SUPERVISED LEARNING BASED AUTONOMOUS CAR

B.E (EL) PROJECT REPORT

BATCH 2016-2017

Prepared by

Muhammad Naufil (EL-16045)

Faiz Ur Rehman (EL-059)

Syed Muhammad Haziq (EL-060)

Syed Muhammad Danish Khalid (EL-068)

NED University of
Engineering & Technology
Karachi- 75270

Internal Advisor:

Dr. Yawar Rehman

Assistant Professor

Department of Electronic Engineering

N.E.D University of Engineering & Technology' Karachi-75270

Abstract

This report explains the challenges, design details of supervised learning autonomous cars and also covers the costs and schedule. The goal of the project is to make the prototype for an autonomous car in low cost, low power which can drive itself on custom made road without the interference of humans and detect the object as well avoid it with the help of the camera. For environment learning, supervised learning methodology is used which drives the car on the custom made track and will predict the steering angles in real time.

Dedication

Our Final Year Project is dedicated to our Parents because whatever we are it is because of our Parents ,Family and Teachers .They make us able to face different challenges and win those challenges , last but not least group members, who support us a lot and contribute their full effort to make this project possible.

Acknowledgement

We are very thankful to Almighty God who is the most Gracious, the most Merciful to bestow us with strength and courage to choose such a challenging project. We have put a tremendous amount of energy into this project. However, it would not have been achievable without the support and help of several people.

We are very grateful to our internal advisor of our project Dr.Yawar for his valuable and constructive recommendations during the arranging and advancement of this project. His readiness to present his time so freely is very much appreciated.

Table of contents

Chapter 1 13

Introduction 13

1.1 Project overview 13

1.2 Report outline 15

Chapter 1: Introduction 15

Chapter 2: Hardware equipment 15

Chapter 3: Software and tools 15

Chapter 4: Hardware 15

1.3 Application of system and environment 16

1.4 Background 16

1.5 State of the art 18

1.6 Literature review 19

1.7 Aims and objective 20

Chapter 2 22

Hardware and equipment 22

2.1 Block diagram 22

2.2 Hardware 23

2.2.1 Microprocessor 23

Cuda Cores 23

Jetson Nano 23

Description 24

Pros 24

Cons 24

2.2.2 Makeronics IMX219-160 25

2.2.3 Buck Converter 26

2.2.4 Servo motor 27

CYS S8218 40KG high torque servo motor 28

2.2.5 L298 Motor Driver 29

2.2.6 PCA9685 Motor driver 30

2.2.7 Arduino Board 31

2.2.8 Channel relay module 32

2.2.9 Ultrasonic sensor 33

2.2.10 Voltage regulator 34

Chapter 3 35

Software and tools 35

3.1 Simulation 35

Step 1: Dataset Collection 35

Step 2: Training a model 35

Step 3: Testing 36

Step 1: Isolate the Color of the Lane 37

Step 2: Detecting Edges of Lane Lines 38

i) Noise Reduction 38

ii) Intensity Gradient 39

iii) Suppression of non-maximum 39

iv) Hysteresis thresholding 40

Step 3: Region of Interest 40

Step 4: Line Segments Detection 41

Step 5: Combine Line Segments into Two Lane Lines 43

i) One lane in the image 43

ii) Vertical line 43

Step 6: Planning of motion 44

Step 7: Detected Two Lane Line 44

Step 8: Single detected Lane Line 45

3.3 Track Designing 47

Step 1: Finding the track shape 48

Step 2: Replicating the design into High resolution on Illustrator 48

Step 3: Rooftop with crayons 49

Step 5: Panaflex printing 50

3.3.1 Track adjustment 51

3.4 Object detection 52

3.4.1 TensorRT on Nano 52

3.4.2 Pre-trained models 53

3.4.3 Flowchart for object detection 54

Chapter 4 55

Hardware 55

4.1 Design concept 55

4.2 Data collecting process 55

4.3 Supervised Learning Based Autonomous Car Hardware Block Diagram 56

Chapter 5 59

Results and discussions 59

5.1 Lane Detection 59

Picture 59

Video 60

On Track 60

5.2 Data set 60

5.3 Training of the data sets 61

5.3.1 Angle distribution 61

5.3.2 Train and valid data 62

5.4 Image augmentation 63

5.4.1 Zooming the frame 64

5.4.2 Pan the frame 64

5.4.3 Brightness adjustment 65

5.4.4 Flipping the frame 65

5.5 Preprocess Training data for Nvidia model 67

5.6 Trained model loss 67

5.7 Object detection 68

Chapter 6 70

Conclusion 70

6.1 General improvements 70

6.2 Future recommendations 70

6.3 Difficulties Faced 71

References 72

Table of figures

Figure 1: Block diagram 22

Figure 2: jetson nano 23

Figure 3: Nvidia Jetson Nano performance benchmark 24

Figure 4: 6 volts battery 25

Figure 5: Makeronics IMX219-160 26

Figure 6: Buck converter 26

Figure 7: CYS S8218 40KG high torque servo motor 28

Figure 8: L298 Motor Driver 29

Figure 9: PCA9685 Motor driver 30

Figure 10: Arduino board 31

Figure 11: 5 volts single channel relay 32

Figure 12: Ultrasonic sensor 33

Figure 13: 3.3BT voltage regulator 34

Figure 14: Nvidia model 35

Figure 15: Steering angle of autonomous car 36

Figure 16: Lane detection block diagram 36

Figure 17: BGR into HSV 38

Figure 18: Suppression of non-maximum 39

Figure 19: Hysteresis thresholding 40

Figure 20: Cropped edges of the frame 41

Figure 21: Line segments 42

Figure 22: Two lane lines 44

Figure 23: Single detected lane 45

Figure 24: Angle predictions 46

Figure 25: AWS Deep Racer 47

Figure 26: Illustrator design of the track 48

Figure 27: Track designing on roof 49

Figure 28: Photoshop track designing 49

Figure 29: Panaflex printed 50

Figure 30: Track adjustment 51

Figure 31: Selection of models 53

Figure 32: Flowchart for the decision making 54

Figure 33: Detection of stop sign and person 54

Figure 34: Block diagram for data collection 56

Figure 35: Block diagram of supervised learning autonomous car hardware 57

Figure 36: Lane test on picture 59

Figure 37: Lane detection test on real time 59

Figure 38: Lane detection on video 60

Figure 39: Lane detection test on track 60

Figure 40: Data set collecting 61

Figure 41: Angle distribution graph 62

Figure 42: Train and valid data graph 62

Figure 43: Frame zoomed 64

Figure 44: Pan the frame 64

Figure 45: Brightness adjustment 65

Figure 46: Frame flip 65

Figure 47: Image data augmentation techniques 66

Figure 48: Lane detection using Nvidia Model. 67

Figure 49: Loss of trained model 67

Figure 50: Accuracy of the model 68

Figure 51: Camera detecting the object and person 68

Figure 52: Object detection 69

Chapter 1

Introduction

1.1 Project overview

For the past few years, autonomous vehicles have become the hot research topic. This project is the requirement of the automotive industry. The autonomous car can aid to reduce road accidents due to human error.

The study shows that driver in US spend an average of more than 17,600 minutes behind the wheel each year. All things considered, the greater part of drivers (52.1%) report having driven while chatting on a hand-held cell phone in any event once in the previous 30 days. Fewer respondents report taking part in occupied driving by perusing (41.3%) or composing content/email (32.1%) on a hand-held cell phone while driving. Despite of big rates of apparent risk and individual/social dissatisfaction with respect to tired driving, about 27% of drivers confess to having driven while being drained to the point that they had a hard time keeping their eyes open ,in any event once in the previous 30 days. Most drivers (95.1%) see driving in the wake of drinking as very or incredibly hazardous. However, almost 11% admitted to having done so in the past 30 days [1]. Uber are willing to use these self-driving cars [2], where it nullifies the iron triangle of health care, because autonomous vehicles will lead to cost reduction in fair rates, while also having maximum quality. It will also provide maximum access because it will provide users protection from peak factors, which drivers may exploit. The research has made to solve all these problems, but they couldn't succeed in it, but the emerging technologies like autonomous vehicles have the potential to do so by driving on the road in a more efficient way which not only cause less greenhouse gas but also could save a precious life of our loved one [3].However, the unpredictability around self-driving is gigantic, each issue is expansive and many-sided and its interdependence with others can't be disregarded. Researchers have attempted to separate the overall significance of each issue and evaluate the effects of increasing rates of autonomous cars [4, 5, and 6] .Autonomous vehicles are programmed to be more economical. Human drivers accelerate and brake more than necessary, which uses excessive fuel. In contrast, self-driving trucks and cars are programmed to operate with maximum efficiency at all times [7].

On the other hand, the introduction of autonomous car will bring the benefit of traffic decreased with itself because all the car when connected to each other and surroundings will help to find the best suited route this can cause less traffic jam it would be beneficial for people's health as traffic jam could rise the blood pressure, anxiety, depression [8]. Equity is another major thought. Self-driving technology will facilitate mobilizing people who are unable to drive themselves, like the senior citizen or disabled persons [9]. Once autonomous cars are fully integrated into our transportation system, experts believe the economic benefits to society will be enormous. Road accidents in Australia are estimated at $27 billion. Even a 5% reduction in the accident rate could save the country millions of dollars every year [10].For a considerable length of time, vehicle and innovation companies have been discussing autonomous cars. Autonomous cars will change the cities forever [11].

Promises life changing safety and simplicity have been held tight on these vehicles. Now a portion of these guarantees are starting to happen as expected, as vehicles with an ever increasing number of independent highlights hit the market every year. An autonomous car is more reliable than a human driver because it may aid to reduce traffic accidents on roads and make a safer environment.

The goal of the project is to make the prototype for an autonomous car in low cost, low power which can drive itself on custom made road without the interference of humans and detect the object as well avoid it with the help of the camera. The remote control car will be driven manually to generate a dataset on a customized track. Frames given to the neural network will predict the car dynamics in real time.

