Self Driving Car

Samyak Jain

UG Student, Dept. of Computer Science Engineering, Chandigarh University, Mohali, India

---

Abstract - *In the modern era with increasing needs of life, Self Driven Car will also become one of basic requirements. These vehicles are focused to be automated to give the driver relief from time in busy schedules, traffic, parking and such other strains of driving etc.*

Google started working on self driven car since 2010 and still working on it to give the best self driven car. In this paper I will be focusing on how to build a self driven car what are tools required and what is the path of building it. The computer science knowledge needed to build a self driven car. Self Driven car measure input, track a track and send commands to various actuators that control speed increase, slowing down, and directing. The programming tracks traffic through hard-coded rules, preventive calculations, prescient displaying, and "brilliant" segregation on objects, assisting the product with following rules on transport.

Key Words: Self Driving Car, Deep Learning, Convolution Neural Network, Traffic Dataset, Tools for Self Driven Car.

1.INTRODUCTION

Thousands of people who are handicapped, don't know how to drive the vehicle will help them a lot. It will also help the driver as driving in traffic is difficult so they can just put the vehicle on self driving mode and it will drive for them. The Driver continually needs to screen signals, street wellbeing signs, boundaries, and paths for customary vehicles and settle on choices likewise. Self-driving is not, at this point, a modern dream, yet it's anything but a reality. Organizations announce their commitment to creating and dispatching self-sufficient vehicles and a considerable lot of them talk about the degree of self-rule being created. Unquestionably, a self-driving car can be risky to a few yet it additionally enjoys its benefits. This would bring about reduce traffic congestion, decreased discharges, lower travel costs for all, and a decrease in the expense of new streets and administrations.

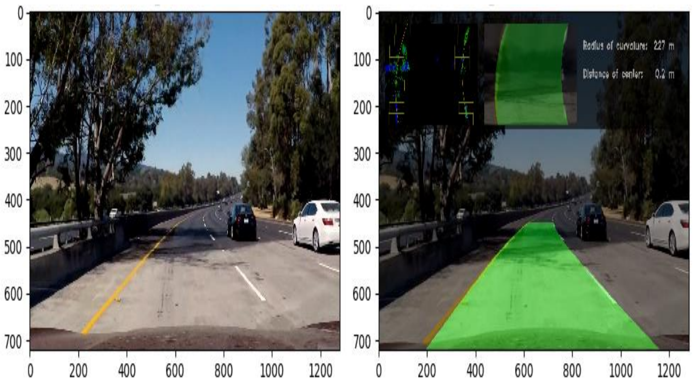
2.Procedure For Building

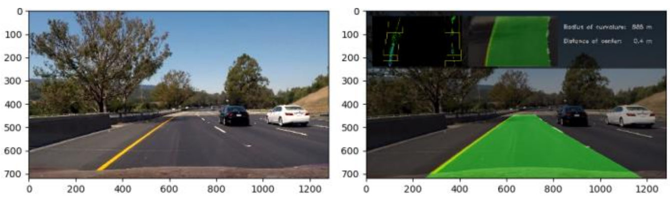
Step :1 - Finding Lane Lines

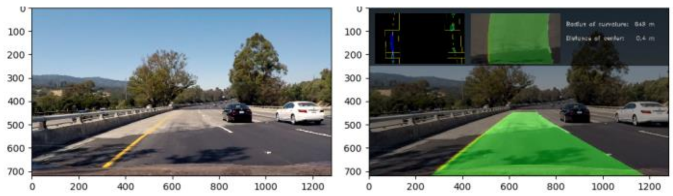
When we drive, we use our eyes to decide where to go. The lines on the road that show us where the lanes are act as our constant reference for where to steer the vehicle. Naturally, one of the first things we would like to do in developing a self-driving car is to automatically detect lane lines using an algorithm.

