

## Self-Driving RC Car

Evan Fabry (efabry2), Harshit Agarwal (hagarwa3), Moderator(s): Charlie Martel, Rishabh Jain

- **Abstract**

- **Project Purpose**

- In this project we will create a self-driving RC car

- **Background/Motivation**

- We have researched Computer Vision algorithms before and have some experience with point cloud systems

- **Technical Specifications**

- **Platform:** Python on a raspberry pi, OpenCV or Tensorflow Convolutional Neural Network, RC car/controller, possible a game controller, possibly Arduino
  - **Programming Languages:** Python
  - **Stylistic Conventions:** function signature commenting
  - **SDK:** OpenCV or TensorFlow
  - **IDE:** Sublime text + terminal + PyCharm
  - **Tools/Interfaces:** camera, RC car, json settings, local SSH
  - **Target Audience:** Prospective employers, tbh

### Functional Specifications

- **Features**

- The RC car will be able to drive around the room and avoid obstacles
    - Timer permitting, there will be a video feed hosted at a local IP address showing object and distance detection over the feed from the car's camera
    - The car will make use of Convolutional Neural Networks to identify objects
    - Time permitting, the user will be able to input some direction via a game controller (semi-autonomous driving)

- **Scope of project**

- Limited to people with hella time to burn, a raspberry pi, maybe an arduino, a small camera, and enough technology

- **Timeline:**

- **Week 1**

- Figure out acquisition of RC car, Raspberry Pi (Evan owns one), small camera
    - Set up a camera feed to Raspberry Pi
    - Build or otherwise deploy a multithreaded TCP server on the Pi to accept video feed, other possible sensor feedback, and possible external commands
    - Note that the RC car is not needed at this stage

- **Week 2**

- Build the interfacing to control of RC car via either the Pi itself (if possible considering weight, etc.) or use an Arduino

- Write Arduino code to accept streaming controls over serial connection
- Interface between Python and Arduino on the Pi to control car from Python
- Establish interfacing over Flask server to control the car remotely
- Build testing scripts for controlling the car
- Set up multi-threaded website to show video stream and control the car
- Show video of testing on web site
- **Week 3**
  - Research other classifiers and begin to train detection of individual objects
  - Begin training distance detection
  - Train basic edge detection and obstacle detection
  - Program an exploration mode that drives around and avoids obstacles
- **Week 4**
  - Make car perform specific actions for specific objects classified (stopping for a red block or stop sign)
  - Figure out anything that isn't working by then
  - Build in controls to let a user partially guide the system via a game controller
  - Stream video to a separate local IP address time permitting
  - Embed video in a basic page with Python and Flask (time permitting)

### **Future Enhancements**

Build a video stream that shows objects detected and estimated distances, host that feed on a website, explore state of the art algorithms, train with more data, turn it into a cat toy.