1.2 Report outline

The project work is divided into six parts:

Chapter 1: <u>Introduction</u>

This chapter includes a conversation about the projects' past, present and future and its main goals and difficulties that we have confronted and our project dealt with.

Chapter 2: <u>Hardware equipment</u>

This chapter includes a conversation about the hardware components used for the project

Chapter 3: <u>Software and tools</u>

In this chapter, we have talk about the software used for both lane detection and object detection for the autonomous car

Chapter 4: <u>Hardware</u>

In this chapter we have discussed about the design concept for the car and it also includes detail information about the hardware connections.

Chapter 5: <u>Results and discussions</u>

In this chapter we have discussed about the results of lane and object detection.

Chapter 6: <u>Conclusion</u>

In this chapter we have discussed about the future work and difficulties faced during the project.

1.3 Application of system and environment

To train the car in the real conditions like streets and highway is quite risky and very costly .keeping in mind that the project is based on low cost and low power autonomous car, Jetson Nano is used for the autonomous car and raspberry pi camera with the custom made track so that the car can be trained on the customized track without any fear .The environment in which the car is trained consist of complex road track including sharp turns with traffic signs and traffic lights which all are included in helping the car to make its own decision based on supervised learning. When the autonomous car is prepared in this environment, it can without much of a stretch be scaled to some other greater environment with a costly vehicle.

1.4 Background

History shows that the research on autonomous is not a new thing. It was started back in 1926 when the first step was taken towards the autonomous and to introduce a radio controlled car and it was the achievement of Houdina Radio Control in New York City. It consists of a transmitting antenna in its rear compartment which catches the radio impulse signal, and then sends information to its electric motors which direct the car's movement towards the destination and it is one of the most primitive steps towards the autonomous. Later on RCA labs proposed a bit modified form of an autonomous car and it was guided by the wires laid on the floor and after that this idea was experimented to a greater level by installing it on an actual highway [12]. During the 1980's timid-1990's research was done at Bundes wehr University Munich it was done by Ernst Dickmanns and his team they worked on building up vision-based system to implement in Mercedes S-Class and by implementing that it can detect the lane marking and it was a successful project the car travelled about 1700 km and 95% distance covered autonomously and it can also reach up to a speed of about 112 mph. And later on several laboratories and universities started working on autonomous vehicles the LIDAR, neural network and computer vision techniques started using to control the vehicles [13]. Coming to 2000's the US started to invest in this new advance technology, which is unmanned vehicles while the DARPA are asked to accelerate their development of such car which can detect the and avoid the object and can drive at a reasonable speed and later on the competition held by DARPA in 2004 with a winning prize of 1 million in which several contestants took apart

The goal of the race was to navigate through the desert up to 150 miles in California. Unfortunately, there was no winner as nobody was able to navigate the farthest was 8 miles. And in 2005 there were the five winners of the competition. The first place was taken by a vehicle designed by Stanford University and Volkswagen, using technology adapted from the Stanford Cart and it was named Stanley. During the last few years the research is conducted not only by universities, but the giant manufacturers like BMW, TESLA, and Volkswagen are also testing their own autonomous cars. Among the leaders, one of them is the Google Self-Driving Car project, which was initiated in 2009 it used the technologies like lidar, GPS, maps to navigate through its path. In 2015 the Tesla introduced its 'autopilot' mode in a Model S vehicle which can switch lanes and park on command [14]. The keynote is that unlike using LIDAR in Google Self Driving cars, Tesla is solely relying on cameras. Nvidia also spends a large amount on deep learning like they develop a convolution neural network which can predict the steering angle and also train the car to drive in an urban environment using only the pictures of a single camera. It can also operate in such a place where the vision is not clear like a parking lot, sometimes lane marking is not proper but the result will always be better because the internal component will optimize itself to maximize its performance [15]. As of now, various prominent organizations are leading on-road tests like BMW, who plan to have a completely self-driving vehicle underway by 2021, plans to double their checking fleet to around 80 autonomous vehicles and have pledged to hold out a hundred and fifty five million test miles[16].The research is still going on and it has been predicted that by the end of 2020 many automobile manufacturers will release their fully or semi-autonomous vehicles and by the end of 2035 most of the vehicles will be autonomous .

1.5 State of the art

Autonomous vehicles are the future of the world. Most of the vehicles will be autonomous in the near future the research on autonomous is has already started long back and still going on like recently in 2017 a research has been made related to the safety of autonomous car the idea was to implement the safety module in the car that can interact with both systems of the car and create the protection layer and it is independent of the way the vehicle is designed so it can be tested individually. It can benefit when the manufacturer introduces their fully autonomous car in near future [17].

The paper in 2015 analyzes how the autonomous technology develops from 1920's when we use to have a radio controlled guided vehicle and now in present we are using advanced technology such as lidar, GPS, computer vision base guided vehicle and this paper also predict what will be the future of autonomous vehicles in the world and what will be the challenges will have to face while introducing this .The paper in 2014 analyze that advanced intelligent car technology can bring important social and economical benefits.[18].

Other studies has been made in 2018 the goal was to build an autonomous car using the Deep Reinforcement Learning applied with the Q network to drive the car through the urban environment their approach was to take input from the camera sensor, laser sensor implemented in front of the car and the was also cost efficient and capable of driving at a high speed [19].This paper focuses on the implementation of the autonomous car using raspberry pi. It uses the same neural network of Nvidia's Dave 2 car. The main focus of this paper is to check if the microcontroller like raspberry will be able to handle such a heavy neural network. They check several parameters like effect of core count to inference the timing, effect of Co-scheduling Multiple CNN models, effect of memory bandwidth. So these were the parameter they analyze and yes they successfully implement the CNN Dave 2 models on a raspberry pi [20].So like these researchers there are many more researches have been done and are still going on to build up an autonomous car which is cost efficient, fuel efficiency, high reliability and considering the safety aspects and I'm sure we'll see autonomous vehicles on road in the near future.

1.6 Literature review

S.No	Research Paper	Author	Findings
1	Predict Steering Angles in Self-Driving Cars	Neha Yadav, Rishi Mody, University of Massachusetts Amherst Amherst MA	Data augmentation: 1. Shadowed images 2. Brightness 3. horizontal shifted images

2	Self-Driving Car Steering Angle Prediction Based on Image Recognition	Shuyang Du, Haoli Gao, Andrew Simpson	Different Architectures: 1. 3D Convolutional Model with Residual Connections and Recurrent LSTM Layers 2. New Architecture 3. Transfer Learning
3	You Only Look Once: Unified, Real-Time Object Detection	Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, University of Washington, Allen Institute of AI, Facebook AI Research	Comparison between Real time detectors

1.7 Aims and objective

- Less human interaction.
- Environment friendly
- Minimizing of human error
- To guarantee the planned work is legitimate and not defective.
- Ideal working of software and hardware with fast transmission among the system components.
- Human safety.

- Extremely useful in long routes.
- Very useful for the elders, youth and handicapped people and it also brings relaxation to the driver so that he/she can perform his/her task, send messages or emails at the same time.
- To include the progression with respect to the time

Chapter 2

Hardware and equipment

In this chapter all the hardware components and equipment are defined with all their key features and functionalities. To make the remote control car autonomous in environment supervised learning techniques are used .There could be alternative ways to learn the environment by line tracking or model based learning. Supervised learning is used for the learning of remote control cars to make it autonomous which is explored in detail.

2.1 Block diagram

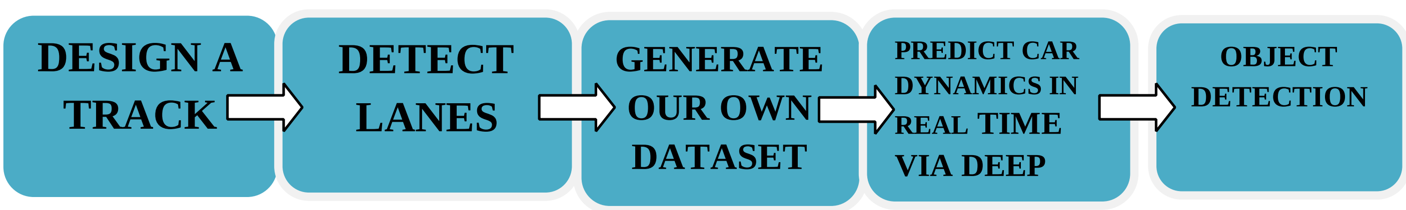

Figure 1: Block diagram

2.1.1 Supervised learning

In supervised learning, input and output variables are given to the system with the goals of learning on how they are related with each other [21]. The objective is to generate precise enough mapping functions. So, that when new data is given, the algorithm can guess the output .Supervised learning is an iterative method, every time the algorithm makes a new prediction, it is rectified or given feedback till it reaches its tolerable level of performance. In this project CNN (Convolutional Neural Network), a deep learning method for supervised learning technique is used.

2.2 Hardware

To build an autonomous car following equipment were used:

2.2.1 <u>Microprocessor</u>

There are a variety of microprocessors to choose from and a design architect should carefully choose in terms of speed and price to satisfy the project need. The variety was:

1- Raspberry Pi with any Deep learning accelerator
2- Jetson Nano

<u>Cuda Cores</u>

Cuda cores allow multiple processing, very handy for Neural Network powered applications.

