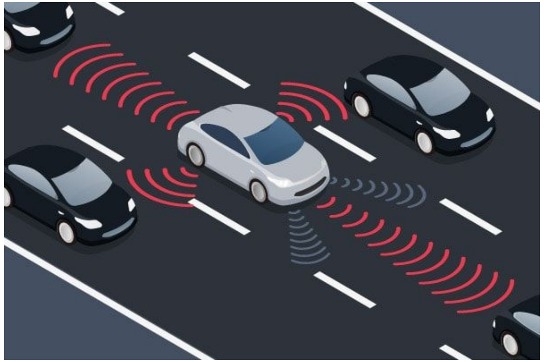
Deep Learning-Based Autonomous Navigation

Mahi, Jordan, James, Bhavik, Ryan

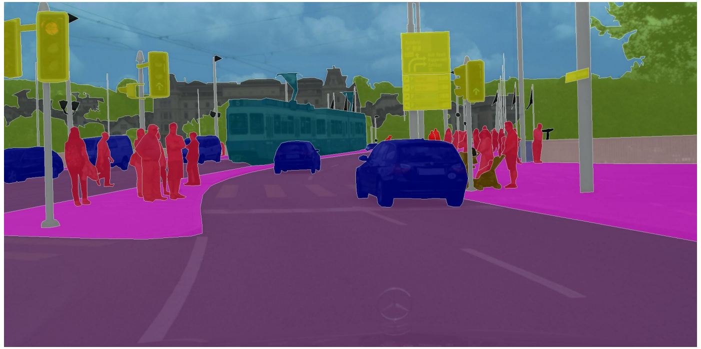
Wednesday May 15th, 2019

RSS 2019 Team 11

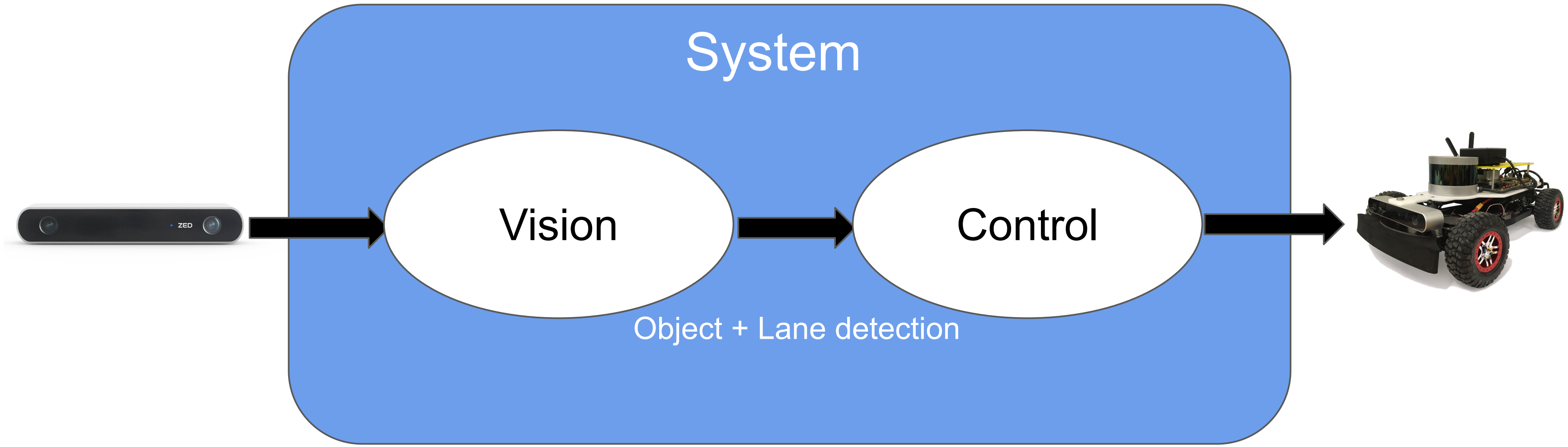
Motivation: Intelligent Autonomous Navigation



