College of Engineering & Architecture
Department of Electrical Engineering and Computer Science
Howard University
Washington, DC 20059

AutoMoe

Final Report
Senior Design Project | EE 404

Spring 2019

Team Advisor: Dr. Danda Rawat
Team Members: Pawan Gaire - EE, Satchin Campbell - CpE,
Samantha Jo Cunningham - EE, Savannah McCoy - CpE

Summary

This project is focused on the leveraging of an existing autonomous car framework, donkeycar, to model a simplified scenario with the intention of teasing possible improvements to complex autonomous vehicular systems. Our team designed, assembled and implemented a basic autonomous car system on two RC cars capable of driving autonomously around a heavily controlled track. These cars are capable of real time(near immediate) intercommunication with the use of TCP as our transport method. The cars are outfitted with a 3D-printed model in order to hold the basic components required for the above as well as five ultrasonic sensors strategically placed around the car and provides the input for the automatic braking system triggered in response to obstacles.

Abstract

The evolution of automated technology aims to provide safety benefits that can one day handle complete task of driving without a human overlooking the vehicle. But in the same time there are Cybersecurity risks such as user's data breach. Implementation of a secure smart-system for autonomous vehicles will reduce the number of accidents caused by drunk driving, distracted or reckless driving, speeding, and blind-spots which allows for more productive commute times and overall safer transportation. This project aims to develop lightweight cyber security schemes, privacy aware communications, adaptive speed control, automatic braking, rerouting, information sharing using wireless access technologies and display vehicle's status information. The final product consists of 3 prototype cars that can move around autonomously with such features.

Problem Statement

Implementation of a secure smart-system for autonomous vehicles which will reduce the number of accidents caused by drunk driving, distracted or reckless driving, speeding, and blind-spots which allows for more productive commute times and overall safer transportation.

Humans get distracted while driving. Lots of accident is caused due to speeding because of frustrations or being late for appointments or frustrating traffics, drunk driving, recklessness, because of blind spots- driver not able to view incoming objects on blind spot while switching lane. Automated driving has the potential of preventing such accidents. It might be difficult to imagine that car operated by microcomputer will be safer than with human driver. Turns out, 94 percent of accidents are caused by human error, such as speeding, driving recklessly, impaired driving, driving under influence or with inattentiveness, according to a study by NHTSA (National Highway Traffic Safety Administration).

On the other hand, self driving cars won't have these errors because they are purely analytical as they rely on sensors, radars and cameras to navigate and make decision. The computers are able to react faster than human. What's more? There is no distraction like smart phones, no visual impairment factor like alcohol or other drugs, and no emotion involved in decision making process. Moreover, all person inside car will be able to utilize their travel time doing something productive and in the meantime travel will be more efficient because using automatic GPS navigation and making most efficient use of fuel.

In the meantime, driverless cars are bound to introduce a new sets of challenges for us to deal with. Aside from having impact on jobs and the working class people, it also brings our privacy to question. The service provider will be tracking one's location and driving habits, similar to how tech companies like google monitors our search history. The software might also have some loopholes. Hence the question arises: what happens if the software on autonomous car fails? It the technology safe? Is it hackable? Who is held accountable if things doesn't work as they are supposed to? As the number of vehicles connected together increases and as the vehicles gets smarter, the number of ways to disrupt them and get into them also increases. But these issues are not impossible to fix.

Year Long Goal

The goal for 2018-2019 academic year was to design and build two autonomous car prototypes capable of privacy aware inter-communication.

