A Proposal To Bring Fully Autonomous Small Scales Vehicles Into U.S. Road Infrastructure

Background:

The growing demand for autonomy in vehicles has skyrocketed in the last decade due to the surge in Machine Learning advancements. The techniques, algorithms, and versatility of these learning processes have garnered this attention of the world stage as being the prolific means to increasing the need for the idea big data to solve some of our most pressing technological problems. [2]

Machine Learning began its journey alongside the invention of computers. Early developers of the digital transistor based computer began hypothesizing about the potential of creating a processing unit that is able to meet the complexity of a human brain. Alan Turing created the Turing Test in the 1950s, which states that in order for a computer to be classified as having intelligent properties in must be able to trick a human into believing it is human as well. In the 1990's IBM's Deep Blue had beaten the world's greatest Chess player. This show cased to the public convincingly enough that the dawn of computer intelligence was revivaling that of human intelligence. Fast-forward to today and the expanse of deep learning has likely hit every branch of science as a means of proving the viability of an assumption through immense computational design from large data sets. [1]

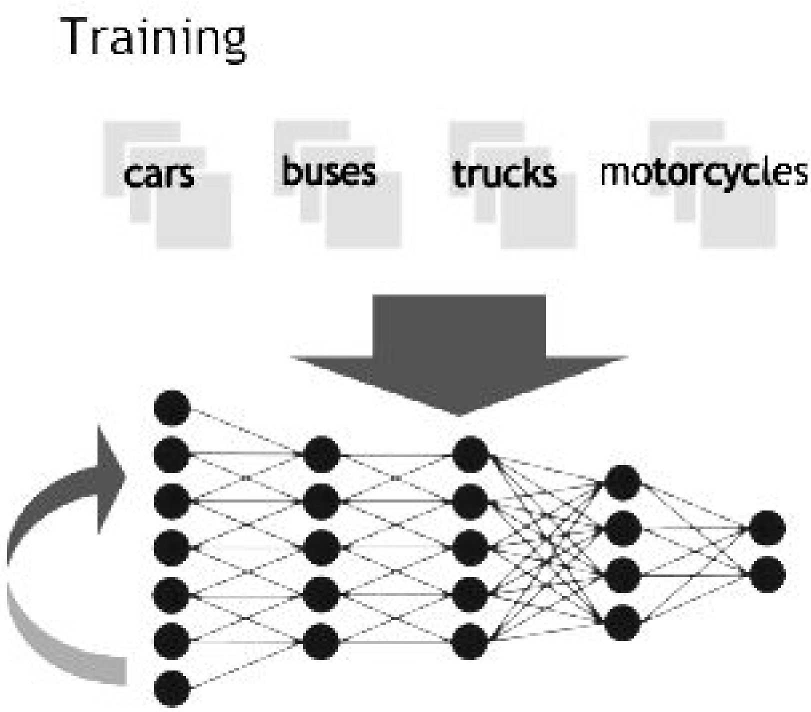


Figure 1: Simplification of designing deep learning model. Shows the the use of layered approach for nodes [3]

One of the largest advancements in the field of Machine Learning is Deep Learning. Deep Learning has proven to be most exceptional for any kind of problem analysis because of its ability to gather high level features from a class of data. These patterns can be divided between easily recognizable, and unrecognizable by the person employing the data. This recognition of pattern analysis helps form the question of how the Deep Learning model should be employed. **Figure 1.)** shows the basic design philosophy of all deep learning models. One of the largest problems that automotive companies are trying to spearhead using these learning algorithms are fully autonomous vehicles. A car that is fully autonomous means that it can operate entirely without the need for human intervention. To solve this problem a variety of different engineering and computer science specializations are needed to carry out the navigation requests no matter the difficulty. [2]

The major reason to employ autonomous vehicles is to reduce the number of car accidents. Car accidents make up a large percentage of the world's annual death toll. To achieve this engineers have been employing various techniques to remove human error from the equation. There are potentially even more benefits when switching to a fully autonomous network of cars like a reduction in government spending for traffic police, mental relief from long commutes, a reduction in the traffic congestion as vehicles are able to successfully maneuver amongst each other with minimized error, and the deployment of extremely convenient and cheap taxi systems. However, before all of these characteristics can be taken advantage of, there are still major hurdles for car companies to endure before this fully autonomous system becomes a reality. A secure network needs to be made to defend against criminal or terrorist activities that aim to dismantle the car system. Software failure is still an issue where irregular patterns can cause incorrect decisions in a high risk scenario. Any malfunction of hardware can result in constant vehicular accidents.