- Robust autonomous driving has massive societal implications:
  - Safer, more efficient transportation
  - Greater mobility
  - Increased productivity
- Autonomous driving requires robust ability to:
  - Understand the environment
  - Respond with well-developed controls



Our vision module informs our control module

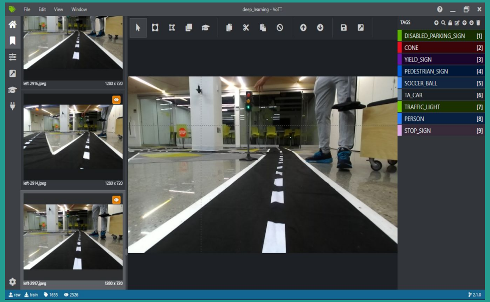


We trained object detection CNN on custom datasets

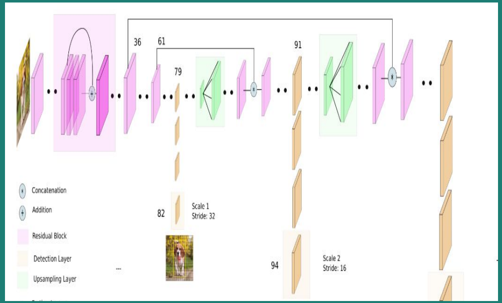
- **YOLOv3 Architecture**
  - Deep Convolutional Neural Network
  - Full Model: 252 layers
  - Tiny Model: 44 layers
- **Training**:
  - 1600 annotated custom images from Zed Camera
  - Keras + Tensorflow environment on Amazon EC2 Instance
  - 2-stage: first stage trains last 3 layers, second stage trains all layers
- **8 Custom Classes**
  - 4 signs: disabled parking, pedestrian, yield, and stop signs
  - car, person, traffic light, soccer ball (obstacle)