<u>Jetson Nano</u>

Figure 2: jetson nano

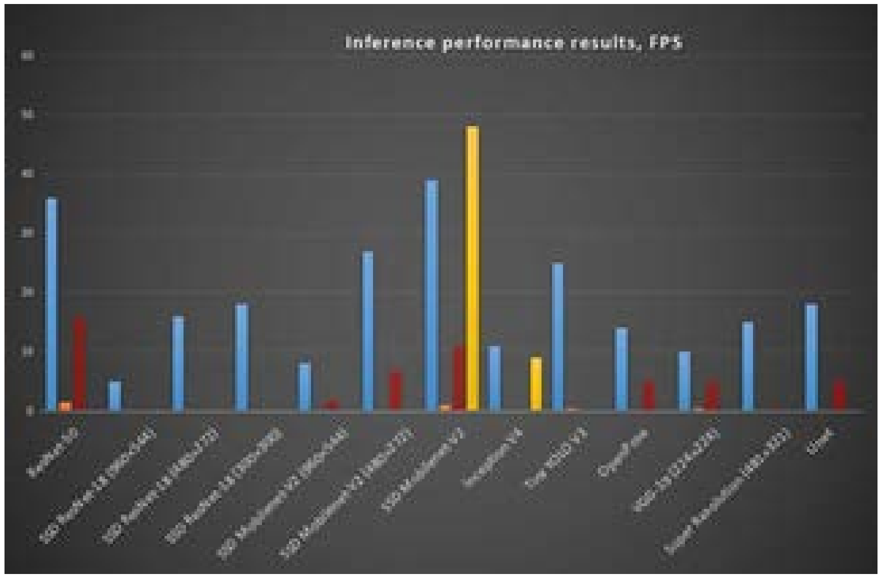

Figure 3: Nvidia Jetson Nano performance benchmark [22]

The blue bar represents Jetson Nano, orange bar represents Raspberry pi 3 while Raspberry pi 3+ Intel Neural Compute Stick 2 represents in red bar .The benchmark for the frame per seconds clearly shows that Jetson Nano has a clear edge over the other two

Description

Nvidia Jetson Nano is an embedded system-on-module (SOM) and developer kit from the NVIDIA Jetson family, a coordinated 128-center Maxwell GPU, CPU of quad core ARM 64 bits, 4GB LPDDR4 memory, in conjunction with support for MIPI CSI-2 and PCIe Gen2 high-speed I/O.

Pros

Jetson Nano is more powerful than raspberry pi, because it has built in cuda 128 cuda cores and can record 4K video in 60FPs and has 4 GB RAM.

Cons

The downsides are it doesn’t have a built-in wifi and Bluetooth connectivity.

powering up the jetson nano is a big challenge as jetson nano is fragile and need exactly 5v 4 amp power through barrel jack so that all of its functions and GPU work on its best performance so that is why we use four 6V 4.5 amp dry batteries in series accompanied with buck converter which convert 6V 4.5amp input voltage and current into exactly 5V and 4 amp and gave us about two and a half hour of backup with its full charge. Two of the motors are used in powering up jetson nano and rest of the two motors are used for dc motors.

Figure 4: 6 volts battery

2.2.2 <u>Makeronics IMX219-160</u>

The previous camera which was raspberry pi cam v2 had an angle of view not more than 62 degree, which was causing problems for the project because whenever the curve or turn came by the camera was not being able to captured both the lanes .keeping this reason, bought a new wide angle camera whose angle of view was more than 160 degree.

The new wide angle camera is Makeronics IMX219-160 Camera Module for Jetson Nano which has a resolution of 3280 × 2464 and dimension of 25mm x24mm x18mm and has an 8 megapixels, the sensor used is Sony IMX219.The specification of lens is that it has a CMOS size of 1/4inch, aperture (F) of 2.8 and focal length of 2.4mm and angle of view of 160 degree.

Figure 5: Makeronics IMX219-160

2.2.3 Buck Converter

The Buck Converter is utilized where the dc output voltage should be lower than the input voltage. It is helpful where electrical imprisonment isn't needed between the trading circuit and the yield, however where the information is from a revised source of AC, confinement between the source of AC and the rectifier likely could be given by a mains limiting transformer [23].

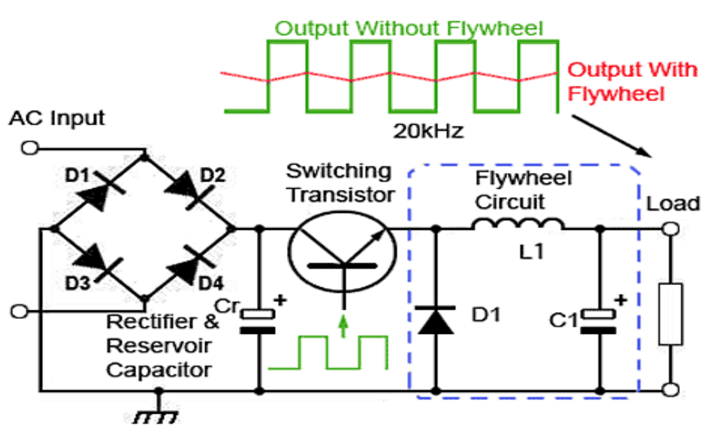

Figure 6: Buck converter

2.2.4 Servo motor

A servo motor is a revolving actuator or a motor that takes into account an exact control regarding the angular position, increasing speed, and speed. Essentially it has certain capacities that a standard motor doesn't have. The servo motors typically used for high development contraptions in mechanical applications like computerization advancement. It is an autonomous electrical contraption that turns bits of a machine with high capability and extraordinary accuracy. Basically, a servo motor is a closed loop mechanism, utilizing position contribution to control its movement and position. Addressing the position coordinates for the yield shaft .The motor unites some kind of encoder to give position and speed analysis. In the clearest case, we measure only the position. By then the intentional circumstance of the yield is differentiated and the request position, the external commitment to the regulator. Now, if the output position varies from that of the expected output, an error signal produces. Which at that point makes the motor turn in either bearing, according to need to carry the yield shaft to the proper position. As the position draws near, the error signal decreases to zero. At long last the motor stops [24].

Motor controls speed and position decisively. Presently a potentiometer can detect the position of the mechanical of the shaft. The motor shaft is coupled with it through gears. The shaft current position is changed into an electrical signal by potentiometer and equated with the input signal command .If the feedback signal is different from the given input, an error signal alerts the user. The error is intensified and applied as the input to the motor, thus the servo pivots. When the required position of the shaft is reached, it becomes null and subsequently the servo stays halt while the position is being holded. The input is in the form of an electrical pulse. The real input to the motor is the distinction between feedback signal (current position) and required signal, henceforth speed of the servo is corresponding to the contrast between the current position and required position.

CYS S8218 40KG high torque servo motor

The CYS-S8218 servo motor has an operating voltage of 6.0 ~7.0 volts and operating temperature range of (-) 10 to (+) 50 degree C .The operating angle of this motor uses 45 degree ,one side pulse travelling. The current drain (6.0v) uses 20mA/idle and 180 mA no load operating and current drain(7.0v) uses 20 mA/idle and 200mA no load operating and the servo motor has a stall current of 7.5A/8.6A.The direction of servo motor is anticlockwise/pulse travelling 1000~2000 usec. The size of servo motor 59.5*29.2*55.2mm.The motor type is NdFeb motor and bearing type dual ball bearing [25].

Figure 7: CYS S8218 40KG high torque servo motor

2.2.5 L298 Motor Driver

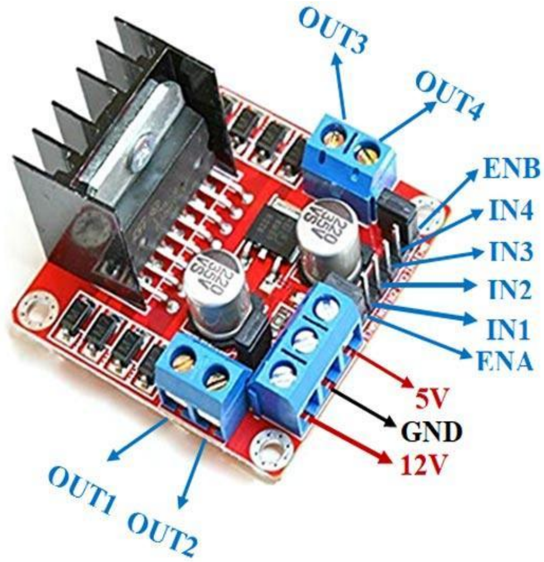

Figure 8: L298 Motor Driver

Motor driver L298 is an effective motor driving force module for DC and Stepper Motors. This module contains a L298 engine driving force IC and a 78M05 5V regulator.L298N Module can manage as much as four DC cars, or 2 DC cars with directional and control of speed.

Features & Specifications

- Module of driver : L298N 2A
- Chip of driver: Double Bridge L298N.
- Maximum supply voltage: 46 Volts.
- Maximum supply current: 2 Ampere.
- Logic voltage: 5 Volts.
- Driver voltage: 5-35Volts.
- Driver current: 2 Ampere.

- Logical current: 0-36 m Ampere.
- Max power: 25 Watts.
- Current sense for all motor

2.2.6 <u>PCA9685 Motor driver</u>

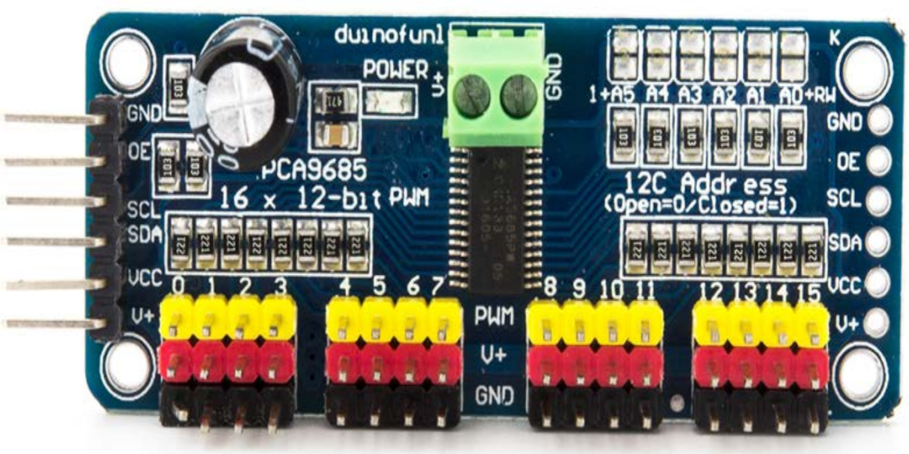

Figure 9: PCA9685 Motor driver

The PCA9685 is a 16-channel I2C-transport controlled LED controller smoothed out for (RGBA) shading backdrop illumination applications. Each LED yield has singular 12 bit resolution PWM regulator with a fixed recurrence. The regulator works at a programmable recurrence from an average 24 Hz to 1526 Hz with an obligation cycle that is flexible from 0% to 100% so the LED can be set to yield a particular brilliance. All yields are set to the equivalent PWM frequency. With the PCA9685 as the master chip, the 16-channel 12-bit PWM Servo Driver only needs 2 pins to control 16 servos, thus greatly reducing the occupant I/Os. Moreover, it can be connected to 62 driver boards at most in a cascade way, which means it will be able to control 992 servos in total.

