Self-Driving-Car Simulation

Rohini Mishra

August 4, 2019

Abstract

Over the last few years, the automotive industry has advanced significantly towards a future without human drivers, but building an autonomous vehicle which is safe, cost-efficient and reliable is not an easy task. This report contains details on how to deploy behavioral-cloning and deep neural networks to program a self-driving-car, which uses a simulation software as a test-bed for training the model. The proposed model of a driver-less car has been implemented in Python programming language using various tools such as- pandas, numpy, keras, scikit-learn, flask-socketio, matplotlib, harfbuzz and so on. The model has been trained using a 9-layer Convolutional Neural Network and simulated using the Udacity open source self-driving-cars simulator. This report also touches on the fact that these artificially intelligent vehicles can be a real boon to mankind and can create a future where accidents will occur only due to car failures, sensor failures, or software bugs and the latter will only happen once because it gets added to a regression test and all the cars running that software automatically learn how to avoid that mistake. However, a future world where only computer driven cars run on the roads does not seem to be very near, because problems with autonomous systems in the real-world are numerous. This report also brings into light various technological, social and political challenges involved in making autonomous vehicles mainstream.

1 Introduction

Automobile manufacturers such as Ford, General Motors, Tesla, and other companies such as NVIDIA are investing billions of dollars in autonomous vehicle driving research. According to Intel, by 2050, this fast-growing industry will be worth 7 trillion dollars [1]. Governments of countries in Europe and of the USA are creating regulations for self-driving cars, as a result of the latest advances of the industry.

The benefits of fully autonomous vehicles can go way beyond removing the need of a human driver. Transportation services such as Uber will start using self-driving cars instead of human drivers, and might become a cheaper and better alternative for end consumers than owning a car. This will represent a shift on the way cities are planned, as fewer parking places will be needed, and most importantly, in a smart city with most of its vehicles being connected and autonomous, traffic optimization will be able to be heavily applied by coordinating movement.

In contrast, there are many problems that need to be addressed regarding autonomous vehicles. Tesla's car autopilot first fatal crash in 2016 [2], for example, brought up discussions about reliability, safety and legal liability regarding autonomous vehicles among researchers. Other important topics, such as security and transparency will need to be heavily discussed as these vehicles become mainstream. Another problem of this relatively new field is the lack of data sharing between vehicles. Autonomous vehicles are complex systems that rely heavily on technologies such as LIDAR, GPS, high-definition maps, and artificial intelligence for navigation and collision avoidance. This means that each autonomous vehicle is constantly collecting and analyzing a very high volume of data, which according to Intel is about 2.6 terabytes per hour [3].

Different cars are capable of different levels of self-driving- from driver-only to robo-taxi. **The Society of Automobile Engineers (SAE) described vehicular automation on a scale of 0-5:**

- Level 0: All major systems are controlled by humans
- Level 1: Certain systems, such as cruise control or automatic braking, may be controlled by the car, one at a time
- Level 2: The car offers at least two simultaneous automated functions, like acceleration and steering, but requires humans for safe operation
- Level 3: The car can manage all safety-critical functions under certain conditions, but the driver is expected to take over when alerted
- Level 4: The car is fully-autonomous in some driving scenarios, though not all

- Level 5: The car is completely capable of self-driving in every situation

Levels 1-3 require a licensed driver, but levels 4 and 5 allow driver-less operation, which is necessary for many predicted benefits.

2 Motivation

Figure 1: *According to a statistical survey the cause behind 94% of car accidents includes human element [4].*

According to a brief statistical summary published by NHTSA's National Center for Statistics and Analysis showed that a mere two percent of accidents were caused by the environment, another two percent were caused by the vehicles, and 2 percent came from "unknown" causes. A full 94 percent, meanwhile, where caused by human error. Mistakes that drivers made led directly to accidents.