Design Requirements

The final implementation should have the following features:

- Adaptive speed control - 0.5 m/s adjustments ✔
- Real-time data sharing using wireless access technology ✔
- Automatic braking - Halt when foreign object is detected ✔
- Vehicle status info display - at least 2 important features ✔
- Rerouting around obstacles - < 300 ms response time ✘
- Lightweight cybersecurity features ✘
- Privacy aware communication - At least 2 levels of privacy protection ✘

The ultimate design of AutoMoe should adhere to **NHTSA** and includes following features:

- Lane Keeping assist
- Adaptive cruise control
- Traffic jam assist
- Secure communication between devices

Some of the constraints for the project can be listed as:

- Available human resources
  - 2 EEs, 2 CpEs and 3 MEs
- Limited finances
- Driver vehicle interface:
  - Allow override for manual control in case of emergency
- Power:
  - Should use rechargeable batteries
- Compliance
  - Must not infringe existing patents

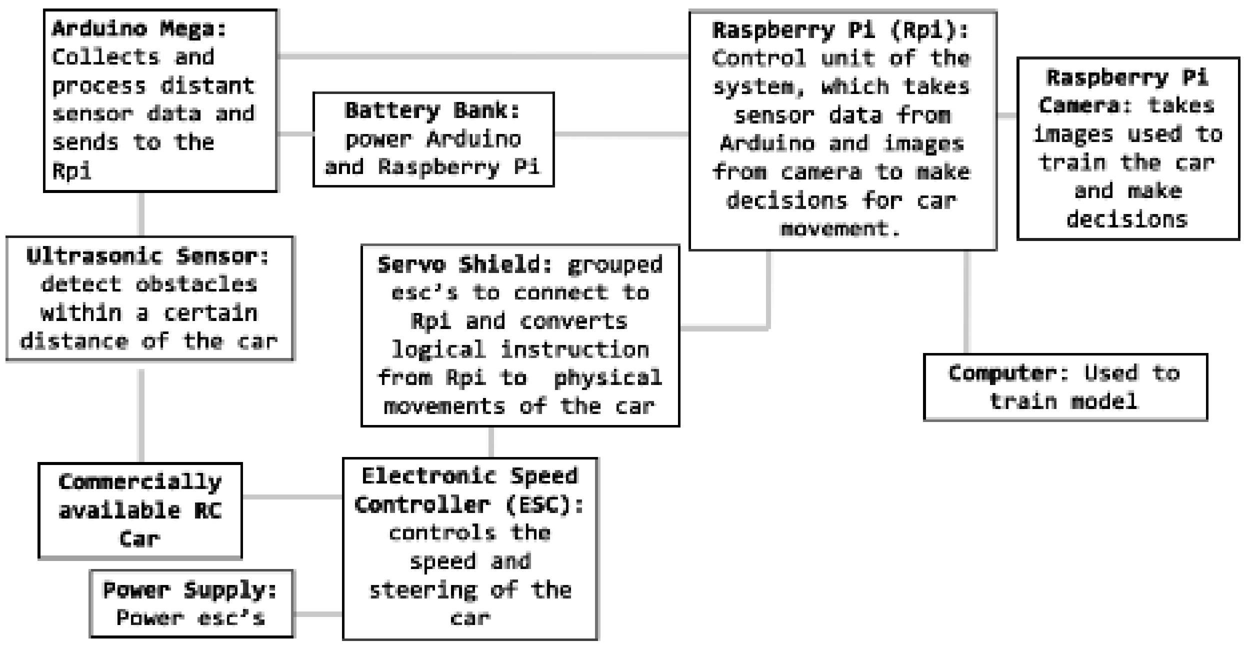
Feasibility & Resources Available:

The design requirements that would meet the needs addressed in the problem statement were addressed as an autonomous vehicle that requires spatial awareness in real time as well as an efficient enough response to avoid obstacles. The AutoMoe vehicle must have the following capabilities in order to have a design that is a solution to the problem, at least 50 cm field of vision ahead of itself, a 15 cm field of vision on both the left and right side, Acceleration control, Motor/Direction control, self sustaining power system. The design requirements is based on our desire for the autonomous vehicle to require no external support, and function without human intervention while also being a cost efficient implementation compared to other systems. A self powering system ensures that a decrease in voltage over time will not have an impact on the 3 system functionality.