<u>Feature</u>

Contains an I2C-controlled PWM driver with a built-in clock. It means, unlike the TLC5940 family, you don’t need to send it signals tying up your microcontroller; it is completely free running!

- We can control it from a 3.3V microcontroller and still drive up to output of 6 volts which is great to control LEDs with a 3.4V+ forward voltage
- Supports utilizing two pins to control 16 free-running PWM yields, even anchor up 62 breakouts to control up to 992 PWM yields.
- 3 pin connectors in 4 groups, so you can plug in 16 servos at one time (Servo plugs are slightly wider than 0.1″ so you can only stack 4 adjacent ones on 0.1″-hole female headers.

2.2.7 <u>Arduino Board</u>

Arduino is a microcontroller board with 8 bit ATmega328P microcontroller .Alongside ATmega328P, it comprises different segments, for example, voltage controller, stone oscillator,

Figure 10: Arduino board

sequential correspondence, and so forth to help the microcontroller. Arduino has 14 advanced information/yield pins (out of which 6 can be utilized as PWM yields), 6 simple information sticks, a USB association, A Power barrel jack, an ICSP header and a reset button. Arduino can be utilized to communicate with another Arduino board or different microcontrollers. An ATmega16U2 on the board channels this sequential correspondence over USB and shows up as a virtual com port to programming on the PC. The ATmega16U2 firmware utilizes the standard USB COM drivers, and no outside driver is required. The Arduino programming incorporates a sequential screen which permits straightforward literary information to be sent to and from the Arduino board. There are two RX and TX LEDs on the arduino board which will streak when information is being sent by means of the USB-to-sequential chip and USB association with the PC (not for sequential correspondence on pins 0 and 1). A Software Serial library takes into account sequential correspondence on any of the Uno's advanced pins.

2.2.8 Channel relay module

Figure 11: 5 volts single channel relay

The Single Channel Relay Module is an advantageous board which can be utilized to control high voltage for example, motors, solenoid valves and AC load. It is intended to be interfaced with microcontrollers, for example, PIC, Arduino and so forth. The terminal of relay (COM, NO and NC) is being carried out with a screw terminal. It additionally accompanies a LED to show the relay status**.** Any microcontroller of 5 volts is compatible such as arduino. It has a maximum current switching of 10A and voltage of 250VAC/30VDC.

2.2.9 Ultrasonic sensor

It is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground individually. This sensor is an exceptionally well known sensor utilized in numerous applications where estimating separation or detecting objects are required. The Ultrasonic transmitter communicates an ultrasonic

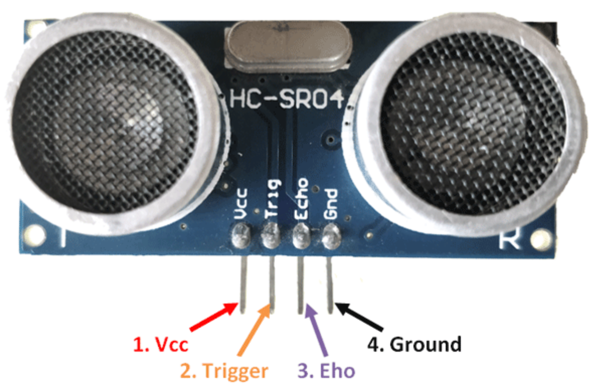

Figure 12: Ultrasonic sensor

wave, this wave goes in air and when it gets protested by any material it gets reflected back toward the sensor this reflected wave is seen by the Ultrasonic beneficiary module. It is used to dodge and identify obstructions with robots like biped robot, impediment avoider robot, way discovering robot and so forth .It is used to quantify the distance within a wide scope or range of 2cm to 400cm.The vcc pin is used to power the sensor with 5 volts. The trigger pin is used as an input pin, echo pin is used as the output pin and ground pin is connected to the ground of the system.

2.2.10 Voltage regulator

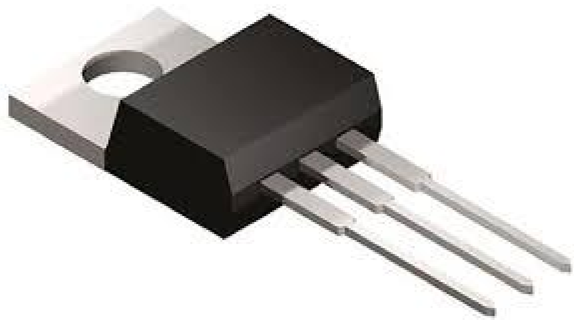

Figure 13: 3.3BT voltage regulator

A voltage regulator creates a fixed yield voltage of a preset extent that remaining parts steady paying little heed to changes to its input voltage or burden conditions, in project voltage regulator 3.3bt is used .It has an output voltage of 3.3 volts and the current of 750mA.The operating temperature is between -40 degree ~125 degree and it belongs to the linear voltage regulator type.

Chapter 3

Software and tools

This part of the chapter briefly describes the different software required to be used in the project:

3.1 Simulation

Udacity simulator is used. Udacity.com designed a self-driving car simulator for its Nano degree and made it publicly available free to use. There is a car and a track made on Unity simulator.

Step 1: Dataset Collection

There is a car with 3 cameras on each side i.e front right and left. Capturing 10FPS and recording their corresponding steering angles.

Step 2: Training a model

Using NVIDIA's model architecture, we trained the neural network on a previously collected data set [26].

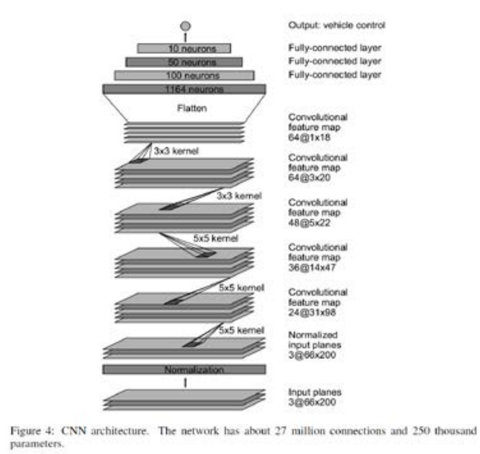

Figure 14: Nvidia model

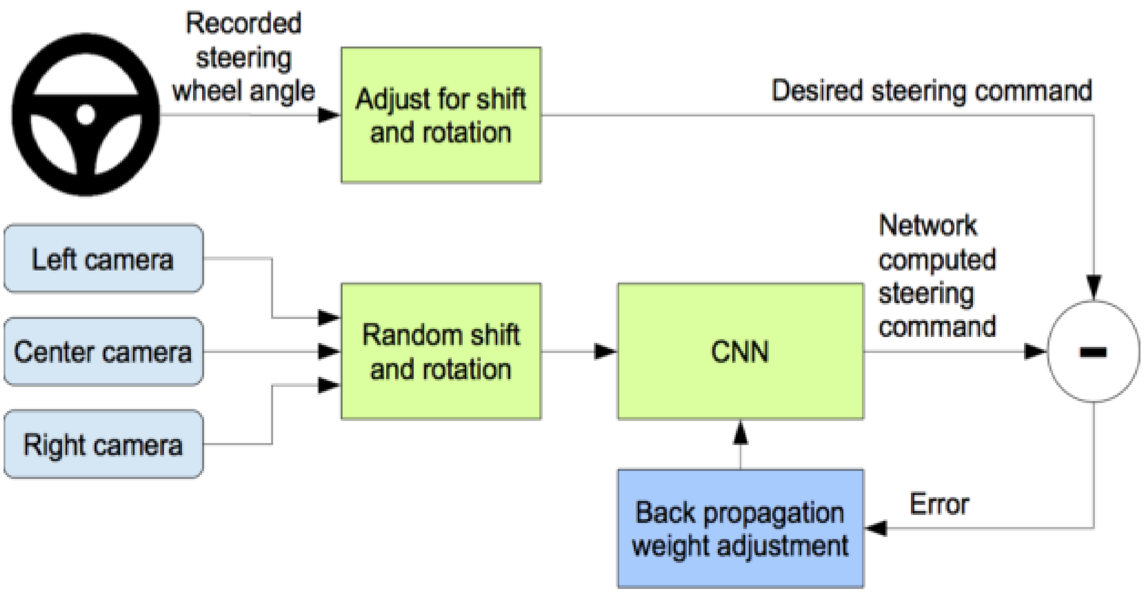

Figure 15: Steering angle of autonomous car

Step 3: <u>Testing</u>

Let the car drive on its own and see how good it is doing.

3.2 Lane detection

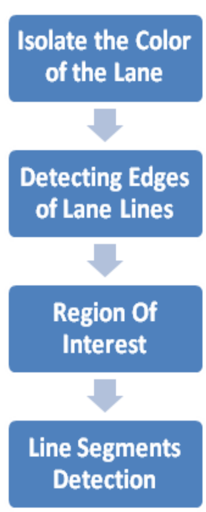

Figure 16: Lane detection block diagram

In contrast to simulation, in real life frames don't come with labeled steering angles. There are two ways to do it:

1. Using a sensor to measure the steering angle
2. Use Camera to detect lanes

The former approach is chosen to avoid extra cost of using a sensor.