- Color selection
- Canny edge detection
- Region of interest selection
- Line detection by Hough transformation
- Lines averaging and extrapolation
- Compute the camera calibration matrix and distortion coefficients
- Apply a distortion correction to raw images.
- Apply a perspective transform of a region of interest on an image ("birds-eye view").
- Use color transforms, gradients, etc., to create a thresholded binary image.
- Detect lane pixels and fit to find the lane boundary.
- Determine the curvature of the lane and vehicle position with respect to center.
- Warp the detected lane boundaries back onto the original image.
- Output visual display of the lane boundaries and numerical estimation of lane curvature and vehicle position.

By following this procedure I have detected the lane lines.



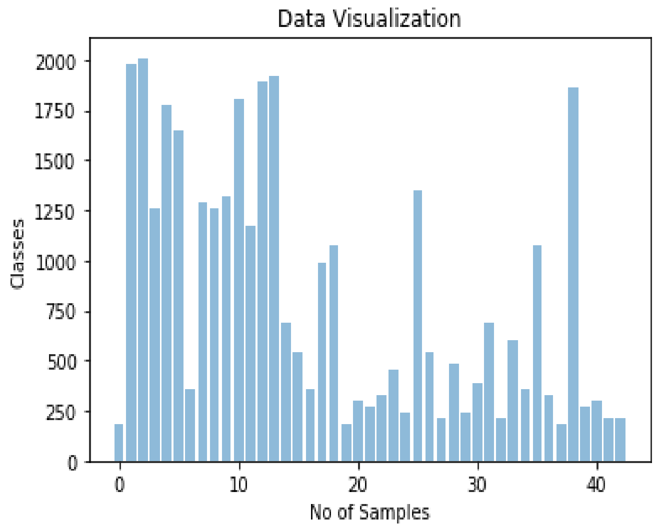




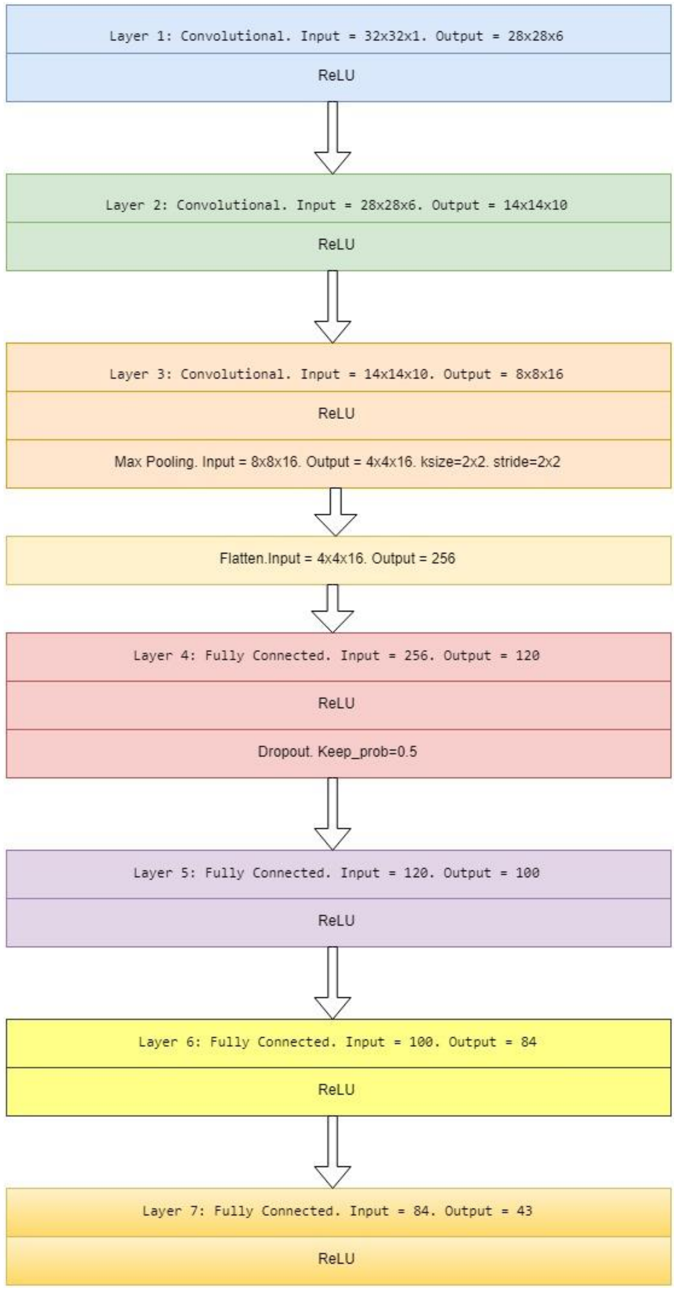
Step -2: Traffic Sign Controller

Build a deep neural network to detect the traffic signs.

1) Load the data set
2) Explore, summarize and visualize the data set
3) Design, train and test a model architecture
4) Use the model to make predictions on new images