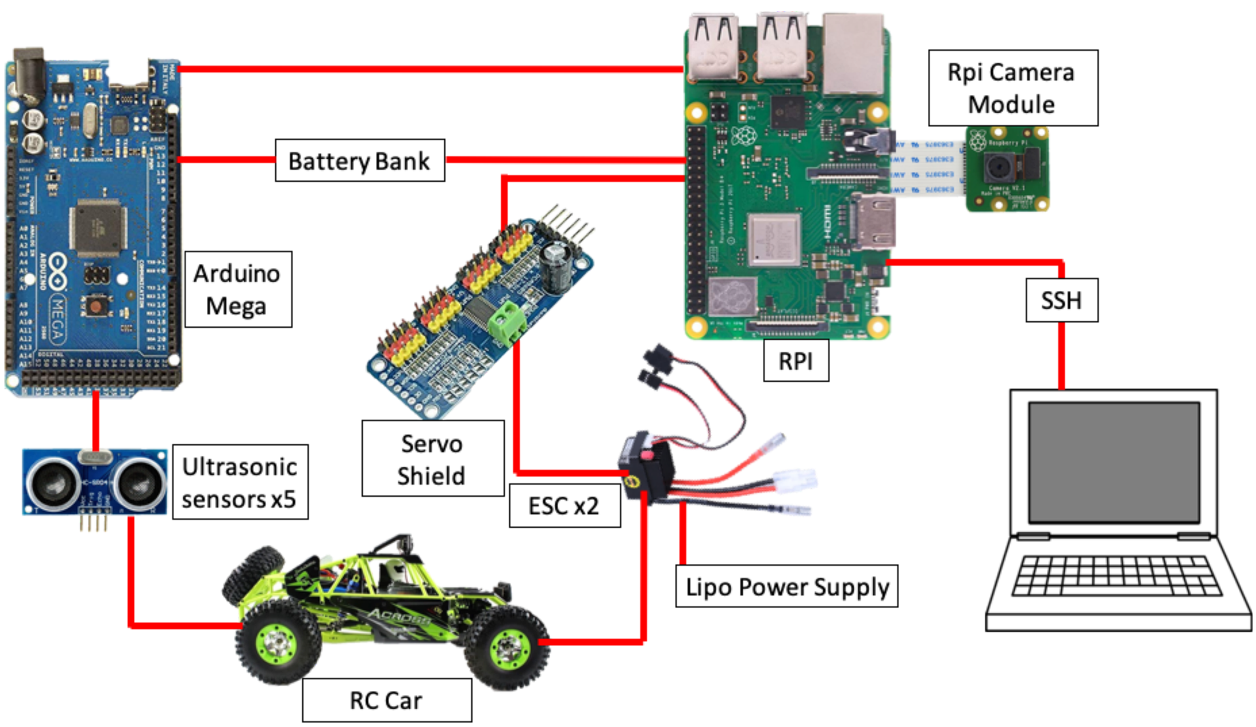
As part of our design requirements, we ensure that our design align with the regulations and standard that have been set for electronic devices and the vehicle industry. The Autonomous Car vehicle must adhere to IEEE standard 1582 in regards to non hazardous electronic systems and the stability of the device. Similarly The IEEE standard 11-2000 standard which applies to rotating electric machinery which forms part of the propulsion and major auxiliary equipment on internally and externally powered electrically propelled rail and road vehicles and similar large transport and haulage vehicles and their trailers where specified in the contract. Based on the requirement, our team chose list of parts needed for the project that is available online.

Solution Design

The Design flow can be shown as:



Above solution design looks like

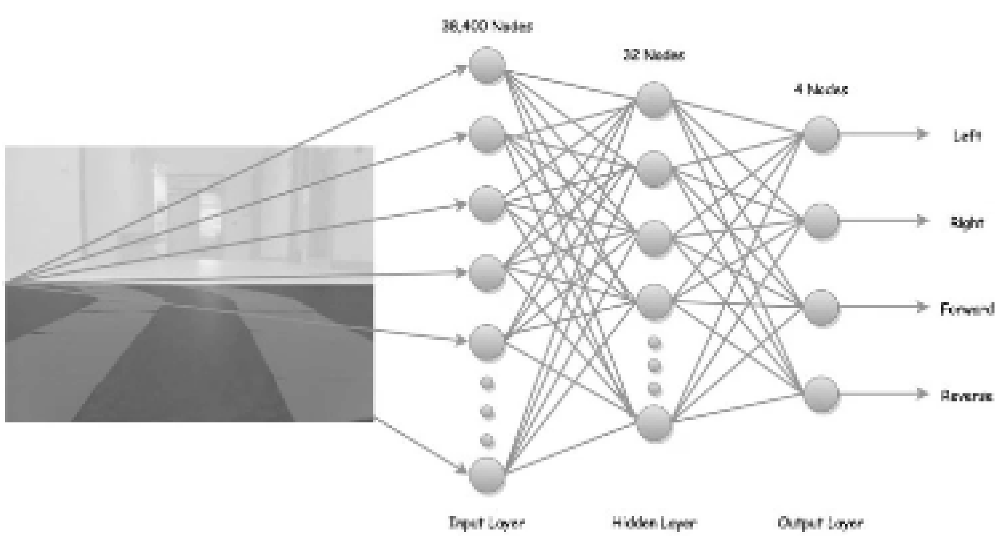


In the figure above we see the components in our system. The raspberry pi is home to most of our code, it runs a loop of code which constantly assess the camera feed from the raspberry pi camera module and outputs instructions for the car through the sda and scl ports to the servo shield. Two of the sixteen channels are used to control the rear throttle and front steering motors through their respective Electronic Speed Controllers (ESC)s. The servo shield outputs a pwm signal to the ESC in order to allow for specific angles of turning as well as speeds and effort from the back motor. This is powered by the LIPO power supply which provides a high discharge with a steady drop off rate.

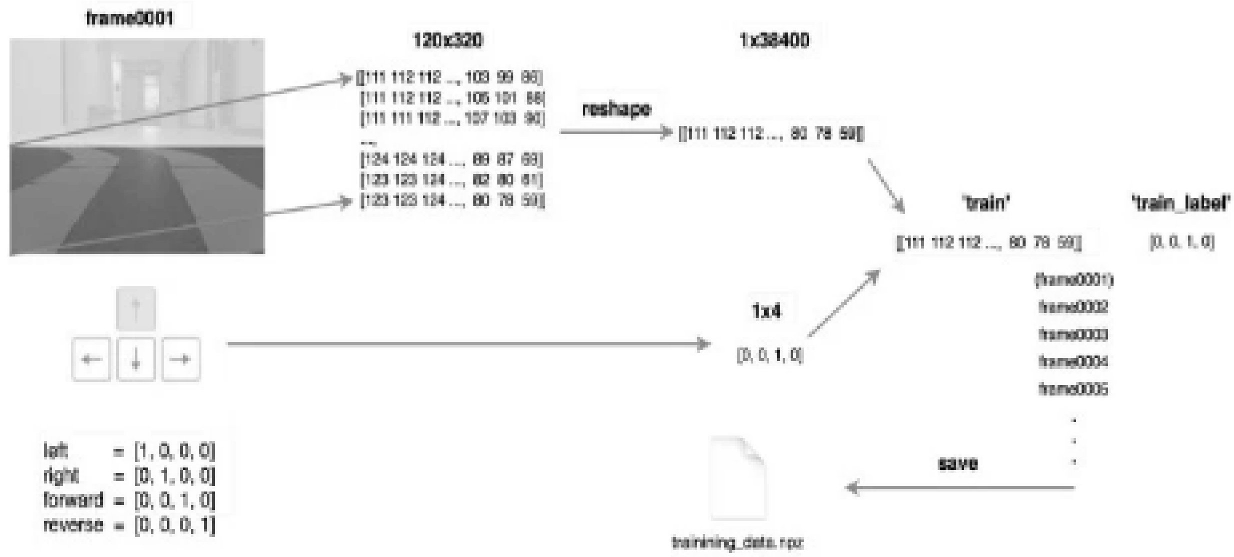
Simultaneously the raspberry pi also polls a serial port dedicated to the arduino. The arduino reads sensor data and reports it to the serial port either "STOP" if there is an object detected too nearby and "GO" otherwise. The raspberry pi will then reverse the throttle until the object is out of range then continue normal operation.