A lane system has two components one of them is lane detection and the other is steering angle. The role of the lane detection is to transform the videotape of the custom made track into the directions or co-ordinates of the recognized line of lanes. OpenCV is used as the technique for the recognition of the lanes but before applying it on the videos, open cv technique is applied on the still images of the road to understand the technique of open cv on how to detect the lanes .Once it is done, it is quite similar to apply it on the video which is just rehashing the equivalent or same procedure for frames in a video. Several steps involved in the detection of lane lines.

Step 1: <u>Isolate the Color of the Lane</u>

The first step is to detect the yellow lane in a single frame. We detect all the yellow areas in this picture. To do this we transform this image from BGR (Blue/Green/Red) to HSV (HUE/SATURATION/VALUE). The main purpose of this to convert this picture into hsv is that yellow colour will have different shades in different lightning condition the hsv will render the entire yellow lane into same shade of colour and once we are done with this transformation the next step is to detect all the yellow colour by selecting the range of colour and then applying mask over it.

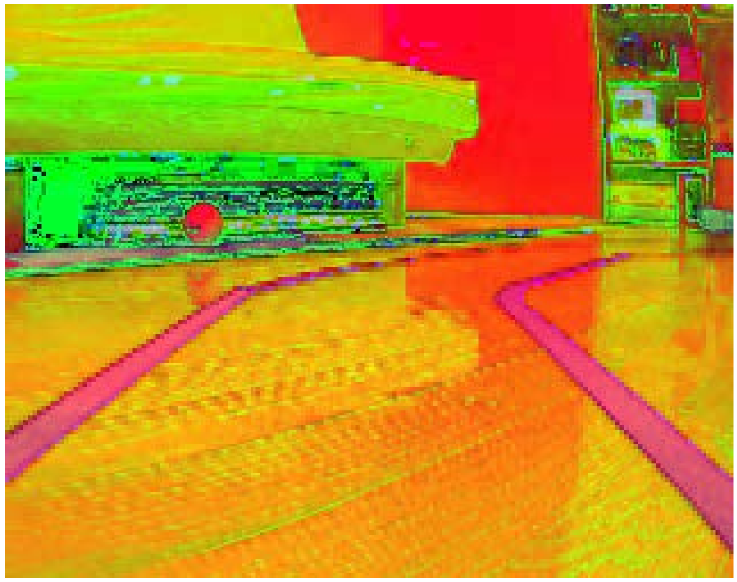

Figure 17: BGR into HSV

Step 2: <u>Detecting Edges of Lane Lines</u>

The second step is to detect the edges from the mask image so that we can have some idea about the lanes to detect the edges .We used canny detection of canny function which is a robust function that can detect the edges from frame. Basically the goal of this function is to detect the sharp edges like changing from black to white by giving them some set of threshold values from which it determines the edges the canny algorithm has four main parts.

i) <u>Noise Reduction</u>

A filter of Gaussian is applied to smooth the frame to bring down the detector's sensitivity to noise. This is finished by utilizing a kernel as per an image of ordinarily appropriated numbers and afterward applying an over picture this will set the estimation of every pixel to the normal of its neighboring pixel.

ii) Intensity Gradient

After that the sobel kernel is applied over the smooth image with the coordinates of x and y to find out the detected edges are vertical or horizontal.

iii) Suppression of non-maximum

It is utilized for the "thin" and edge sharpening. The value is checked for each pixel if it is near most in the path of the gradient calculated previously.

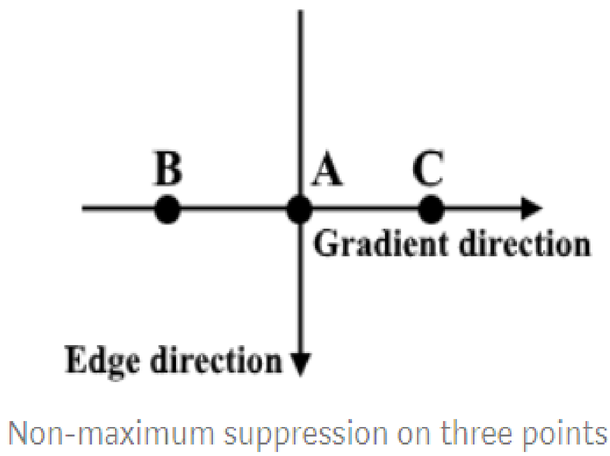

Figure 18: Suppression of non-maximum

A is on the edge with a vertical bearing. As inclination is customary to the edge heading, pixel estimations of B and C are conversely with pixel estimations of A to choose if A will be a nearby most extreme. In the event that A is neighborhood greatest, non-most extreme concealment is analyzed for the ensuing point. Or the consequences will be severe, the estimation of pixel A is set to zero and it is smothered.

iv) Hysteresis thresholding

Solid pixels are tested to be in the last guide of edges. However, dim pixels are likewise examined to choose whether or not it comprises an edge or clamor. Applying two pre-characterized minVal and maxVal threshold values, we set that any pixel with depth gradient greater than maxVal are edges and any pixel with depth gradient lower than minVal are no longer edges and disposed. And those pixels which have the value in between the minimum and maximum value will only consider edges if only they are attached to the pixel having the value exceeded above maximum threshold. Canny edge detection functions take three parameters. The first is the name of the image and the second and third is the upper and lower range which is around 200 to 400 recommended by opencv. By applying this technique on our image we get the edge detected image.

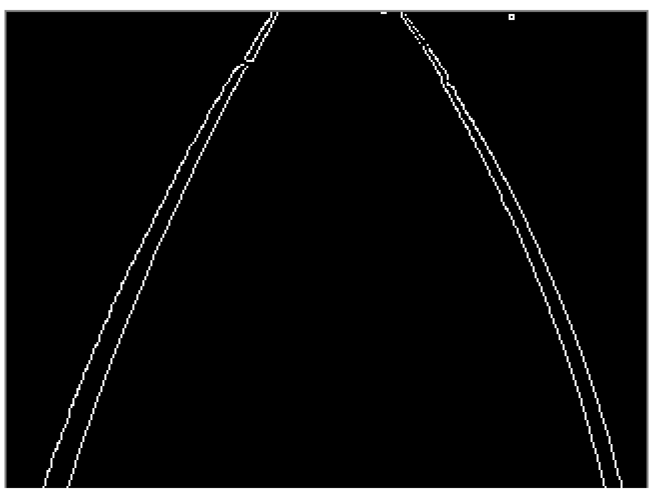

Figure 19: Hysteresis thresholding

Step 3: Region of Interest

The third step is to cut or crop that area which is not our concern because we are only focusing on the lanes and the main objective is to detect them so we have to crop those extra parts to do this first we have to create the triangular mask of the our region of interest so that it will increase the effectiveness of our image for later stages/steps then we merged it with the edge detected image and finally we get our final image which only have the edges of our region of interest.

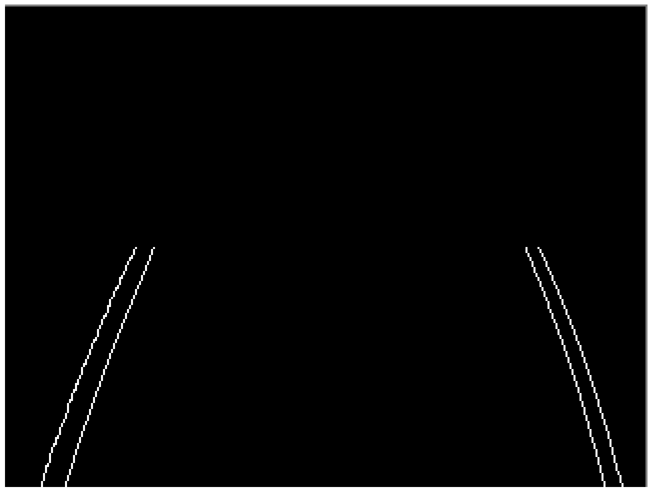

Figure 20: Cropped edges of the frame

Step 4: Line Segments Detection

The final step is to detect the segments of line because in the edges of the cropped frame, it is obvious that four lines are observed, symbolizing two lines of lanes but to a CPU, they are simply loads of white colored pixels on a dark black foundation. The goal is to find the coordinates of lanes from the white colored pixels and to get the coordinate of the lane line that are observed, for this open CV provides a function named as Hough Transform it is basically a technique which is used to detect the features like circles, lines and ellipses in an image, using it to locate linear lines from a loads of pixels that appeared to generate a line. The Hough Lines method attempts to fix several lines along every white colored pixel and return the foremost collection of lines. Hough

Line uses Polar Coordinates to detect lines. Coordinates of polar are better than coordinates of Cartesian, since they can portray all lines, even lines which are vertical that coordinates of Cartesian cannot due to the infinity of vertical line slope.

- Rho is the precision of distance in pixels. One pixel will be used.
- The degree of angle in the radian is precision.
- Minimum threshold is the number of votes necessary for a line segment to be considered. If more votes are for line then Hough Transform sees them as to have a segment of line recognized.
- Minimum Line of Length represents the least length of line segment in pixels.
- Maximum Line Gap represents the limit of two segments of the line in pixels, is divided and viewed as a one section of line.

Parameter's settings is a hit and try method and for the car with a camera running on a track with lanes, they need to be re-tuned.

Figure 21: Line segments

Step 5: <u>Combine Line Segments into Two Lane Lines</u>

The next step is to combine the two lane lines into line segments. Several segments of line with their endpoint coordinates were created, the question is how to merge them into only the two lines they care about, the right and left lane. One method which can be done is by defining the segments of the line by their slopes. The segments of line to the left must be sloping towards upward direction and on the left hand side of the screen while the segments of line to the right must be sloping towards downward direction and on the right hand side of the screen. There are a few special cases worth discussing aside from the logic mentioned above.

i) <u>One lane in the image</u>

The camera can spot both lanes in normal scenarios. There are instances, however, once the car starts to move, it wanders out of the track which is caused by a faulty steering angle or when the curve is too sharp.

ii) <u>Vertical line</u>

Segments of the line are incidentally distinguished as the car turns. In spite of the fact that they are not incorrect identifications, the reason is vertical lines have an infinity slope. Since lines which are vertical are not usual, this doesn't influence the general detection of lanes algorithm performance. Alternatively, the image's X and Y coordinates could be reversed, and will have zero slope that might be involved in average but segments of the horizontal lines will have an infinity slope, but that would be extremely unique. Since the camera points in the identical way as the lines of lanes. Another method is to represent the segment of lines in coordinates of polar, and then average distance and angle from the source.