These stats worked as a motivation for building a model of a self-driving car which can prove to be beneficial in a number of ways:

- Greater Road Safety- Higher levels of autonomy have the potential to reduce risky and dangerous driver behaviors. The greatest promise may be reducing the devastation of impaired driving, drugged driving, unbelted vehicle occupants, speeding and distraction.
- Greater Independence- People with disabilities, like the blind, are capable of self-sufficiency, and highly automated vehicles can help them live the life they want. These vehicles can also enhance independence for seniors.
- More Productivity- Wide-scale deployment of HAVs could allow drivers to recapture time. In the future, HAVs could offer the convenience of

dropping vehicle occupants at their destination, whether an airport or shopping mall, while the vehicle parks itself. In a fully automated vehicle, all occupants could safely pursue more productive or entertaining activities, like responding to email or watching a movie.

- Reduced Congestions- Several causes of traffic congestion could be addressed by HAVs.Fewer crashes or fender benders mean fewer roadway backups. HAVs maintain a safe and consistent distance between vehicles, helping to reduce the number of stop-and-go waves that produce road congestion.

3 Literature Review

3.1 The Idea

The idea of self-driving vehicles dates back to Futurama, an exhibit at the 1939 New York World's Fair. A renowned industrialist named Norman Bel Geddes put forward a vision was remarkable as it inspired many curious minds. It proposed an automated highway system in which cars moved autonomously, allowing passengers to arrive at their destinations safely and in an expedient manner.

3.2 Highway of the Future

RCA, in collaboration with General Motors and the state of Nebraska, ran with the idea and began working on an automated highway technology modeled after Bel Geddes' original concept. In 1958, the team unveiled a 400-foot stretch of automated highway outfitted with electronic circuits built into the pavement. The circuits were used to gauge changing road conditions as well as help steer the vehicles traveling along that part of the road. It was successfully tested and in 1960 a second prototype was demonstrated in Princeton, New Jersey. As part of their involvement in the project, General Motors even developed and promoted a line of experimental cars that were custom built for these smart roads of the future.

Around the same time, officials at the United Kingdom's Transport and Road Research Laboratory began trialing their own driverless car system. The RRL's guidance technology was somewhat similar to the short-lived automated highway system in that it was both a car and road system. In this case, the researchers paired a Citroen DS retrofitted with electronic sensors with a magnetic rail track that ran beneath the road.

Unfortunately, like its American counterpart, the project was eventually scrapped after the government opted to cease funding. What this meant going forward was that to be feasible any work on autonomous cars would require at least a slight shifting of gears, with more of an emphasis on figuring out ways to making the car smarter rather than the road.

3.3 A Change in Direction

Engineers at Stanford were among the first to build on this renewed approach. In 1960, a Stanford engineering graduate student named James Adams set upon constructing a remote-controlled lunar rover. He initially assembled a four-wheeled cart equipped with a video camera to improve navigation and over the years the idea evolves into a much more intelligent vehicle capable of self-navigating past a chair-filled room all on its own.

In 1977, a team at Japan's Tsukuba Mechanical Engineering Laboratory took the first major step of developing what many consider to be the first

stand-alone autonomous vehicle. Rather than relying on external road technology, it was guided with the help of machine vision. The prototype was capable of speeds close to 20 miles per hour and was programmed to follow white street markers.

Interest in artificial intelligence as it applied to transportation grew in the 80's thanks in part to the pioneering work of a German aerospace engineer named Ernst Dickmanns. His initial effort, backed by Mercedes-Benz, resulted in a proof-of-concept capable of driving autonomously at high speeds. This was achieved by outfitting a Mercedes van with cameras and sensors that collected and fed data into a computer program tasked with adjusting the steering wheel, brake, and throttle. The VAMORS prototype was tested successfully in 1986 and a year later debuted publically on the autobahn.

3.4 Big Players, Big Investments

This led to the European research organization EUREKA launching the Prometheus project. With an investment of 749,000,000 euros, Dickmanns and researchers at Bundeswehr Universität München were able to make several key advances in camera technology, software and computer processing that culminated in two impressive robot vehicles, the VaMP and VITA-2.