5) Analyze the softmax probabilities of the new images



I created my model in this way for identifying the traffic signs:

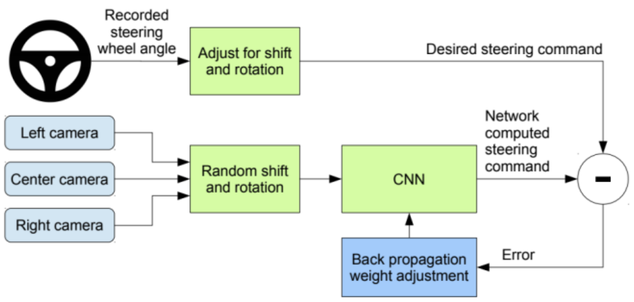


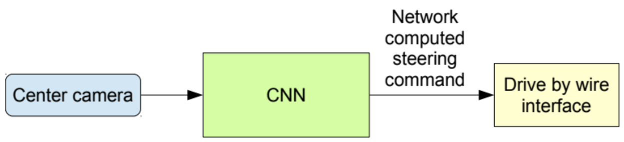
Step - 3: Behaviour Of vehicle

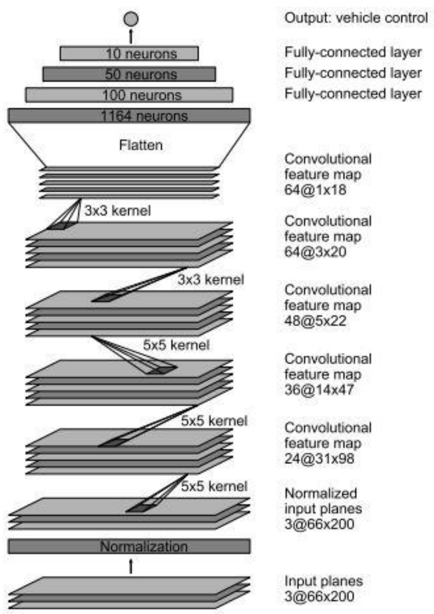
To actually implement on real car first try to implement on simulator and try to modify the code in such a way that it runs on the Lane.

- Use the simulator to collect data of good driving behavior
- Design, train and validate a model that predicts a steering angle from image data
- Use the model to drive the vehicle autonomously around the first track in the simulator. The vehicle should remain on the road for an entire loop around the track.

My summary for building the model was this :







Step -4: Extended Kalman Filter:

I utilize a kalman filter to estimate the state of a moving object of interest with noisy lidar and radar measurements

["sensor_measurement"] => the measurement that the simulator observed (either lidar or radar)

["estimate_x"] <= kalman filter estimated position x

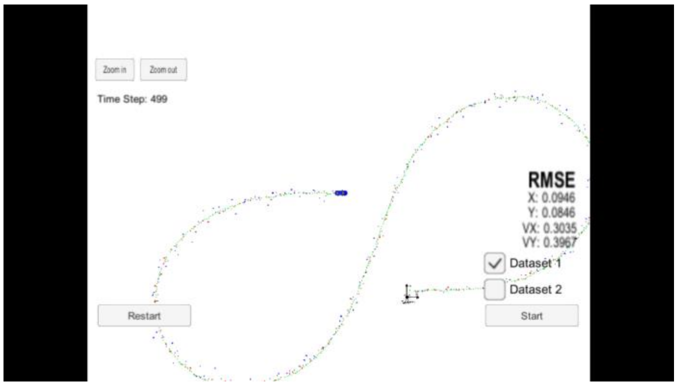
["estimate_ y"] <= kalman filter estimated position y