The YOLO Pipeline

[1] Labeling bbox via VoTT



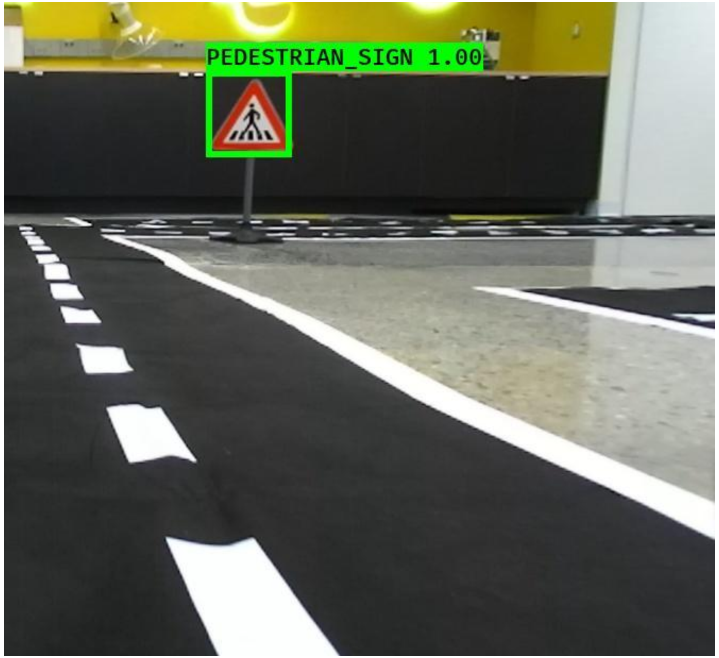
[2] Training YOLOv3



[3] Real Time Object Classification

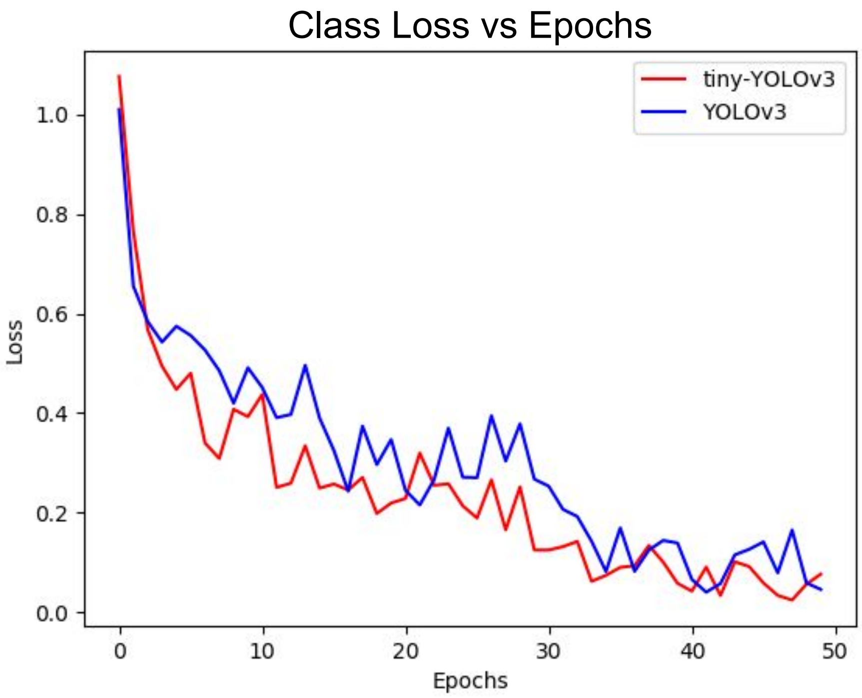
Real-time Object Detection

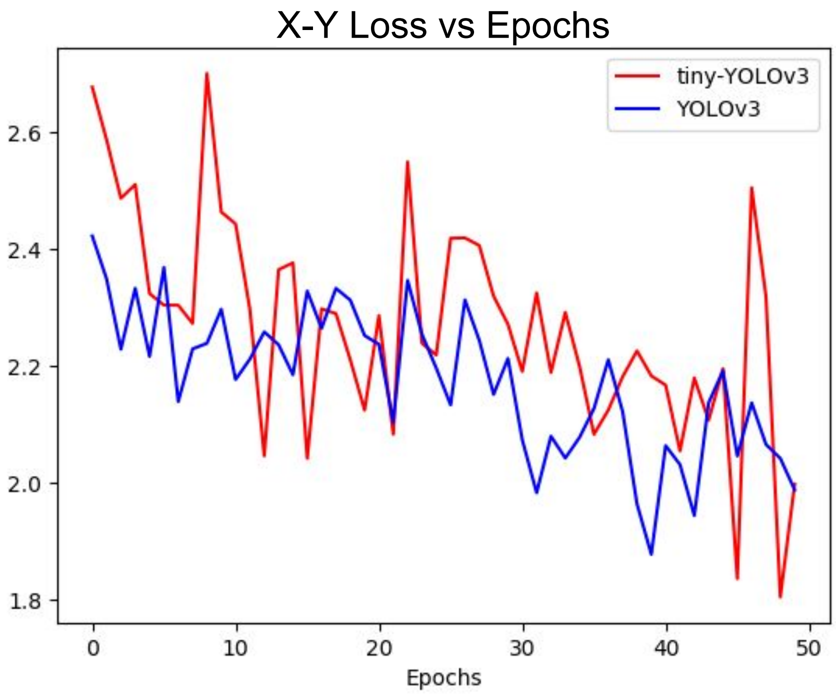