The following year, engineers at Hughes Research Labs showcased an autonomous car capable of traveling off-road. In 1996, engineering professor Alberto Broggi and his team at the University of Parma initiated the ARGO project with an aim to show that a car can be turned into a fully-autonomous vehicle with minimal modifications and low-cost parts. The prototype they came up with, a Lancia Thema equipped with little more than two simple black-and-white video cameras and a navigational system based on stereoscopic vision algorithms, ended up running surprisingly well as it covered a route of more than 1,200 miles at an average speed of 56 miles per hour.

3.5 Google enters the Race

In 2010, internet giant Google announced that some of its employees had spent the previous year secretly developing and testing a system for a self-driving car in the hopes of finding a solution that would reduce the number of car accidents each year by half. The project was headed by Sebastian Thrun, director of Stanford's Artificial Intelligence Laboratory, and brought onboard engineers who worked on cars that competed in DARPA's Challenge events. The goal was to launch a commercial vehicle by the year 2020.

The team started out with seven prototypes, six Toyota Priuses and an Audi TT, which were souped up with an array of sensors, cameras, lasers,

a special radar and GPS technology that allowed them to do much more than just circumnavigate a pre-determined route. The system can detect objects such as people and numerous potential hazards up to hundreds of yards away. By 2015, Google cars had logged more than 1 million miles without causing an accident, though they were involved in 13 collisions. The first accident for which the car was at fault occurred in 2016[2].

They lobbied for and got legislation passed to make self-driving cars street legal in four states and the District of Columbia, unveiled a 100 percent autonomous model it plans to release in 2020 and is continually opening up testing sites throughout the country under a project named Waymo.

Other companies that have started developing and testing autonomous car technology include Uber, Microsoft, Tesla as well as traditional car manufacturers Toyota, Volkswagon, BMW, Audi, General Motors and Honda. Progress on advancing the technology took a major hit when an Uber test vehicle hit and killed a pedestrian in March of 2018[5]. However, the entire industry hopes to achieve the dream of an automated and much safer road transport system in the near future.

4 Proposed Methodology

4.1 Approach Followed

After a thorough reading of all the resources, I decided to do something called behavioral-cloning to capture the human subcognitive-skills used while driving and reproduce it into a program. I have used a nine-layer Convolutional Neural Network to build a model which can be trained, validated and tested in a virtual environment. I have written the testing and training scripts in python programming language with the help of libraries such as keras, sci-kit, numpy, pandas, harfbuzz, matplotlib, etc and used a simulator where one can steer a car around a track for data collection. The image data and steering angles thus produced are used to train the nine-layer CNN and then this model is deployed to drive the car autonomously around the track.

4.2 About The Simulator

I have used Udacity's Self-driving cars Simulator. Udacity built this simulator with unity for its Self-Driving Car Nanodegree program, to teach students how to train an automated vehicle to navigate along roads using deep-learning. They recently open-sourced the simulator and it can easily be downloaded and used for training and testing our own models. It does not require installation of any dependencies. It just requires downloading this zip file[6], which can be extracted to run the executable file present inside it.
Note: The link in the reference is for Linux users. However, Windows and Mac users can easily install it from Github.

Figure 2: The Simulator

4.3 Data Collection, Pre-processing and Batch Data Generation

The simulator has two modes- training mode and testing mode.

Figure 3: Modes of the Simulator

To collect data, run the simulator and drive the car in training mode. You can control the car using W-A-S-D keys. Press 'R' to start recording and after 1-5 lapse(ideally), press 'R' again to capture the data and save it to a csv file. The first three columns of the file contain 'left', 'right' and 'centre' image. The next four columns contain 'speed', 'throttle', 'steering angle' and 'brake'.

