Analysis of Previously Published Works to Develop an Autonomous Self-Driving Formula Electric Vehicle

Harry DeCecco, Computer Science B.S.

Mentor: Aviral Shrivastava, Ph.D.

School of Computing and Augmented Intelligence

Research Question

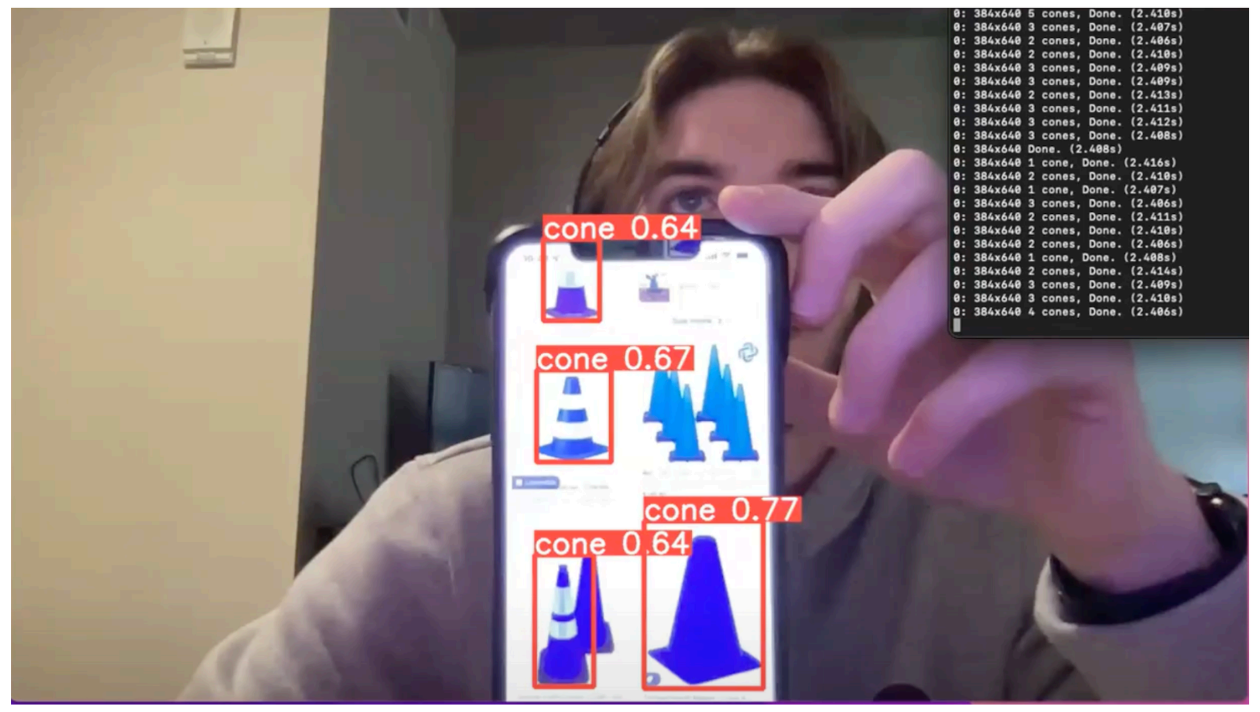
Using a collection of algorithms that will take in and interpret data from a camera, can a Formula Electric Vehicle be able to autonomously drive multiple laps around a competition track at the same speed as a human driver?

Background and Approach

To answer this question the approach that was taken is referred to as Simultaneous localization and mapping(SLAM). The SLAM approach is more of a concept than a single algorithm. The goal of the software is to build a map of an unknown environment, while also navigating the environment with a vehicle or something similar. To do this, SLAM must collect forms of data, such as camera feeds.

YOLOv5

YOLOv5 is a vision AI by ultralytics. It is an object detection algorithm that divides images into a grid system where each cell in the grid is responsible for detecting objects within itself. For our research we used this as the base software to detect cones in the frame. In order to do this we adjusted the weights for image recognition by giving it a custom dataset to learn from.