["rmse_x"]

["rmse_y"]

["rmse_vx"]

["rmse_vy"]



Step -5: PID Controller:

Pid Controller: The main challenge is to drive a car around the track using a Proportional-Integral-Derivative (PID) Controller.

Components of PID-

- P (Proportional Component) - This component tells that the car will steer proportional to cross track error(CTE). CTE is the deviation of car from the reference trajectory(which is exactly in center of track in this project). If your car is to left of the center obviously you will want to steer it back to center by moving right. But in this process the car overshoots when it moves to right and then it tries to move left and so on.. the process continues. This component tells that exactly the steering value is how much proportional to CTE. I estimated the P value to a low value to reduce the direct proportionality.
- D (Differential Component)- This component accounts for the rate of change of CTE. This means if the derivative is quickly changing, the car will move quickly towards center as in case of a curve where steering values are normally higher. If the value of this component is less then the oscillations are too much. To reduce the oscillations I kept the value of this component to be a bit higher
- I (Integral Component)- This component accounts for the sum of all CTE's at a point. If the value of I component is too high, then the car oscillates too much and does not tends to pick up speed. I choose the value of I to be really small.

Information for Tuning Parameters:

Rise time – the time it takes to get from the beginning point to the target point

Overshoot – the amount that is changed too much; the value further than the error

Settling time – the time it takes to settle back down when encountering a change

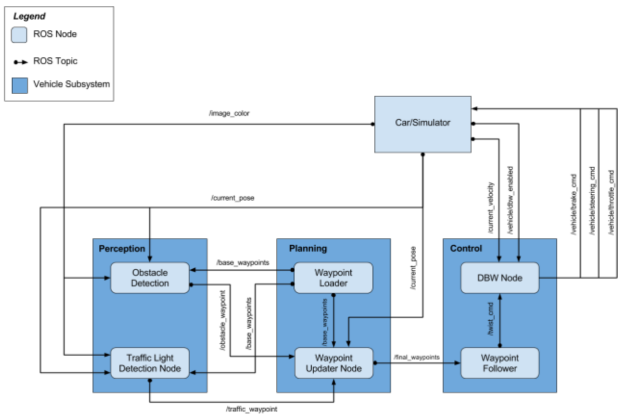
Steady-state error – the error at the equilibrium

Stability – the "smoothness" of the speed

Step -6: System Integration in ROS:

I have done system integration on ROS.

Procedure for the same is:



1) Waypoint Planning:

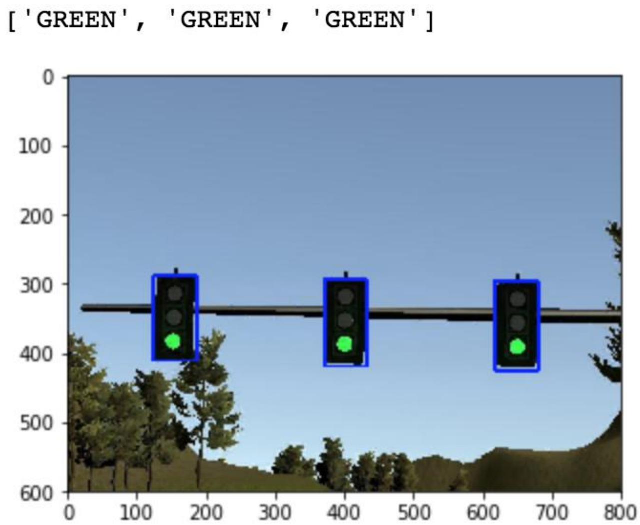
This module charts the desired course for the car and the desired speed. This module is responsible for integrating the information about traffic lights and car state and map to determine the best course.