In order to start this entire system, one must ssh into the raspberry pi and start the script. This is also how one trains the car, once connected, you can manually control the car through a web interface which allows you to control the car with the keyboard a gamepad or virtual joystick. Once enough training data is collected it is transferred to the more powerful laptop from the raspberry pi and processed to produce the model.

The neural network runs on the raspberry pi is depicted in the following:



The working of neural network can be shown by the image below:



An advantage of using neural network is that once the network is trained, one only needs to load the trained parameters afterwards, making prediction very fast.

Project Implementation Plan

Week	Tasks	Members
Jan 21	**University Closed - MLK Day**	------
Jan 22 - Jan 27	**Setup code repo**	All
	Obtaining Parts (1)	All
	Send list of parts to be ordered to Dr. Kim/ Dr. Rawat	Satchin
	Meet with ME students	All
	Obtain temporary boards, sensors to use for testing/development	Savannah + Pawan
Jan 28 - Feb 03	**Begin Neural Net training and optimization**	Savannah + Pawan
	Decide on framework such as donkeycar	Savannah + Pawan
	Use existing software such as donkeycar (opencv, tensorflow) tools to train model	Savannah + Pawan
Feb 04 - Feb 10	**Integration of Controls into the RC car**	Satchin + Samantha
	Study RC Car control system	Satchin + Samantha
	Obtaining Parts (2)	Satchin
	Follow up on status of parts order	Satchin
Feb 11 - Feb 17	**Obtaining Parts (3)**	All
	Recieve Parts for RC Car, Ultrasonic, etc..	All
	Test all sensors, boards, and RC cars	All
	Meet with ME students	All
Feb 18	**University Closed - President's Day**	------
Feb 19 - Feb 24	**Test controlling RC car using the Arduino**	Satchin + Samantha
	Implement controls needed to accommodate our design	Satchin + Samantha
	Connect Rpi to arduino and test connection	Satchin + Samantha
Feb 25 - Mar 03	**Laptop Server Creation + RPI client creation**	Satchin + Samantha
	Choose laptop to use as host	All
	Test host connection speeds	All
	Integration of RPI and server	Satchin + Samantha
Mar 04 - Mar 18	**Integration of neural network into system**	Savannah + Pawan
	Deploy machine learning model	Savannah + Pawan
	Perform Initial test by training with simple route around completed track	Savannah + Pawan

	Optimize Neural with our system (test and iterate)	Savannah + Pawan
Mar 09 - Mar 17	**Spring Break**	------
Mar 18 - Mar 24	Midway Checkpoint	All
	Create track for testing	ME
	Determine appropriate location for testing	ME
	Obtain parts for track	ME
	Assemble track	ME
Mar 25 - Mar 31	**Car to Car Communication**	Satchin + Samantha
	Simple message passing setup with the server as host	Satchin + Samantha
	Test more complex messages and observe delay and response to heavy load, improve if necessary	Satchin + Samantha
	Designate test cases that require or may benefit from c2c communication	Satchin + Samantha
Apr 01 - Apr 07	**Car to Car Communication - cont.**	Satchin + Samantha
	Integrate car to car communication with RPI override for movement	Satchin + Samantha
	Implement override cases	Satchin + Samantha
Apr 08 - Apr 14	**Testing Final product**	All
	Test car to car communication	All
	Test speed control	All
	Test the behaviour of car in various situations	All
Apr 15 - Apr 18	**Testing Final product**	All
	Fix any glitches found during testing	All
	Perform final tests	All
	Prepare for EECS Day (Presentation + Poster)	All
Apr 19	**EECS Day**	All

Project Implementation Process

Obtaining Parts

We obtained the initial parts listed below :