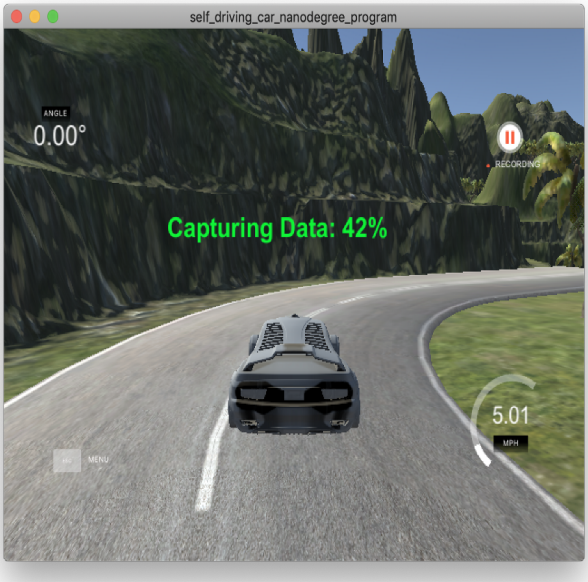
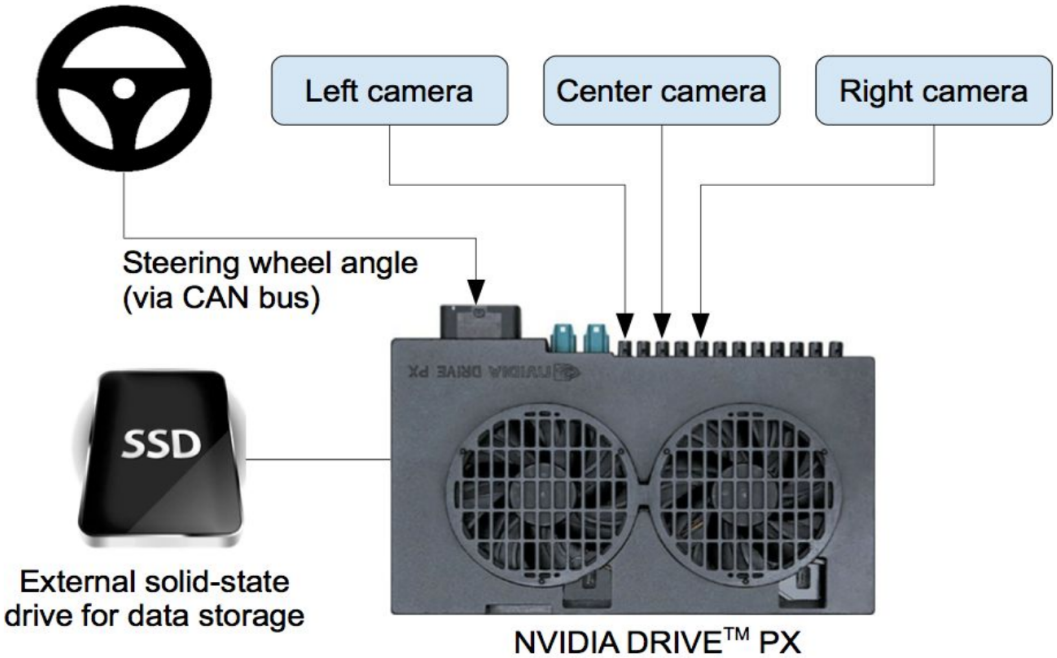


Figure 4: Capturing data

In order to make the system independent of the car geometry, the steering command is 1/r, where r is the turning radius in meters. 1/r was used instead of r to prevent a singularity when driving straight (the turning radius for driving straight is infinity). 1/r smoothly transitions through zero from left turns (negative values) to right turns (positive values).

The high-level view of this data generating model may look something like this:



The collected data needs to be pre-processed and augmented to create a batch of data on to which the model can be fitted.To perform all these operations, we have a python program called 'utils.py' besides our testing and training script. While preprocessing, we crop the image in order to remove the sky at the top and the car front at the bottom. Then we resize the image to the input shape used by the network model and finally convert the image from RGB to YUV. After selecting the final set of frames we augment the data by adding artificial shifts and rotations to teach the network how to recover from a poor position or orientation. The magnitude of these perturbations is chosen randomly from a normal distribution. Finally, using a batch generator function, a batch of images and associated steering angles of a given batch-size is produced.