Objective:

The objective of this Research Proposal is to address some of the problems with modern day semi-autonomous car systems of which include, navigating a car from a central control system to eliminate problems that will occur from using different algorithmic models for driving in different vehicles. In order to meet this objective this control system of the car must be connected to the logical control systems of the computer software associated with it. The control system on board the computer will handle the image processing aspect to relay basic steering maneuvers back to the vehicle. This can be done by creating a TCP Server on board the vehicle where much of the throughout will be used to provide video streaming to the central computer. The second part of achieving this objective is to communicate a plethora of sensor information back to an onboard microcontroller system for faster on site reactionary vehicle decisions. By handling cruise control the amount of required throughput to the central computer is greatly reduced and simplified. Introducing a system for quick breaking will allow the vehicle to respond much quicker to unexpected obstacles without having to rely on the possibly more latent computer system. Integrating both systems into one homogeneous system will require communication high level communication protocols established by the members of this Senior Design Team.

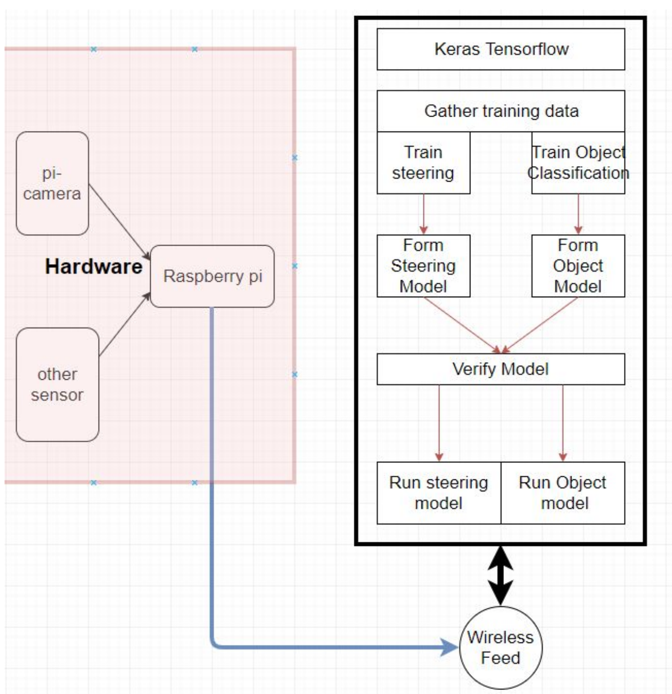
Challenges:

Autonomous Vehicle Systems are highly variablistic and thus require many mechanisms and fault handling systems for the multitude of different driving scenarios.

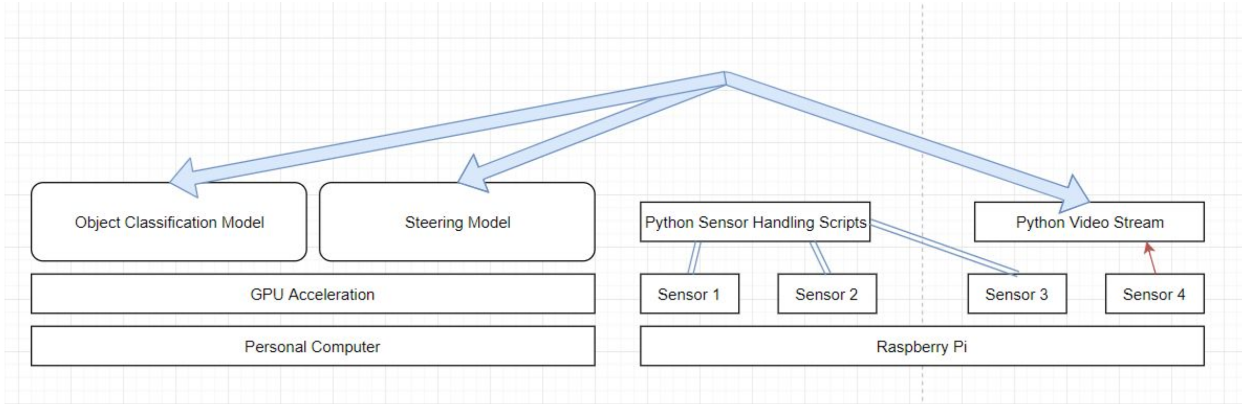
1. Object Classification for traffic sign recognition so autonomous vehicles can be introduced into the existing transportation infrastructure.
2. Feedback system for maintaining constant speed given varying road conditions for cruise control
3. Ensuring mechanical stability given the increased number of sensors and power demands of the new system
4. Merging on board vehicle control systems with remote vehicle control systems
5. Keeping latency low for both hardware systems to ensure adequate performance levels for real time driving
6. Establish the environmental path the car will travel on and the track it will follow.
7. Determining the exact methodology for the acceleration and braking mechanisms.

Design And Feasibility:

Software



a.)



b.)

Figure 2: Abstract structure of software system for onboard and offsite control. Image a.) shows the communication link between the deep learning Keras model and the raspberry pi. Image b.) shows the software layers between each of the separate control systems and how they stack on top of each other in a hierarchical approach.

Running the model for classification and steering are resource intensive we will be sending a video stream back to our computer to handle the load and then submit steering details back to the car. Using the object classifier we will also be observing the various objects in the scene to relay information to the accelerator in the vehicle. If the vehicle observes a stop sign it will begin to stop. If a vehicle observes a stop light it will register the color of the light and correctly map the appropriate deceleration. If it recognizes people it will attempt to either come to a complete stop or maneuver around the person. This control system is illustrated in **Figure 2).**

Below I outline the procedure for creating the systems in place for image processing.

