Self Driving Remote Control Toy Car

By: Hunter Wright, Jonathan Rodriquez and Alex Eurice

Objectives & motivation

In this project we will create a self driving rc car.The motivation behind this project is to learn the basics of machine learning. In this project we will focus on the neural network subset of machine learning. We will be using tensorflow framework to develop the neural network. Tensorflow will do the heavy lifting for us since this is a new topic for all of us. It is possible to develop our own neural network for this project but with the timeframe and learning curve it will add unneeded complexity. A high level overview of the project is as follows. The rc car will send a video stream to a server (personal laptop). Then the server will interpret the video stream which will be a collection of still images. The server will then give a prediction in a form of a directional output. That output will be then read in by the car which will be translated to motion via the dc motors in the rc car. In order to accomplish this we will need to create training data that the neural network will interpret and learn when right, left and forward movements need to occur. The main objective of the project is for the rc car to follow a road while staying inside its lane. Once the rc car is able to navigate a road without going outside its late then stop sign detection will be the next objective. If time permits we will incorporate stop light detection where if the light is red the car will stop and if the light is green it will continue as normal. On front of the car we will have an ultrasonic radar that will detect object directly in front of the car this will add a level of safety to the car to avoid head on collisions.

Hardware

Cheap RC Car

Found a cheap RC Car from a big box store that has 2 dc motors. One motor that is for forward and backward motion and the 2nd motor for left and right steering. We decided to go with a larger car so that all the components will easily fit inside the body.

Raspberry pi 3 B+

The raspberry pi will be in charge of streaming the video to the computational computer. The raspberry pi is also in charge of receiving signals from the computational computer that translate into dc motor on/off states. The computational computer will run the model produced by tensorflow and send what commands the RC Car will execute. With the built in WiFi the raspberry pi has it will make it possible to send the video stream to the computational computer, in this case our laptops.

L293D

The L293D is a small chip that allows the control of the dc motors from the raspberry pi GPIO pins. This chip allows bidirectional control of the dc motors which allows forward/backward and right/left motion of the RC Car.

HC-SR04

This sensor is a ultrasonic distance sensor that will prevent head on collisions occurring. Since the car will only have 1 webcam it will be difficult to determine how far objects are from the car without other kinds of sensors attached.

Fisheye Raspberry pi Camera

The Fish eye camera will give a wide angle view of the road ahead of the rc car.

Wireless Router

A wireless router will allow communication between the pi and the computer doing the computation. The video stream will be streamed through the router. And the commands for the rc car will be sent to the arduino via the wireless network. It's possible that the laptop will be able to create its own private network, in that case a router will not be needed. testing needs to happen.

Battery Pack

We will have 2 generic battery packs that will give power to the raspberry pi and one for the dc motors. This will give the Rc car full range or motions without the need to worry about wires getting in the way.

APIs

Tensorflow

This is a machine learning API produced by google. This API will allow us to create a neural network that will interpret a video stream and translate it to motion for the rc car to perform. Training data needs to be created so that the neutral network can learn when the car needs to drive forward, turn left and right.

Open CV

This API will allow is to have access to the video stream that the fisheye webcam will capture. Open CV also allows you to create a black and white video stream which will allow the neural network to make faster decisions because of how the video stream stored. If the video stream is in color the video will be stores as 3 Matrices as opposed to 1 matrix if its in black in white.

Plan of work and timeline

- Acquire all hardware needed to complete project (Due March 10th)
- Wire components together (Due March 17)
- Be able to obtain video feed on personal laptop from raspberry pi (Due March 24)
- Develope a controller on personal computer to move rc car for training of the neural network (Due March 31)
- Construct road that Rc car will run on (Due April 4)
- Develop a way to sync video feed with keyboard pressing used to train the neural Network (Due April 7)
- Construct neural network model in tensorflow
- Recording training data
- Input training data into tensorflow model
- Test model on real track with different turns and courses
- Adjust model based and training data based on results
- Incorporate stop sign detection with distance estimation
- Incorporate HC-SR04 to detect when objects are directly in front of vehicle
- Incorporate red light/ green light detection (if time permits)
- Incorporate pedestrian detection (if time permits)

Expected results and evaluation methodology

At the end of this project we expect to have the beginnings of an autonomous vehicle. We expect for the rc car to drive a course that mimics the real world without any help from us. We expect the rc car not to only drive a course that it has seen before it should be able to drive a different course with different turns and intersections that it hasn't seen before. To test the model first we will test car with a straight track. Then we will test the model with a track that has a right turns and left turns. Then we will test the model against a course that has a stop signs throughout the track. Depending on our results we will adjust the model and retest until desired results. Lastly we will test the ultrasonic to ensure it detects object that will cause a collision and stop the car appropriately . If the car passes all the test drives we will conclude the project a success.

Deliverables

Jonathan:

- Obtain parts needed to complete project
- Wire components together
- Create testing script to ensure motors are working and wired accordingly
- Create test script to video stream from pi camera to laptop
- Help create training data
- Help create tensorflow model

Hunter:

- Create class that allows keyboard input to control rc car
- Help create training data
- Help create tensorflow model

Alex:

- Help create training data
- Help create tensorflow model
- Create class that links video and keyboard input for model input

References

https://github.com/RyanZotti/Self-Driving-Car/blob/master/README.md

https://zhengludwig.wordpress.com/projects/self-driving-rc-car/

https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html#haar-cascade-detection-in-opencv

https://www.tensorflow.org/guide/keras/