4.4 Network model

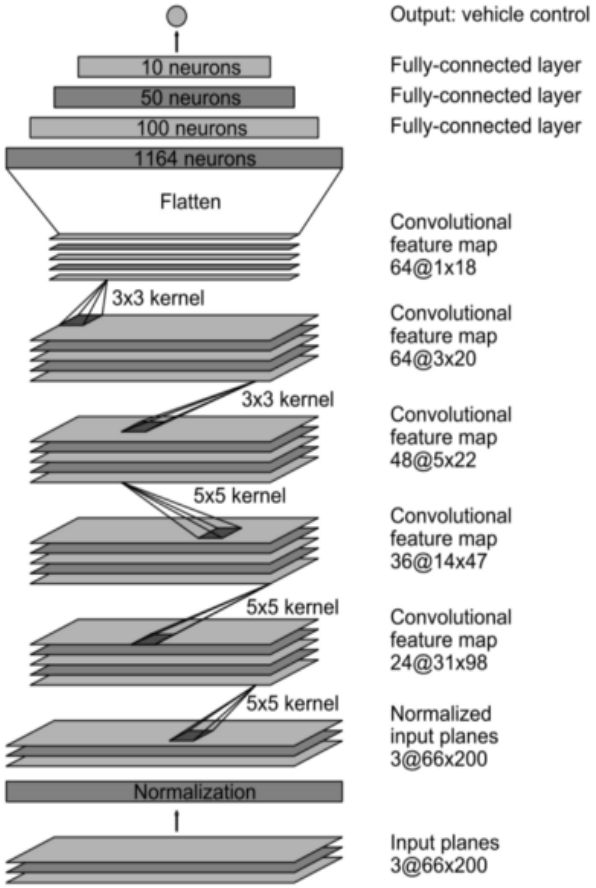


Figure 5: Network Architecture- CNN architecture. The network has about 27 million connections and 250 thousand parameters.

We train the weights of our network to minimize the mean squared error between the steering command output by the network and the command of either the human driver, or the adjusted steering command for off-center and rotated images. Our network architecture is shown in Figure 5. The network consists of 9 layers, including a normalization layer, 5 convolutional layers and 3 fully connected layers. The input image is split into YUV planes and passed to the network. The first layer of the network performs image normalization. The normalizer is hard-coded and is not adjusted in the learning process. Performing normalization in the network allows the normalization scheme to be altered with the network architecture and to be accelerated via GPU processing. The convolutional layers were designed to perform feature extraction and were chosen empirically through a series of experiments that varied layer configurations. We use strided convolutions in the first three convolutional layers with a 2×2 stride and a 5×5 kernel and a non-strided convolution with a 3×3 kernel size in the last two convolutional layers. We follow the five convolutional layers with three fully connected layers leading to an output control value which is the inverse turning radius. The fully connected layers are de-

signed to function as a controller for steering, but we note that by training the system end-to-end, it is not possible to make a clean break between which parts of the network function primarily as feature extractor and which serve as controller.

Some important points regarding this architecture are as follows:

- Normalization is done to avoid saturation and make gradients work better.
- The activation function used here is exponential linear unit. The reason why we chose ELU it speeds up the training process and also solves the vanishing gradient problem.
- The convolution layers are meant to handle feature engineering.
- The fully connected layers are for predicting the steering angle.
- Dropout avoids overfitting. Dropout refers to ignoring units. More technically, At each training stage, individual nodes are kept with probability p, so that a reduced network is left. In our model, we have used p=0.5.
- 10 epochs of 20000 samples per epoch and batch size 40 are generated.

5 Workflow

Here are a few block diagrams to show how the model is developed:

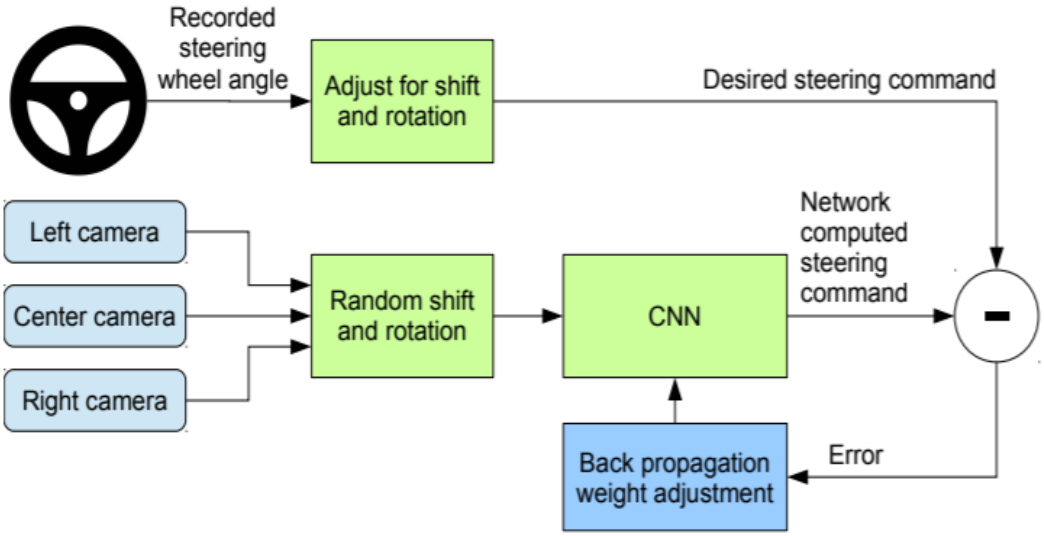


Figure 6: Training The Neural Network

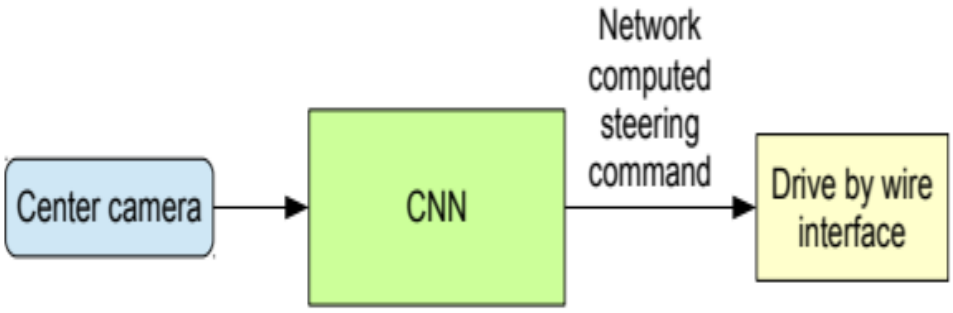


Figure 7: The trained network is used to generate steering commands from a single front-facing center camera

6 Testing Mode

After training the model, run the simulator in autonomous mode and type the following command in the terminal:

python drive.py model.h5.

You will find that the car runs on the track without any manual intervention.

Figure 10: Autonomous Mode

Testing mode is somewhat like a feedback loop.