<u>CNN For Road Sign Classification</u>

Step 1

- Create training data in this format : TFRecord File Format using the Oxford Pet Dataset
- Begin scraping photos for the things i would like to classify
  - Ideally would like to get 200 images for each item
- Install all dependencies for tensorflow like numpy, Matlpotlib, OpenCV
  1. Red Light
  2. Green Light
  3. Yellow Light
  4. Person
  5. Stop sign
- Then I begin labeling each of the images, all 1000 of them and then convert them to the TFRecord File Format

Step 2

1. Need to create object training pipeline

2. I will use mobile-net SSD configuration file as base template for training process
3. Use data augmentation procedure to generate more images that slightly different to remove biases
4. *Create a Keras Model*
    a. Sequential Model
    b. Add input layers
    c. Add hidden layers
    d. Add activation layers
    e. Create training generator
    f. Begin training process with configured settings
    g. Save weights
5. *Begin Testing*
    a. Test using webcam on computer for basic recognition of items
    b. Adjust values and params as needed

<u>CNN for Steering</u>

I will be using Keras to build the steering model so I can apply the same methodology

Step 1

1. *Create Keras Model*
    a. Add input layers
    b. Add hidden layers 3 of 5x5 convolutional layers, using a stride of 2x2
    c. Add activation layers (Relu)
    d. Add series of fully connected layers
    e. Add size 1 output layer for degree of steering
    f. Use Batch Normalization for faster convergence
        i. Begin training process with configured settings
        ii. Save weights

Step 2

2. *Begin Testing*
    a. Test using video data from GTA V vehicle on test-track
    b. Adjust values and params as needed

Training Data

There are several options on how to receive training data.

- Using a video game such as GTA V to mimic the driving that we will see from the vehicle that we will be making
- Using a video camera placed on a hollow version of our vehicle and manually driving it around a track that we will be using
- There exists training data for driving in the real world that already exists
- Generate data using server simulator CARLA
- Use Udacity's self driving approach for simulation.

Responsibilities:

Diego(CpE) - Handle image processing for steering and object classification for traffic sign recognition on offsite central computer system

Savis (EE) - Work on deriving mathematical modeling of the cruise control and receiver

Suri (EE) - working on the cruise controller, mechanical and also mathematical modeling of the control feedback system including microcontroller,motor and process system

Samantha (EE) - Calculate system components needed to understand the car's functionality. Determining the correct equipment needed to implement a powered car.

Timeframe:

Software

Fall Quarter

Week	Outline
1	Create group and find mentor
2	Meet with mentor and decide project
3	Create plan of attack and begin work on CNN for steering
4	Continue work CNN for steering (Find means of collecting training data)
5	Continue work CNN for steering (Gather training data)
6	Continue work CNN for steering (Build model)
7	Continue work CNN for steering (Begin training process)
8	Begin work on object classifier and detection
9	Begin work on object classifier and detection + Finals
10	Finals

Winter Quarter

Week	Outline
1	Test and Debug Phase object classifier and detection
2	Test and Debug Phase object classifier and detection
3	Test on built setup for car
4	Test on built setup for car + Research Essay

5	Test on built setup for car + Research Essay
6	Research Essay + Poster Board + testing
7	Research Essay + Poster Board + testing
8	Poster Board + testing
9	Continued testing and optimization
10	Finals

Hardware
Fall & Winter Quarter

10/15-10/22	Initial modeling of cruise Controller
10/22-10/29	Apply for Urop funding project. Look into building a transceiver for instruction sendings to the car
10/29-11/5	Complete list of items we need to buy considering our budget.
11/5-11/12	Midterm Week
11/12-11/26	Come Up with the final plan about the design and implementation techniques(the transfer function of the Cruise control and its gain and constant)
11/26-12/9	Start working on transceiver Design(Remote Controller Design)

The build of the autonomous vehicle will increase the expenses of the project in order to create a successful on road car. The expected cost of the car will be $300, which will go directly to the hardware needed for the car. An additional cost will be attributed to using cloud computation services to train the model.

References:

1. Marr, Bernard. "A Short History of Machine Learning -- Every Manager Should Read." Forbes Magazine. Web Article. Feb 19, 2016. [Accessed Online] Oct 23, 2018. Available: https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/#47602f9d15e7

2. Bimbraw, Kishav. “Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology.” IEEE. 2015 12th International Conference on Informatics in Control, Automation and Robotics (ICINCO). Web Article. July 2015. [Accessed Online] Oct 23, 2018. Available: https://ieeexplore.ieee.org/document/7350466
3. Brown, Larry. “Deep Learning For Image Classification, GEOINT Training”. Slideshow Event. June 2015. [Accessed Online] Oct 23, 2018. Available: https://www.nvidia.com/content/events/geoInt2015/LBrown_DL_Image_ClassificationGEOINT.pdf