Detection of Disabled Parking Sign



Detection of Pedestrian Sign

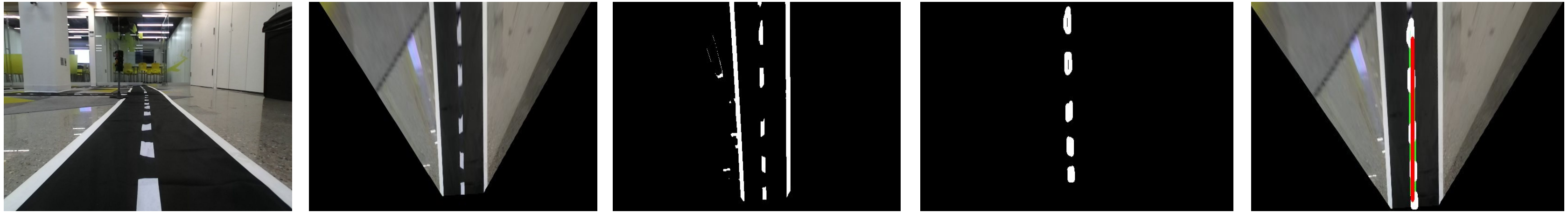
Tiny vs Full YOLOv3 Validation Error





Vision: Lane Segmentation

Original Image → Perspective Warping → White Segmentation → Contours Filtering → Hough Transform, Clustering

