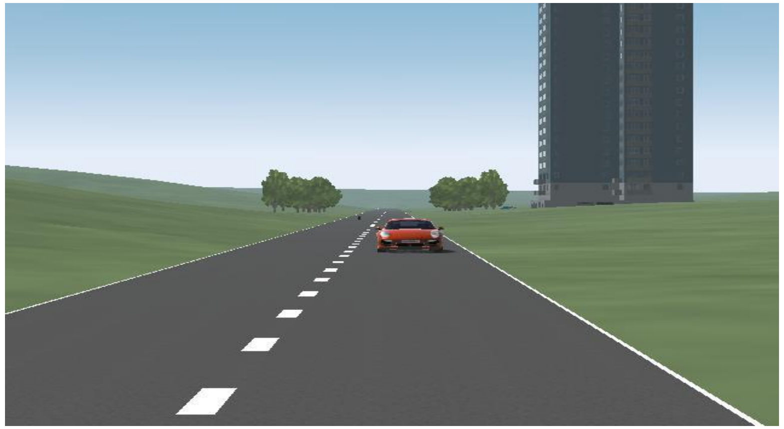
- Total Number of Classes: 43 [German Traffic Sign Dataset]
- Training Images 34799
- Testing Images 10000
- Accuracy of the Model is 97%
- Average Time for computation is 0.0033 s

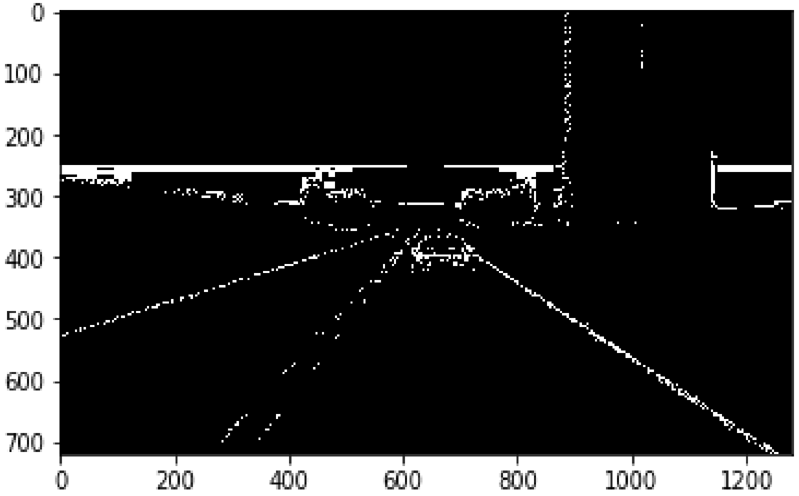
Lane Detection

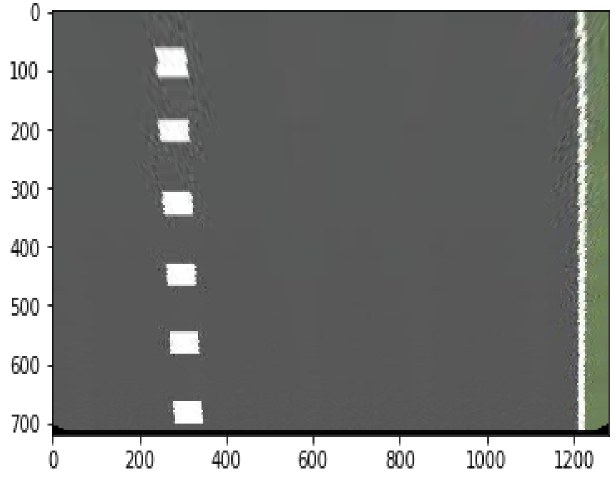
- Lane detection module consists outputs detected lane co-ordinates as well as radius of curvature of the road.
- We have used computer vision techniques in this module.

