Self-Driving Robotic Car

Undergraduate Students: Jack Circus, Logan Herr, and Sam Presti
Faculty Mentor: Professor Martin Takac **Graduate Mentor:** Xi He
Lehigh University, Industrial and Systems Engineering

Goals of the Project

1. Integrate car with computer to enable full control of car movement and functions
2. Detect position and orientation of car, and objects, using video feed and deep learning
3. Automatically have car self-navigate to an objective point using reinforcement learning

Introduction

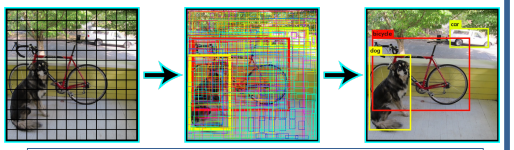
We began the semester full of passion for research and brought a few ideas to Professor Takac for approval. Our first thought was to be able to control the car using neural commands. We were able to use the Emotive EPOC+, a Scientific Contextual EEG that the Lehigh Industrial Engineering department had purchased for another research project. After initial success with the built-in applications, we hit many roadblocks when trying to access the raw EEG data transmitted through the headset. After numerous attempts and resorting to posts on online forums, we saw many other teams encountering the same problems and decided to pivot from our initial idea. We created a new goal of creating a self driving car using deep learning combined with reinforcement learning. The plan was to use artificial intelligence and a live video stream from overhead for the car to be able to navigate successfully through a closed course with obstacles. The environment would look like a bird's eye view of a parking lot area and our goal would be to navigate the car successfully to a target point while avoiding any collisions.

Controlling the Car

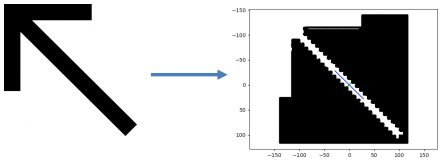
- Built and assembled car
- Setup and gained control of car through Arduino
  - Created firmware in Arduino to upload to the car
- Work with pybluez library to communicate with Python
  - Enabled bluetooth interface with the car
- Control car through Python command
  - After being synced with Arduino
  - Car can read commands and move throughout environment

Image Recognition

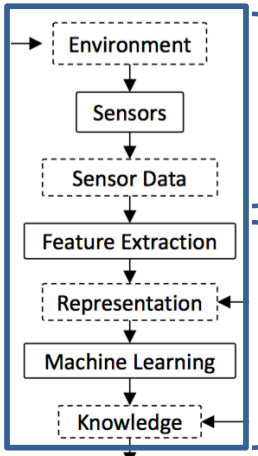
- Goal: Determine position and orientation of the car, as well as identify obstacles and their relative position to the car
- We implemented YOLO, a platform for image detection
  - YOLO uses a convolutional neural network to determine what is in an image. It can do this with live video and uses varying levels of confidence. Includes a pre-trained model that we used for testing

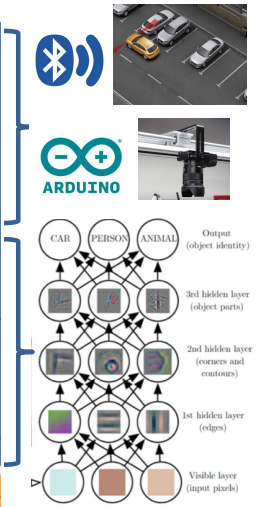


- Read live feed from IP webcam using openCV
- Used PyTorch YOLO wrapper to analyze images and recognize objects
- Used PCA to determine orientation of an object from above
- Next step: train YOLO on images of our robot
  - Use image manipulation software to generate large training set from only a few images



What We Accomplished





Next Steps