2) Control:

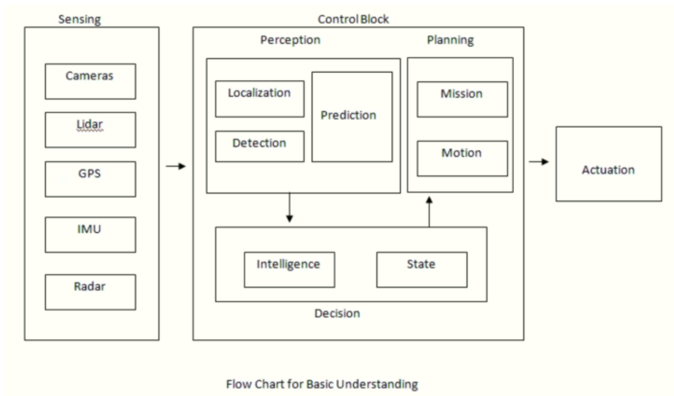
This module was responsible for controlling the car along the desired course. A PID controller was used for the speed while the steering controls were handled by a yaw controller fed by pure_pursuit.cpp.

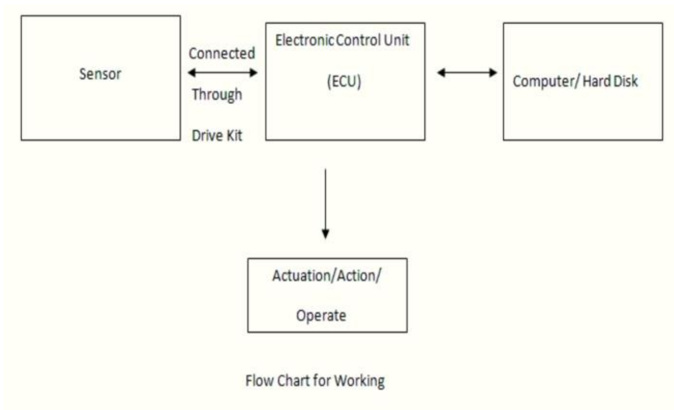
3) Perception(Traffic Light sensing and classification):

The traffic light module: ros/src/tl_detector identifies traffic lights in a camera feed and classifies them as Red, Yellow, or Green. This was accomplished by using a state of the art SSD trained by google and a simple opencv color thresholding script. Due to tensorflow version conflicts, resnet was used on the final version instead of the faster SSD mobilenet. On modest hardware this led to some interesting problems caused by the lag, optimizations are needed.