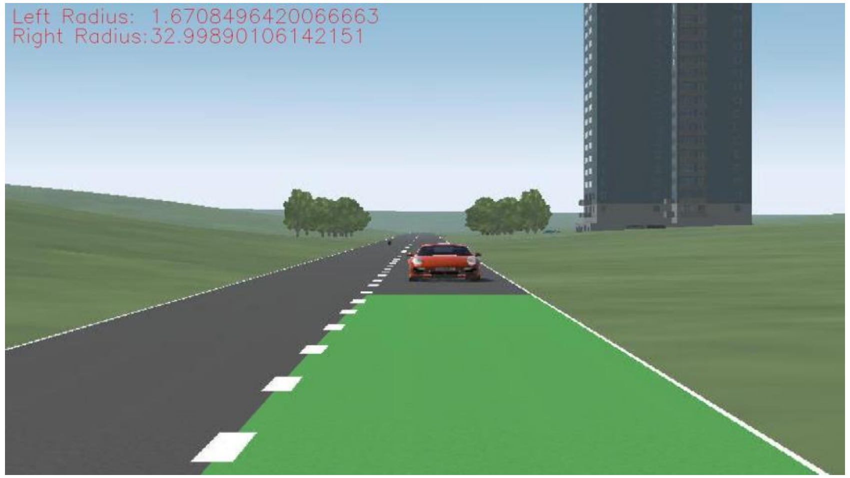


Lane Detection



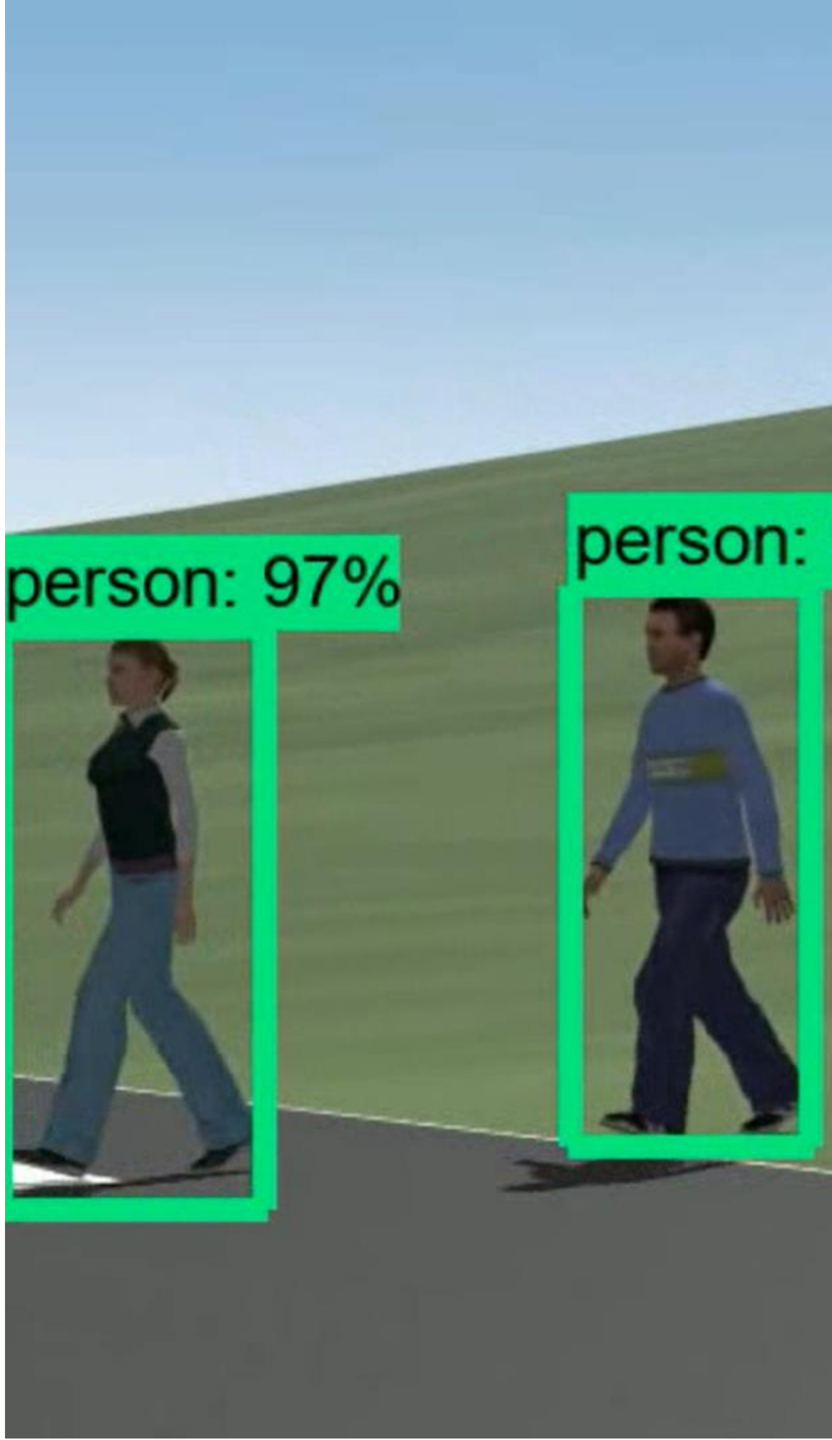






Traffic Depth Estimation

- ➢We have used stereo depth estimation techniques to measure depth.
- ➢For this purpose we have used two camera sensors in our ego vehicle (in IPG-CM).