Step 6: Planning of motion

Coordinates of line of lanes are created, have to turn the steering wheel to stay inside the lanes, and try to keep it in the center of the lanes. Essentially, given the lane lines observed, we need to determine the car's steering angle.

Step 7: Detected Two Lane Line

The direction of the heading can be measured by means of the end points of each lane. The line of red color shows the steering angle, should be in the center because the camera is mounted on the center of the vehicle and directing towards straight.

Figure 22: Two lane lines

After detection of two lanes lines, we detected single lane line which was quite challenging.

Step 8: <u>Single detected Lane Line</u>

Detecting one line of lane is challenging, as there will no longer be an average of two endpoints. If the camera sees only the left lane that means we have to steer it to the right and if it sees the right lane, have to steer it to the left direction.

Figure 23: Single detected lane

Since we know where we are going, we need to convert that into the steering angle, so that we can make decision and tell the car where to steer, the steering angle of 90 degrees is heading towards straight, 45 to 89 degrees turning towards left, and 91 to 135 degrees turning right.

We will calculate the slope from the coordinates of the lanes we get by using the two point's formula.

$$\text{Slope}=\frac{y2-y1}{x2-x1}$$

If the calculated slope is less than zero its mean we have detected the left lane and on the other hand if the slope is greater than the zero than we detected the right lane. To predict the steering angle we just have taken the average of two end point of the lanes x1 and x2 which gives us the x-offset we have to minimize the error which is the difference between the mid-point and the x-offset and to get y-offset we simple divide the height by two. Once we get the x-offset and y-offset we will calculate arctangent by dividing the two values using math.atan function which will be in radian.

$$y\_offset = \frac{height\ of\ the\ image}{2}$$

$$Angle\ in\ radians = \tan^{-1}\left(\frac{x\_offset}{y\_offset}\right)$$

After that we will convert it into degree which will be our final angle.

Angle in degree= $Angle\ in\ radians\ \times \frac{180}{\pi}$

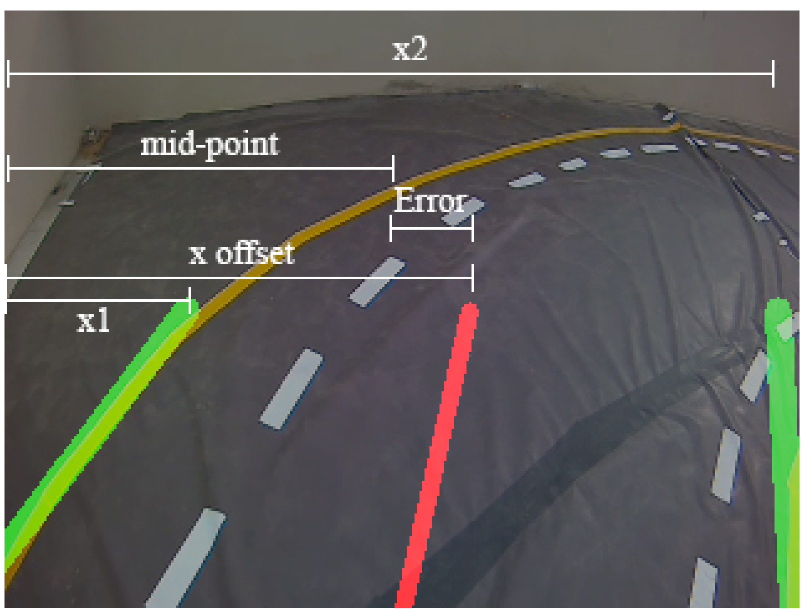

Figure 24: Angle predictions

When the car was being tested on the track, the car was continuously moving left and right, sometimes go completely out of the track. We at that point discovered that it is caused by the steering angles. We tested car on the track without the logic of stabilization to see the results. Sometimes, the steering angle of the car was 90 degrees but after some time, the steering angle jumped widely, 120 or 70 degrees .Therefore, the car was moving left and right within the lane. For this reason we need to stabilize the steering if the new angle is more than the degree from the current angle, just steer up to maximum angle deviation degree in the direction of the new angle. We used two types of maximum angle deviation, both lanes will be detected if degree is 5, one lane will be detected if the degree is 1.

3.3 Track Designing

The track dimensions are 21x37 ft. The design idea is taken from AWS Deep racer.

Figure 25: AWS Deep Racer [27]

Step 1: <u>Finding the track shape</u>

This is the official AWS Deep racer track. AWS and Udacity are collaborating to encourage AI and get ready understudies to test their aptitudes by taking part on the planet's first self-governing dashing association—the AWS DeepRacer League. Understudies with the top lap times will acquire full grants to the Machine Learning Engineer Nano degree program. This course will set you up to make, train, and adjust support learning models in the AWS DeepRacer 3D hustling test system. You will have the option to use the vehicle's tech specs, get together, and adjust to prepare and send your dashing model utilizing AWS in both recreated and true tracks.

Step 2: <u>Replicating the design into High resolution on Illustrator</u>

Figure 26: Illustrator design of the track

Step 3: <u>Rooftop with crayons</u>

Figure 27: Track designing on roof

Step 4: <u>Sizing it on original size i.e 21x37 ft</u>

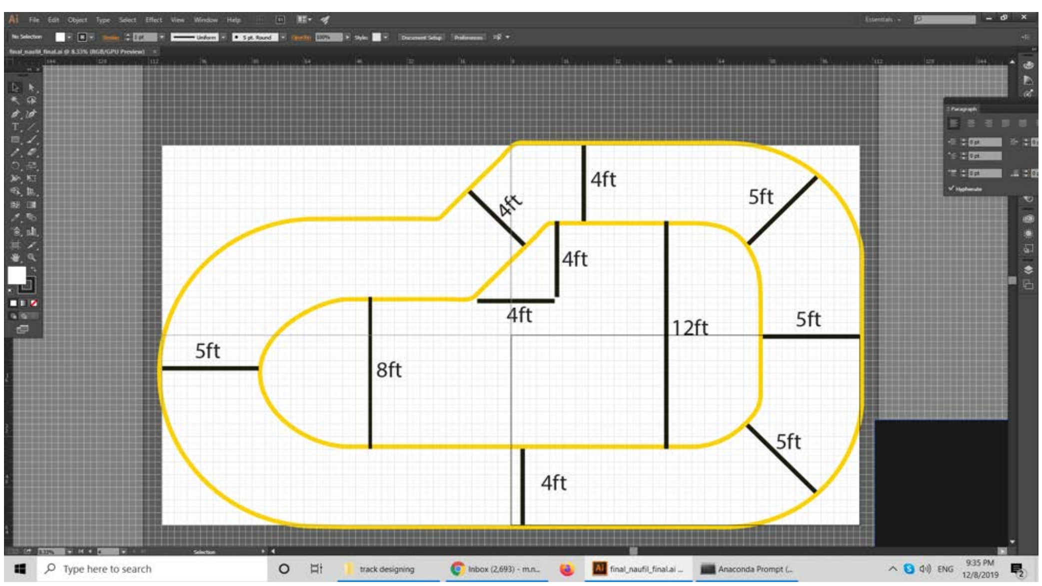

Figure 28: Photoshop track designing

Step 5: **Panaflex printing**

Figure 29: Panaflex printed

3.3.1 Track adjustment

The original track which had a size of 21 x 37 ft. was trimmed and cut because the track which was designed had a much sharped turns which was causing the algorithm to have problems detecting the lanes at those sharp turns and taking this under consideration, cut the sharp edges and smoothen the track. Tried different approaches to detect the lanes before cutting up those sharp turns .The goal was to bring the lanes of those turns close to each other so that it can easily detect the lanes of those turns and also changed the color of the lanes as well from yellow into white using Hough Transform technique but after having problems in detecting the white lanes, kept the Hough transform of yellow lanes

Figure 30: Track adjustment

After cutting some portion of the track, we had our desired track which can be seen in the figure.

3.4 Object detection

Object detection is the procedure of deciding the presence of an item and assessing its area in the picture canvas. Recognition of an object characterizes the object which is detected from the list of previously seen objects [28]. Technique of detection forms a rectangular box around the item. Technique of recognition classifies or arranges the object with a specific probability or faith in the prediction.

In a picture with numerous items, it is a demanding duty or task to decide all the individual objects location and then recognize the objects, because of few reasons:

- There can be a potential cover between various objects causing impediments for one or all.
- Object in the picture can have fluctuating directions.
- The objects must be somewhat present in the picture.
- Pictures from low fps video can be hazy and twist the highlights of the object.

3.4.1 TensorRT on Nano

TensorRT SDK is given by Nvidia to superior deep learning inference. It has an optimizer of inference which runs deep learning models with low latency in real-time situations. Nvidia claims induction on deep learning applications is up to multiple times quicker than CPU only platforms. It is based on CUDA and the equal programming highlight empowers advancement for various framework and improvement tools. The Jetson board provides FP16 compute power and utilizing TensorRT's chart improvements and fusion of kernel, level of production performance can be obtained for NLP, image segmentation and object detection [29].

3.4.2 <u>Pre-trained models</u>

An open-source repository provided by Nvidia with instructions for deploying deep learning models for applications and more. Among the given models, we utilize the SSD-MobileNet-v2 model, which represents single-shot identification on cell phones. It is an aspect of the DetectNet family. This model is pre-prepared on the MS COCO picture dataset of more than 91 distinct classes. It can identify or detect multiple objects in the same frame with occlusions, different directions and other unique nuances. The model is pre-prepared on normal items like seat, sofa, toaster oven and a few other regular objects [30].