Part Name	Link	Model	Quantity
Car	Link	Feiyue FY03 Eagle-3 1/12 2.4G 4WD	2
Raspberry Pi	Link	Pi3 B+	2
Ultrasonic Sensor (pack of 5)	Link	Elegoo HC-SR04	2
Jumper Wires	Link	Generic	1
RPI Camera Module	Link	Module V2-8	2
Arduino	Link	Mega 2560 REV3	2
IC Motor Control	Link	L293D	4

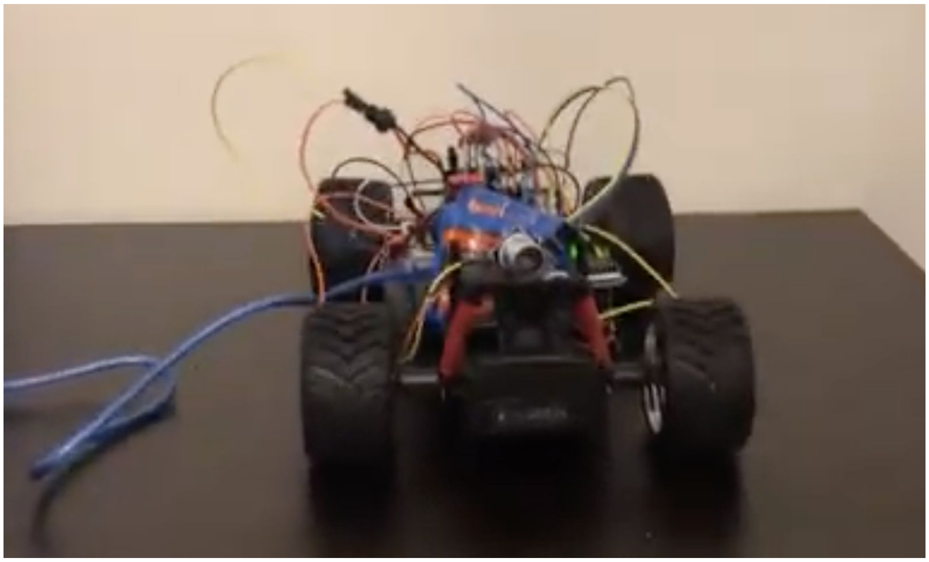
However as we decided to leverage the donkeycar framework and the car itself drew a lot of power, we obtained several new noteworthy pieces:

Part Name	Quantity	Purpose
ESC 10 Amps	4	To have variable control of the motors
Adafruit 16-channel PWM/Servo Shield	2	Take SDA and SCL input from RPI convert to PWM
LIPO Batteries 2S (7.4V)	2	Power ESCs/Motors
LEDs	~30	Indicate signals such as communication and bake
Router	2	For ease of ssh'ing into raspberry for development
Track (interlocking exercise mat tiles)	3	To have a controlled, consistent environment to train and run the model