Depth Estimation

$$\frac{Z}{f} = \frac{X}{x_L} \rightarrow ZxL = fX$$

$$\frac{Z}{f} = \frac{X - b}{x_r} \rightarrow ZxR = fX - fb$$

$$Zx_R = Zx_L - fb$$

$$Z = \frac{fb}{d}$$

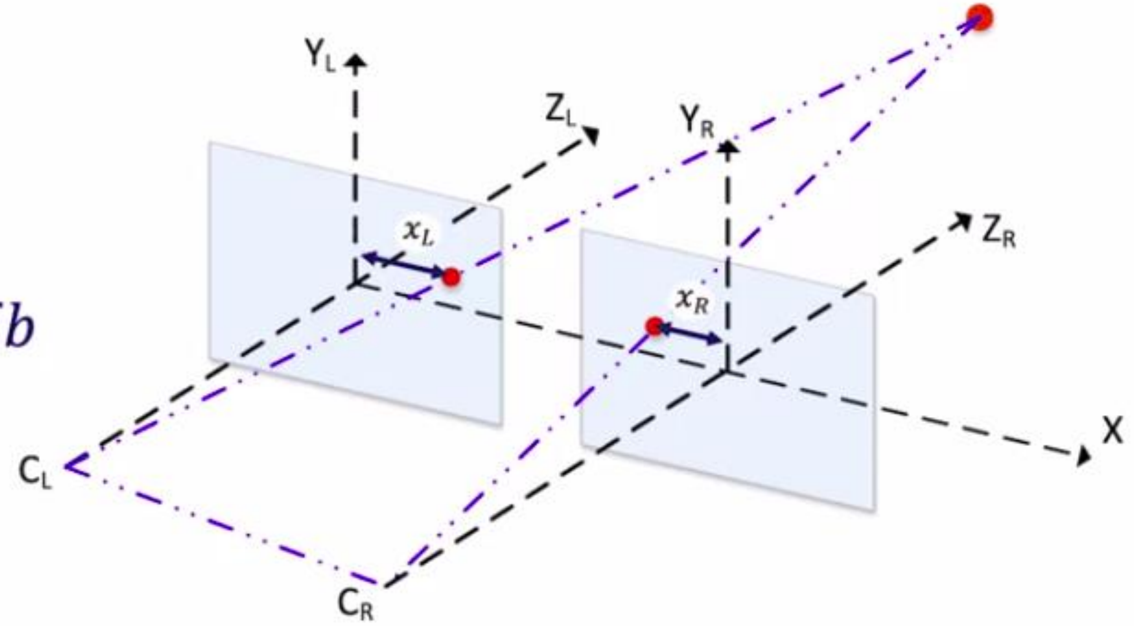
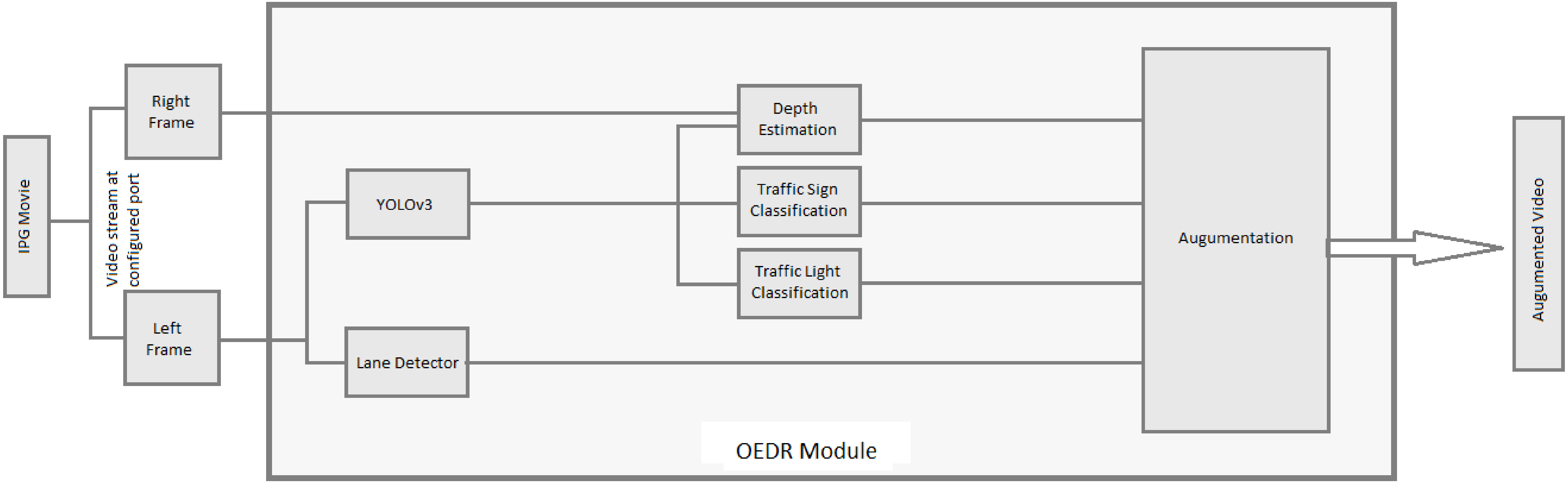