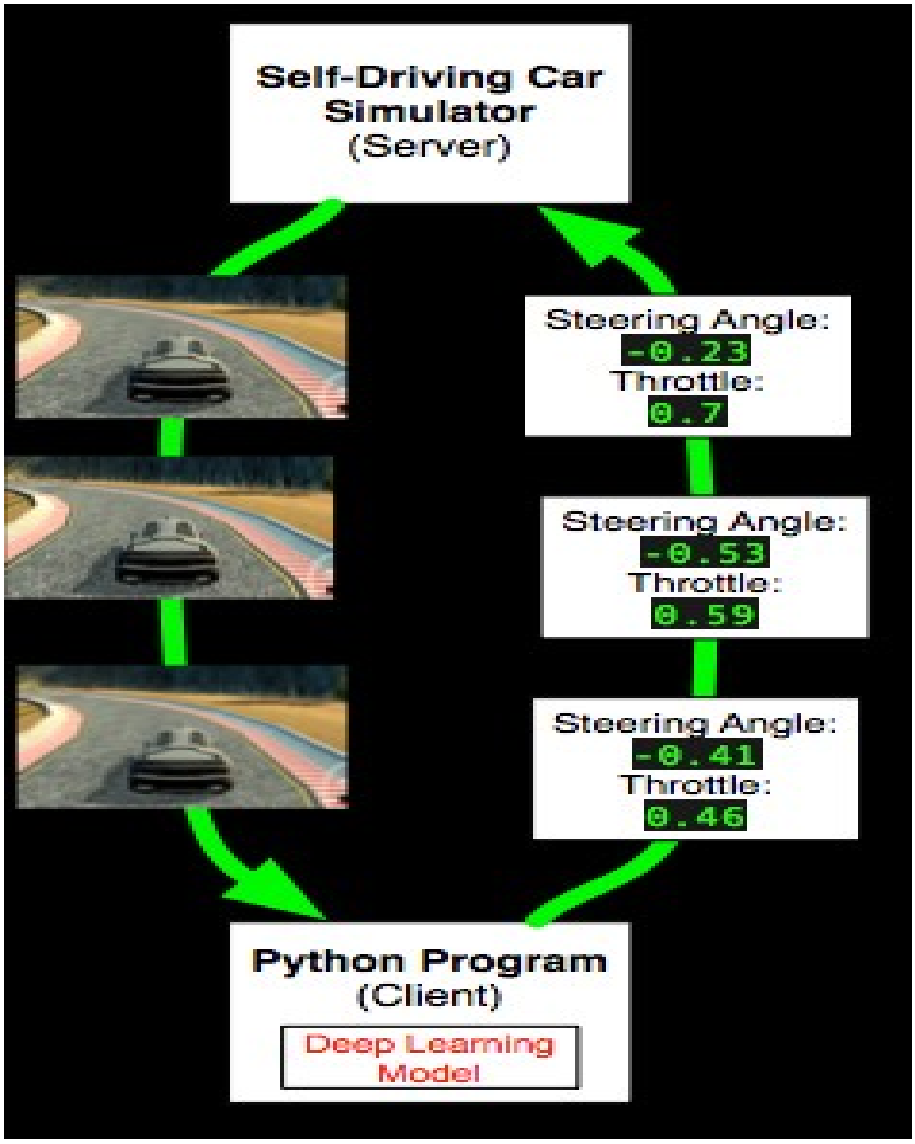


Figure 10: Client-Server loop.

7 Result and Conclusion

We have empirically demonstrated that CNNs are able to learn the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, path planning, and control. A small amount of training data from less than a hundred hours of driving was sufficient to train the car to operate on two entirely different tracks. The CNN is able to learn meaningful road features from a very sparse training signal (steering alone). The system learns for example to detect the outline of a road without the need of explicit labels during training. More work is needed to improve the robustness of the network, to find methods to verify the robustness, and to improve visualization of the network-internal processing steps. With a little more effort, we may also be able to train the car to run in diverse conditions, on highways, local and residential roads in sunny, cloudy, and rainy conditions.

References

[1] The Telegraph. Self-driving cars will add $7 trillion a year to global economy, says Intel. 2017. URL: http://www.telegraph.co.uk/business/2017/06/ 02/self-driving-cars-will- add-7-trillion- year-global-economysays (visited on 07/06/2019).

[2] The Guardian. Tesla driver dies in first fatal crash while using autopilot mode. 2016. URL: https : / / www . theguardian . com / technology / 2016 / jun / 30/tesla- autopilot- death- self- driving-car- elon- musk (visited on 07/06/2019).

[3] Kathy Winter. For Self-Driving Cars, There's Big Meaning Behind One Big Number: 4 Terabytes. https://newsroom.intel.com/editorials/selfdriving - cars - big - meaning - behind - one - number - 4 - terabytes. (visited on: 10/06/2019)

[4] https://crashstats.nhtsa.dot.gov/Api/Public/ ViewPublication/812115 (visited on: 15/06/2019)

[5] Wikipedia-The death of Elaine Herzberg: (August 2, 1968 – March 18, 2018)It was the first recorded case of a pedestrian fatality involving a self-driving (autonomous) car, following a collision that occurred late in the evening of March 18, 2018. (visited on: 15/06/2019)

[6] https://s3-us-west-1.amazonaws.com/udacity-selfdrivingcar/Term1-Sim/term1-simulator-linux.zip