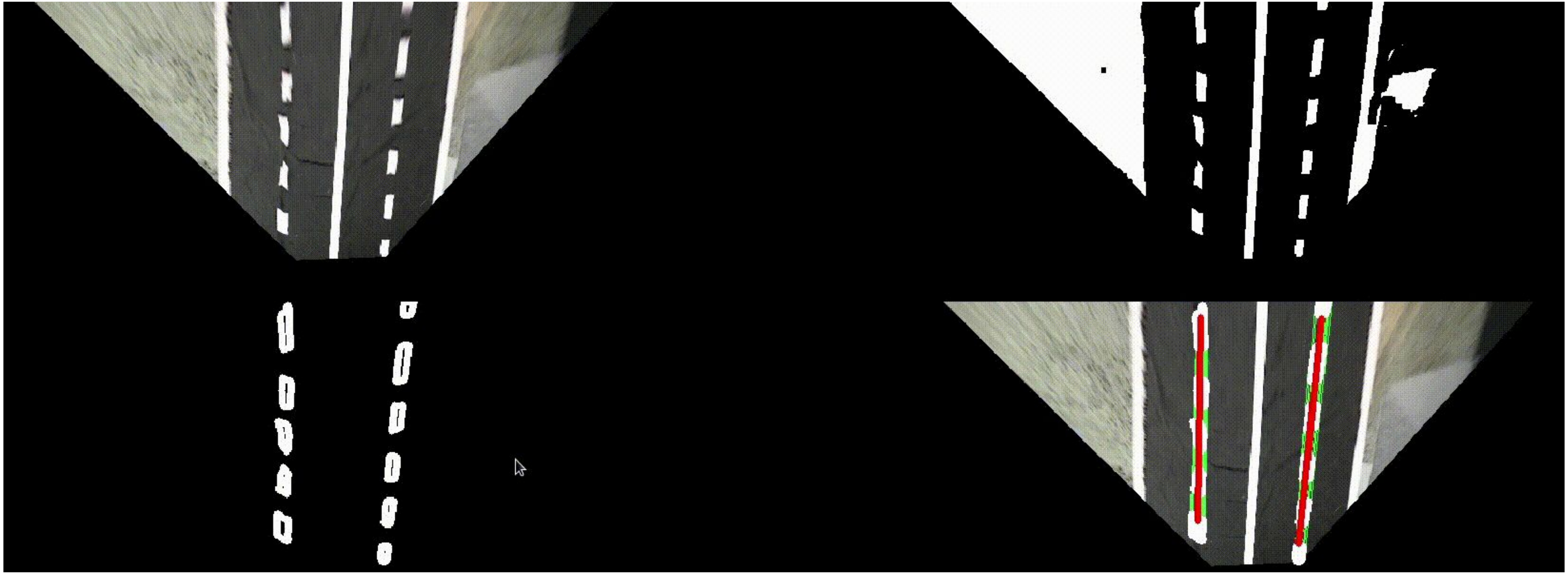


V

Or

sform,
ng

Lane Segmentation in Real Time



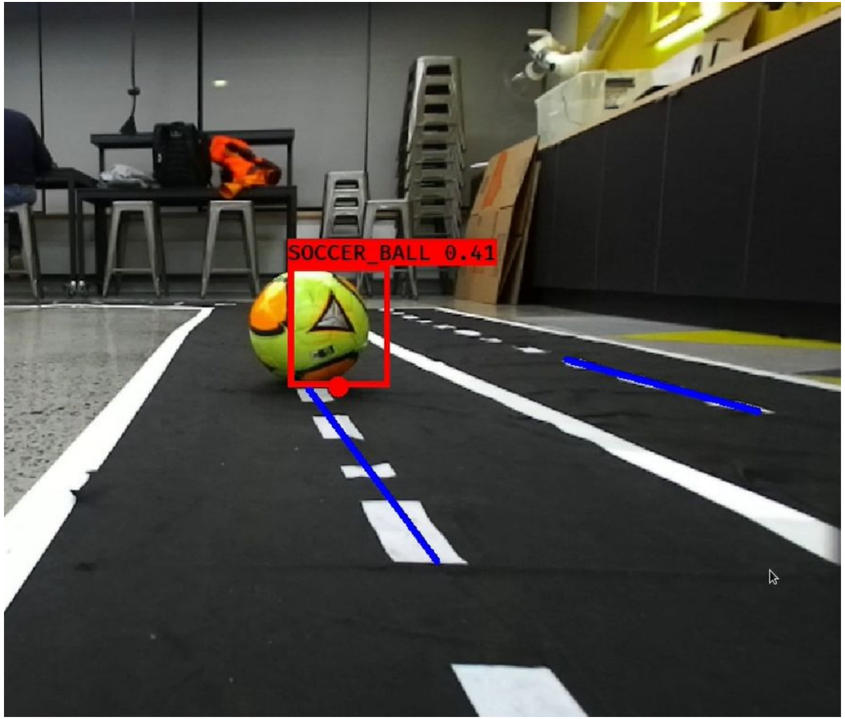
Controller: High Level

Object Detection + Lane Detection → Homography Transforms → $\boxed{\text{Controller Logic}}$ → Drive Commands (steering + speed)

Three states:

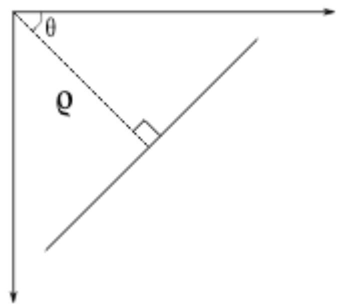
1) **Autonomous Driving**: lane changing/following, turning, stopping, traffic rules, etc.
2) **TA car following**: PD control based on relative distance/location from TA car
3) **Parking**: valid parking space detection + PD control

Steering controller finds the *best* unobstructed lane



1) Check for obstacles in current lane
2) *Best* lane: smallest difference to lane of previous timestep (in ρ, θ space)