Image Credit: Coursera

Depth Estimation

- Semi Global Block Matching
- Accuracy of depth estimation is directly proportional to the area of the image overlap.
- We can improve the accuracy by increasing the field of view of camera sensors.
- Block size: 11
- Max disparity value: 400
- Minimum distance for estimation from the camera: 3m

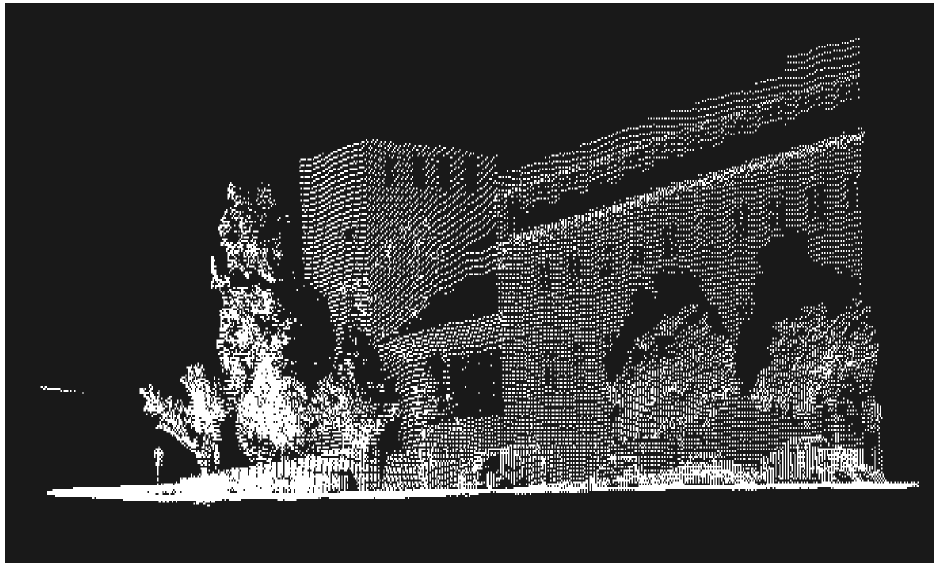
Camera Integration

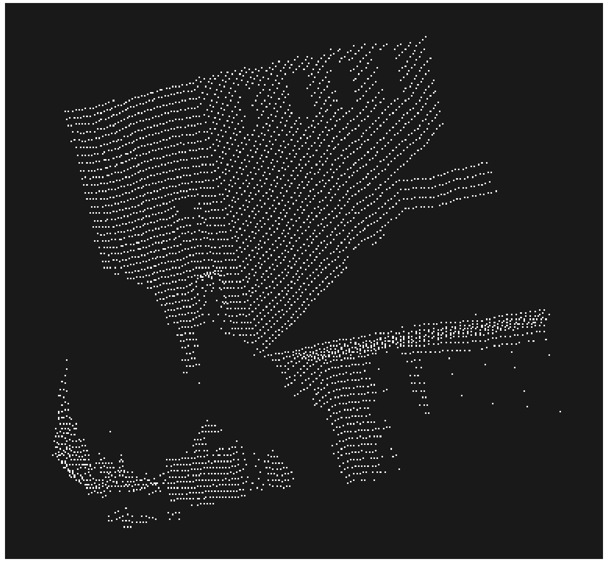


LIDAR

- ➢LIDAR stands for Light Detection and Ranging
- ➢It is used for improved detection
- ➢We have used Euclidean clustering algorithm

LIDAR






Left Radius: 21.545631547631 95
Right Radius:28.587614604007133
car 0.87
person 0.77
Depth:5.0
RED
trafficsignal

Future Scope

- Automated Test Scenario Generation
- Motion Prediction
- Safety Of The Intended Functionality (SOTIF)
- More sensors and Sensor fusion

THANK YOU