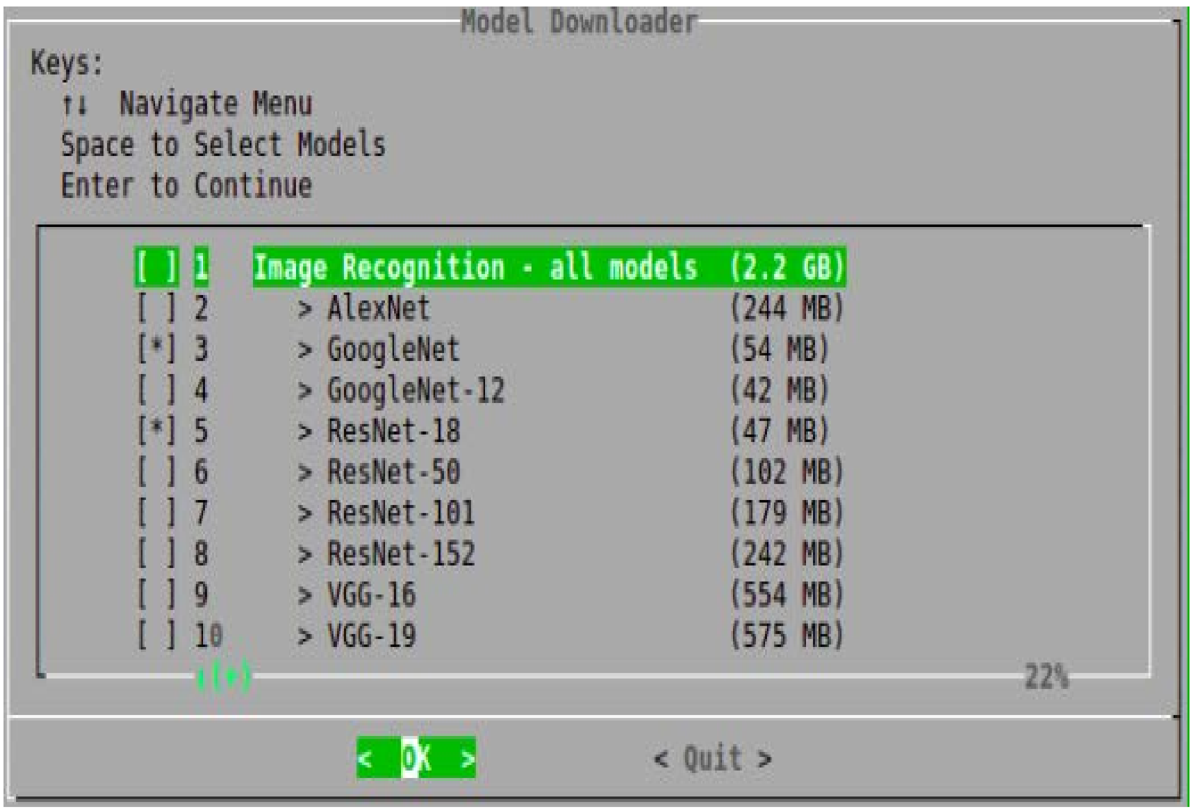

Figure 31: Selection of models

Finally, the code captures every image in the video stream and runs the detection algorithm and then draws translucent bounding boxes around the objects.

3.4.3 <u>Flowchart for object detection</u>

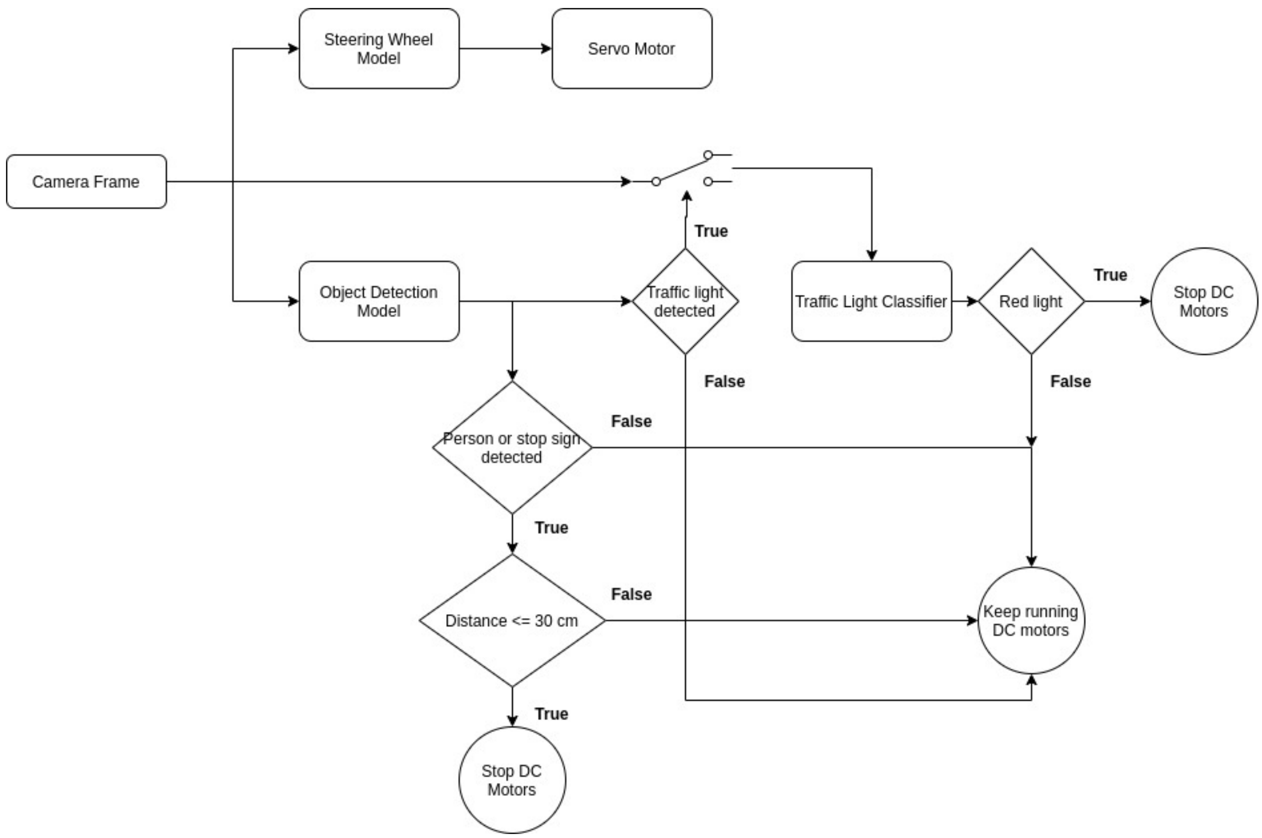

Figure 32: Flowchart for the decision making

Figure 33: Detection of stop sign and person

Chapter 4

Hardware

4.1 Design concept

The design of our project is to make a low cost supervised learning autonomous car using low power and can drive on its own on the custom track or road without the interference of humans and can detect the objects using wide angle camera and can avoid it with the help of sensors .For environment learning, supervised learning methodology is used

.Jetson nano is used as a microprocessor. Using Python programming and c language for the arduino, the circuits are interfaced with Jetson nano and arduino along with motor drivers .The motor drivers used are L298 driver and PCA9685 driver.

The L298 driver is used for dc motors used in cars and the pca9685 is used for servo motors. For the car to see and collect the data for lane detection and object detection, a wide angle camera with 160 degree angle above is used which is connected to the jetson nano in camera connector pin.

4.2 Data collecting process

The data collecting is the process where the camera is mounted on the car and the car is driven on the track with the help of the remote control to collect the data sets. Over hundred laps were taken on the track to collect the datasets. Thirty six thousand and five hundred frames were collected and these frames were trained using Google Collab Pro.

The camera is connected to the camera connector pin and buck converter is connected (j25) of jetson nano. The buck converter is used because the 12 volts cannot be given to the Jetson nano.

So to keep the voltage down we used a buck converter which gives approximately 5.3 volts to Jetson nano.