$$\rho = x\cos\theta + y\sin\theta$$

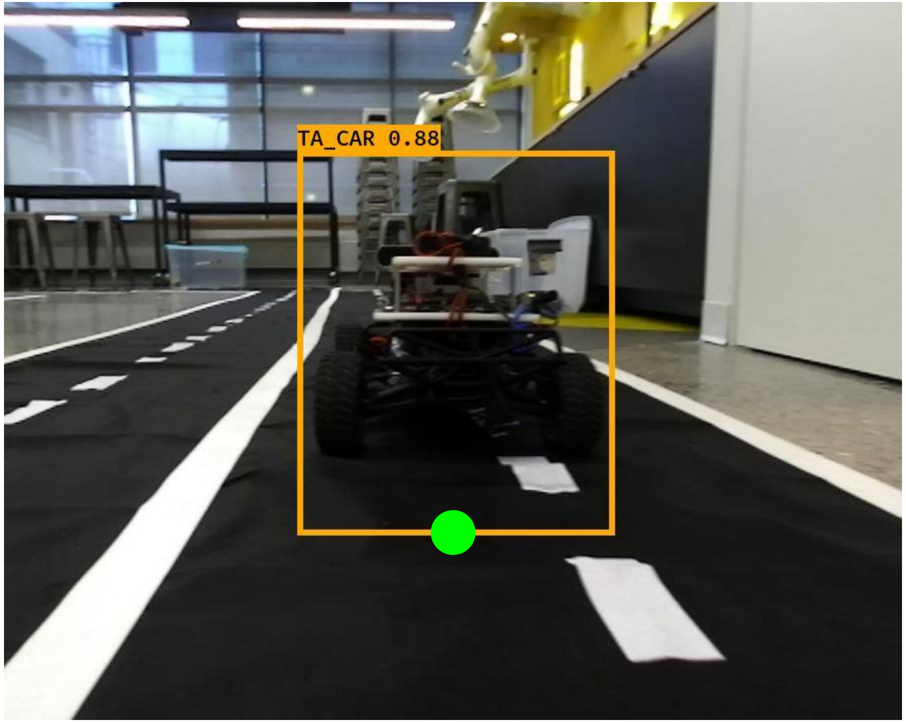


3) Angle calculation by Pure Pursuit

Limitations we faced, and our solutions

	Problems	Our solutions
1	Once at an intersection, robot can no longer see the intersection due to the camera’s limited field of view	● Plan out turns in advance ● Hardcode drive action ● (Ideally: more cameras)
2	Object detection does not recognize objects in every frame	● Rely on detected objects in the last frame(s)
3	YOLO often fails to load and crashes other nodes	● Wait… many hours ● Stressful hours ● (Ideally: explore options)