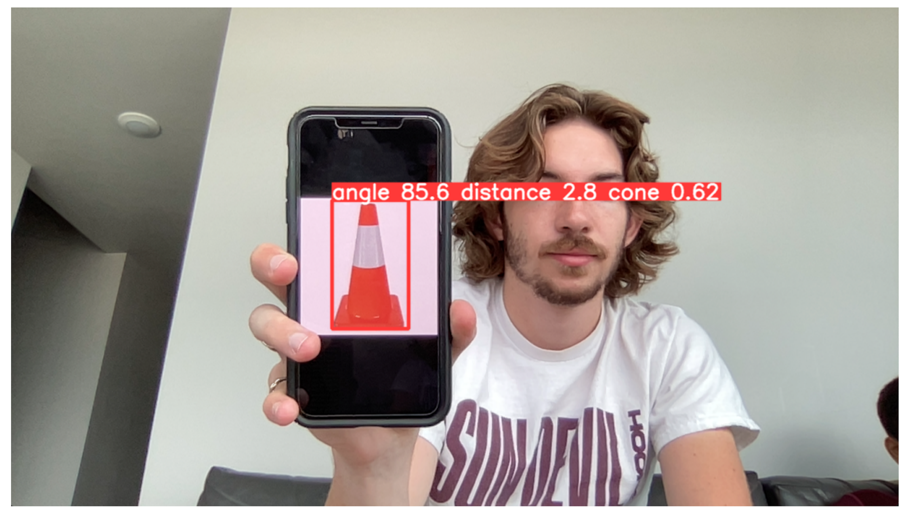
YOLOv5 detecting cones and displaying probability of cones

Challanges Faced

Image Latency is still a massive issue as with no dedicated GPU it takes on average 2.7 seconds to process one frame of data. Our solution to this is working with AWS services to use AWS E2 which is a server we could process these images on.

Extracting Data from YOLOv5

There are methods of finding the distance of an object from only camera information. The requirments being you know the real height of the object, the object height in pixels, the height of the camera, and the focal length. From there you can take the hight of pixels detected in the bounding box to determine distance. Similarly you can use this data to determine the field of view of the camera and can extrapolate the angle.



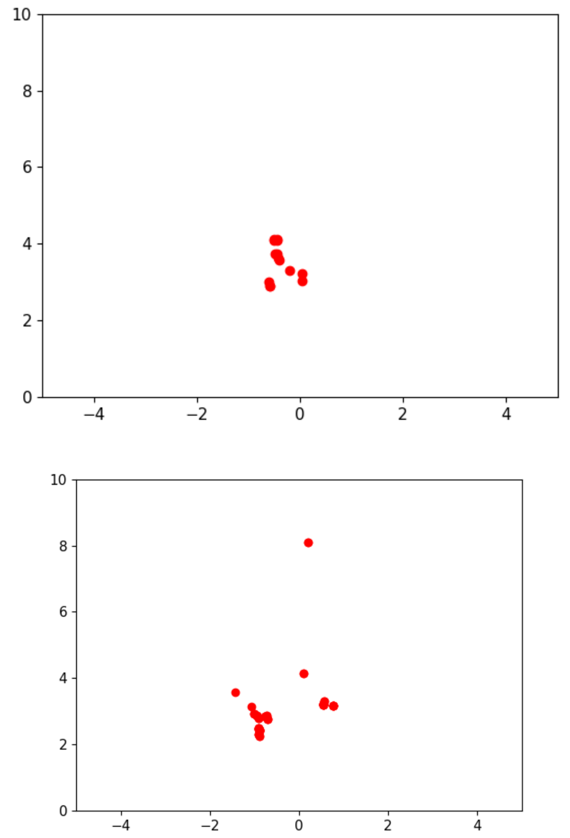
YOLOv5 displaying distance, angle, and probability of cones

YOLOv5 displaying distance, angle, and probability of cones

$$\text{Distance to object}(mm) = \frac{f(mm) \times \text{real height}(mm) \times \text{image height}(pixels)}{\text{object height}(pixels) \times \text{sensor height}(mm)}$$

Plotting the Cones

Mapping these cones from this data becomes a relativly easy task. Taking the angle and distance we can use trigonomitry to get the x and y cordinate of each cone realtive to the origin, the origin in this case being the location of the camera. We used the library Matplotlib to visualize and plot these points in real time.



Matplotlib displaying scatterplot of cones

Finding Best Fit Path

This is the current step in the research project, one that has not been fully completed yet. The approach for this is to take the closest left most and right most cone, plot the point equadistant from them, and traverse the track following a line connecting each cone.

Conclusion

While there is still massive steps needed to be taken in order to answer the research question presented, the achievements of this research project show that through a collection of different algorithms self driving capabilites can theoreticaly be plausable with only one camera as input.

Demonstrating x and y cordinate calculations being displayed in YOLOv5

Acknowledgments

Christian Polo: for his recomendation to the FURI program and general guidance

Aviral Shrivastava: for mentoring this project and taking it on at such late notice

Ethan Carlson: for the original training set and weight of YOLOv5

Jared Torres: for his help on finding the distance and angle of detected objects