3.Flowchart

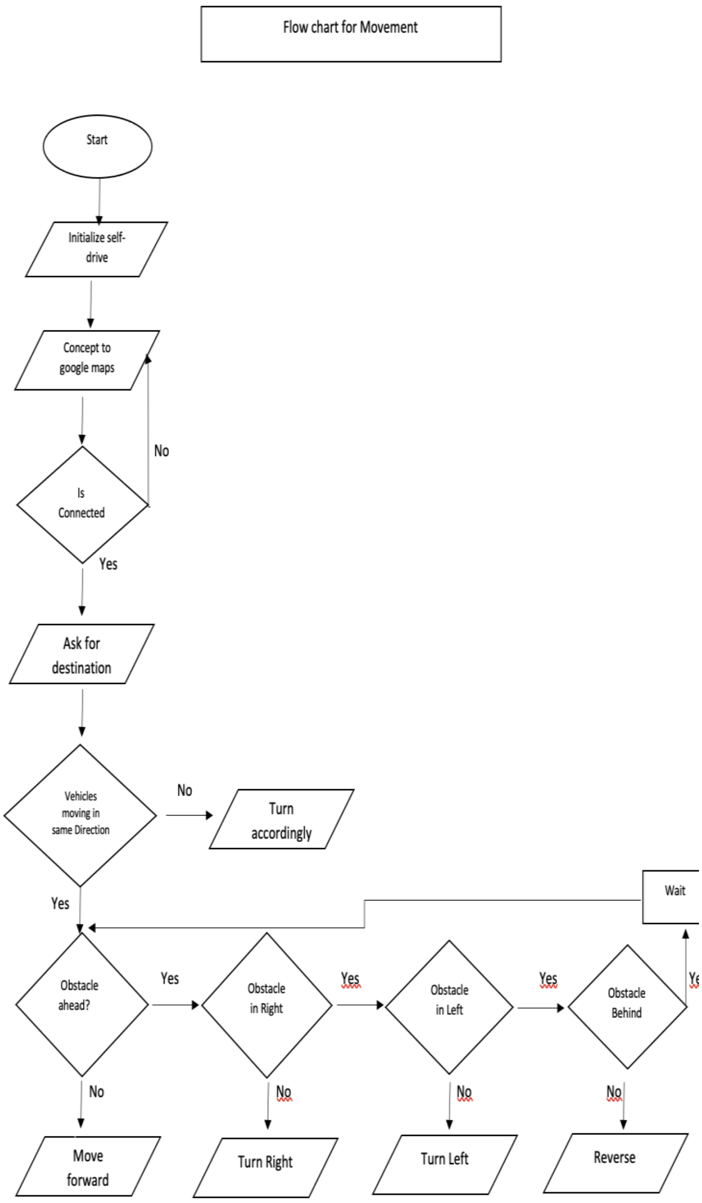




I have implemented on ROS but if you wish to implement in real world for that major tools required are:

1) LiDar
2) Camera

3) Radar
4) Electronic Control Unit
5) Servo Motor
6) Drive kit
7) Nvidia(For Computer)
8) Navigation Sensor
9) Car Dash Software



4. CONCLUSIONS

Development of autonomous car is still in progress and many organisations have come up with brilliant models like Google, Tesla. Despite the fact that everything began from a driverless idea to radiofrequency, cameras, sensors, more semi-self-ruling highlights will come up, along these lines lessening the blockage, expanding wellbeing with quicker responses, and less mistakes. In spite of the inborn advantages, self-governing vehicle innovation should conquer numerous social hindrances

REFERENCES

[1] Truong-Dong Do, Minh-Thien Duong, Quoc-Vu Dang, and My-Ha Le, “Real-Time Self-Driving Car Navigation Using Deep Neural Network” in International Conference on Green Technology and Sustainable Development (GTSD), 2018.

[2] Miao, Xiaodong, Shunming Li, and Huan Shen. "ONBOARD LANE DETECTION SYSTEM FOR INTELLIGENT VEHICLE BASED ON MONOCULAR VISION." International Journal on Smart Sensing & Intelligent Systems 5.4 (2012).

[3] Hajer Omrane, Mohamed Slim Masmoudi, and Mohamed Masmoudi, “Neural controller of an autonomous driving mobile robot by an embedded camera” in International Conference on Advanced Technologies For Signal and Image Processing - ATSIP, 2018.

[4] M. Frutiger and C. Kim, “Digital Autopilot Test Platform with Reference.Design and Implementation of a 2-Axis Autopilot for Small Airplanes,”Department of Electrical and, pp. 1–24, 2003.

[5] M. Weber, “Where to? A History of Autonomous Vehicles,”2014.[Online].Available:http://www.computerhistory.org/atchm/where-to-a-history-of-autonomous-vehicles/

[6] D.Helbing, “Traffic and related self-driven many-particle systems, ”Reviews of Modern Physics, vol. 73, no.4, pp. 1067–1141, dec 2001.

[7] P. Rau, “Target Crash Population of Automated Vehicles,” in In 24th International Technical Conference on the Enhanced Safety of Vehicles.

BIOGRAPHIES

I am currently pursuing B.Tech in Computer Science Engineering with specialization in Artificial intelligence and Machine Learning from Chandigarh University, Mohali, India. Special. I am currently doing an internship in KDOTS Digital Pvt. Ltd.(Fenix Commerce, US) as Product Research and Development Intern (in Artificial Intelligence and Machine Learning)