TA car following uses PD controller

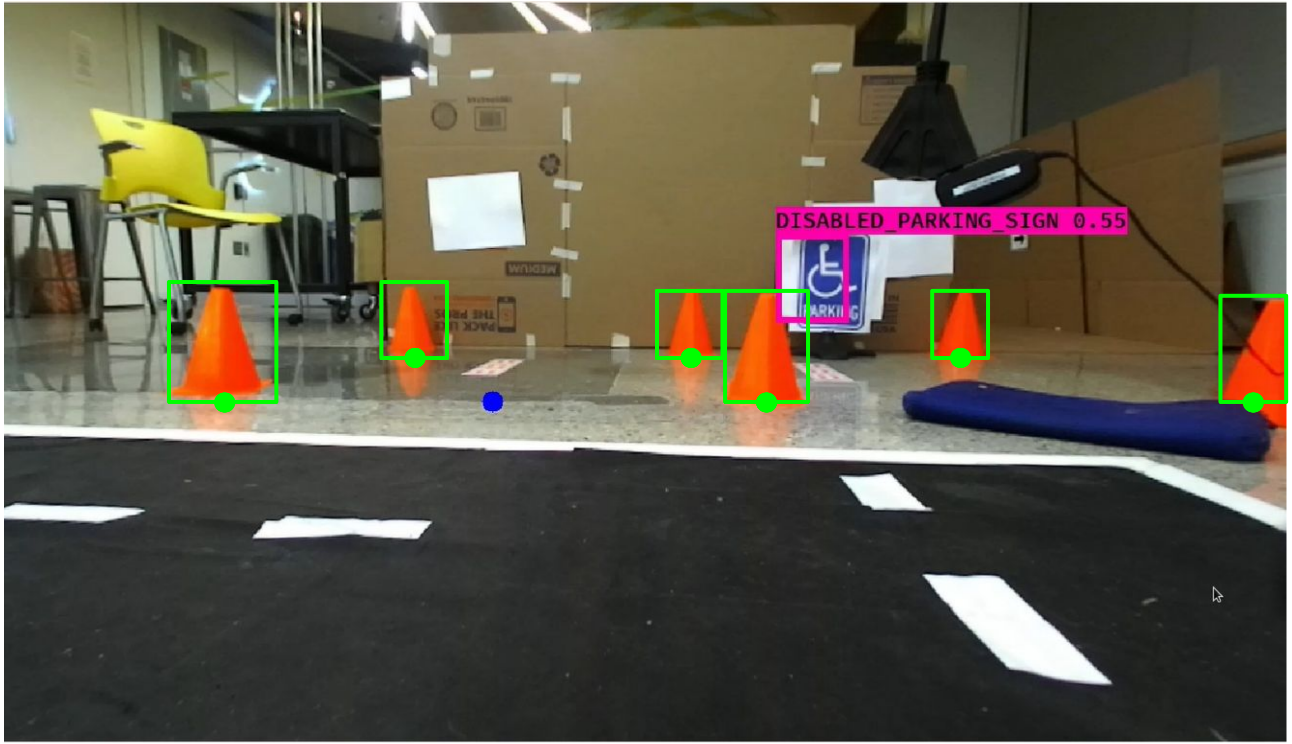


PD control on midpoint of bottom of bounding box

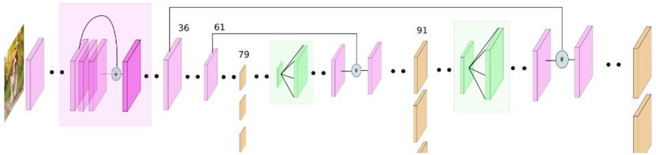
Desired distance: 1 meter

Parking controller uses color segmentation and YOLO

Color segment + bound orange cones → YOLO on disabled parking sign → Stage 1: Find spot with front 3 cones → Stage 2: Move to midpoint of back 2 cones → Stage 3: Stop when < 2 cones

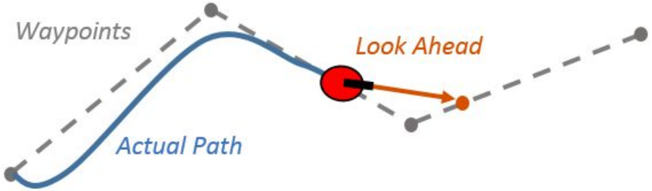


What We Learned

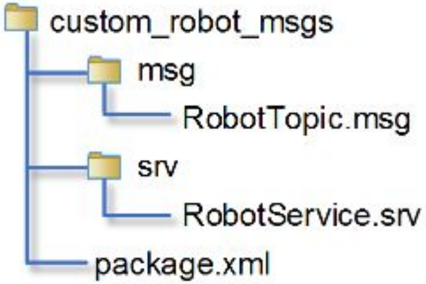


- **Object detection methods:**
  - YOLOv3 Deep Convolutional Neural Networks
  - Color segmentation

- **Lane detection methods:**
  - Hough Transform
  - Edge Detection
  - Clustering



- **Applications of pure pursuit control:**
  - Parking
  - Turning
  - Lane following
- **Developing our own ROS package**



What We Can Improve on

- **Object detection**
  - More data (e.g. lighting and placement variations) to decrease overfitting
  - Other network architecture for higher inference speed
- **Lane detection**
  - More robust algorithms, especially around the corners
- **Controller**
  - Higher driving speed for greater practicality

MIT RSS Team 11

Jordan Docter	MIT ‘21
Ryan Sander	MIT ‘20
James Pruegsanusak	MIT ‘19
Mahi Elango	MIT ‘20
Bhavik Nagda	MIT ‘21

With thanks to Marwa Abdulhai, Muyan Lin, Prof. Carlone, Prof. Karaman, and the rest of the RSS Staff!

MIT EECS
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

MIT | BEAVER WORKS
Lincoln Laboratory | School of Engineering