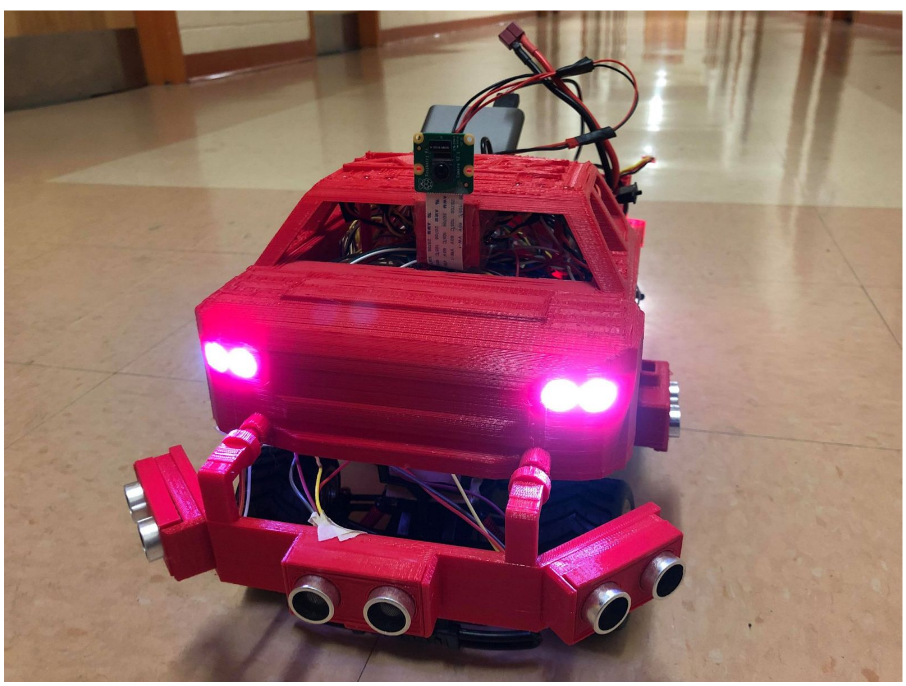
Assembling Car

We were able to setup the car and control the motors and basic movement of the car with the Arduino and L298N Bridge Motor Control and looked into controlling the raspberry pi through serial port which quickly became very messy, primarily from a control flow perspective.

Also the arduino control of the car was less than ideal with sharp movements and not many degrees and variation in speed and turning angle.



We then moved on to use a design in which the raspberry pi controls the movement of the car along with the electronic speed controller and the servo shield. The arduino was used mainly for collecting sensor data from the ultrasonic sensors, which is then passed on to the raspberry pi. To power all the components, a power bank was used to power the raspberry pi and arduino and the LiPo battery used to power the ESCs.



Assembling Sensor System

Our original design planned for 8 ultrasonic placed around the car however, we scaled it down to 5. We train the car to drive from camera placed in the front, reversing and avoiding obstacles while reversing was deemed unnecessary.

Five was enough to give us the information and behaviour we were trying to achieve which is to have an override reaction to stimuli from the environment that conflicted with the camera and trained model response.

Assembling Track

To assemble the track, we ordered 3x orders of Interlocking Exercise Mat Foam Tiles and assembled them in a 7 x 5 tile configuration to have a ~10 x 14ft space to drive the car.

Implementing V2V Communication

To implement vehicle-to-vehicle communication we utilized UDP Client-Server python programming using the socket library to allow each raspberry pi to send packets between each other while on the same network. We were able to gather the IP address of each Pi on the network from our laptops using nmap tool in the terminal. Using the GPIO library for Raspberry Pi we made an LED blink whenever a packet was received, wait 0.5s, then send a packet in response.

Training Model

For training, the car is manually driven around the track for about 20 laps. The collected images are converted to numpy array and paired with training label (human input). This pair is feeded to the neural network which output the weights which is saved in npz format. This training is done in keras using back propagation method. Now, to test this model, similar neural network as training is constructed, the trained weight is loades, RPi camera feeds the images to the neural network real time and one of four choices is made- left, right, forward and reverse.

Conclusion

We aim to meet the customers' need for safer transportation by developing a secure autonomous car system. Our final design uses a R/C car and custom track to demonstrate the functionality of our system. We have laid sufficient groundwork for the development of an autonomous car system with a framework for communication.

Future goal is to continue to train model to operate on more complex tracks and implement more realistic functionality like rerouting around obstacles, turning at intersection, yielding for "pedestrians", recognize and respond to traffic signals/lights. The next group will further develop V2V communication to communicate more meaningful data based on added functionality of the autonomous car system. Future work also includes developing and implementing lightweight cyber-security features and privacy aware communication.

References

Federal Highway Administration. 2016. Traffic Volume Trends, January-December 2016. Washington, DC: U.S. Department of Transportation. Available: https://www.fhwa.dot.gov/policyinformation/travel_monitoring/tvt.cfm

Karla, Nidhi, “Driving to Safety”, rand corporation, 2016

Vijh, Rahul, “Autonomous Cars - Patents and Perspectives” ipwatchdog.com, 2016

Plumer, Brad, “Big challenges that self-driving cars still have to overcome” vox magazine, 2016