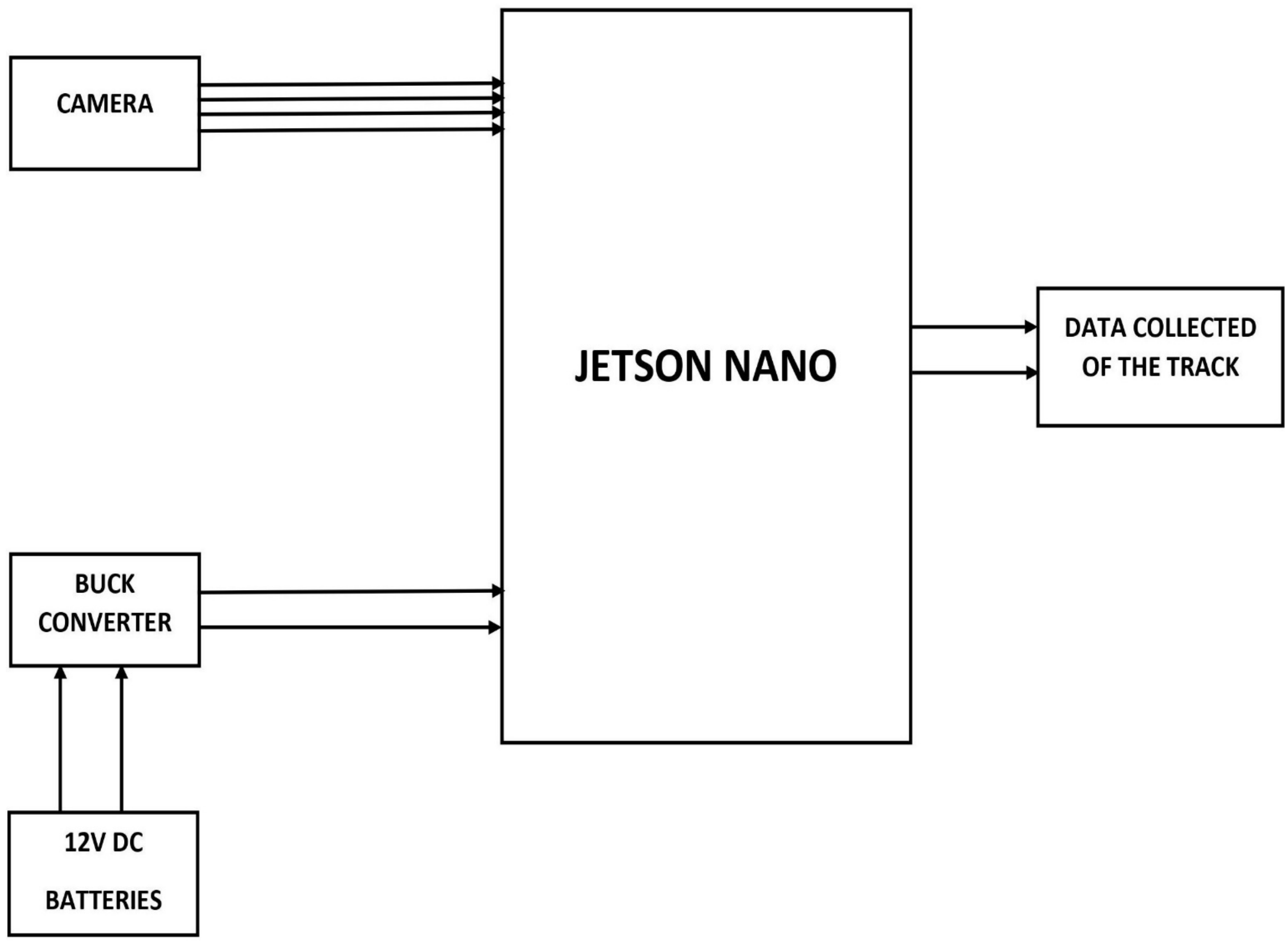

Figure 34: Block diagram for data collection

4.3 Supervised Learning Based Autonomous Car Hardware Block Diagram

The jetson nano is powered up by using buck converter which is connected to j25 pin .The 12V

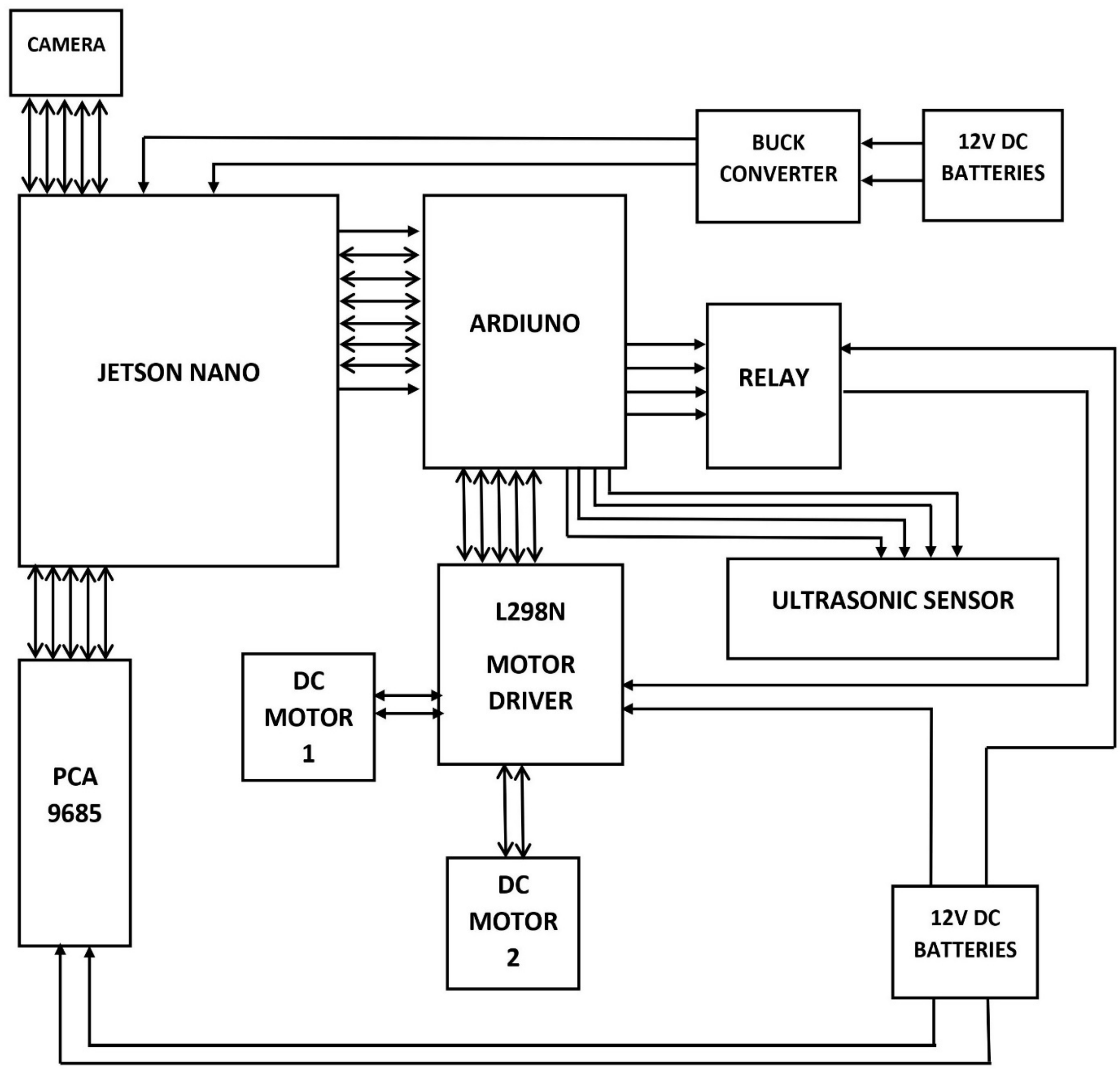

Figure 35: Block diagram of supervised learning autonomous car hardware

battery is connected to buck converter because jetson nano cannot handle that much voltage, so to lower down the voltage, buck converter is used which brings the voltage down to 5.3 volts.The wide angle camera which has an angle of view more than 160 degrees is connected to jetson nano at camera connector pin which shows both the lanes unlike the previous camera which was unable to show both the lanes.

The 40kg servo motor is connected to the PCA9685 motor driver's gnd ,vcc and pwm .And the motor driver's pin scl ,sda , vcc , gnd ,v+ ,oe, is connected to the jetson nano at pin j41.Another two 6 volts batteries are connected with dc motor which they are connected to the motor driver of L298 .The pins of L298 OUT1 and OUT2 is connected with dc motor 1 and OUT3 and OUT4 of L298 motor driver is connected with dc motor 2.The 5 volts are send to the arduino using the motor driver L298 and the purpose of using Arduino is that there is no PWM in jetson nano .

When the object is detected by the camera then it will measure the distance by using an ultrasonic sensor to determine how far the object is from the car. If the distance is less than 15cm than pin 7 of arduino will be high and will give output of 5 volt but 5 volts cannot be given to jetson nano pin because jetson nano gpio takes 3.3 volts input .So, we step down 5 volts to 3.3 volts using 3.3BT and give the voltage to jetson nano pin number 18.we have made the condition, when pin 18 is high, it will cut off the supply of arduino and the car will stop.

Chapter 5

Results and discussions

In this chapters we will discuss about the results of the lane detection and object detection

5.1 Lane Detection

Picture

Figure 36: Lane test on picture

Real Time

Figure 37: Lane detection test on real time

<u>Video</u>

Figure 38: Lane detection on video

<u>On Track</u>

Figure 39: Lane detection test on track

5.2 Data set

After the completion of the track adjustment, over hundreds laps were taken on the track to collect the datasets .Once the datasets were collecting, we changed the angles of the frames where it was causing the problems .After changing those specific frame's angles all the datasets then were trained on the Google Collab pro. The results are as follow:

Figure 40: Data set collecting

5.3 Training of the data sets

To train a model Nvidia's model architecture is used.

5.3.1 <u>Angle distribution</u>

We created specific graphs to understand how many kinds of angles there are in our frame .We can notice in the graph that it contains mostly 90 angles. This makes more sense because the data sets which we were collecting on our custom made track, the car was usually going straight. The distribution graph is shown below.

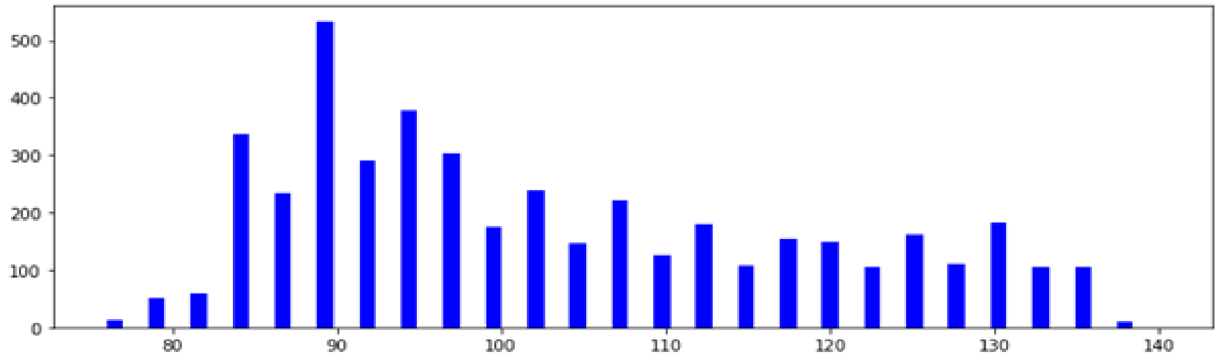

Figure 41: Angle distribution graph

5.3.2 <u>Train and valid data</u>

After the distribution of angles and finding out which angles were mostly there in the datasets, we made the distribution of train and valid graph and we make sure they both are consistent. Once we trained the model, we tested the performance of the model on validation and we can see the validation data and training data are mostly uniform which is what we wanted.

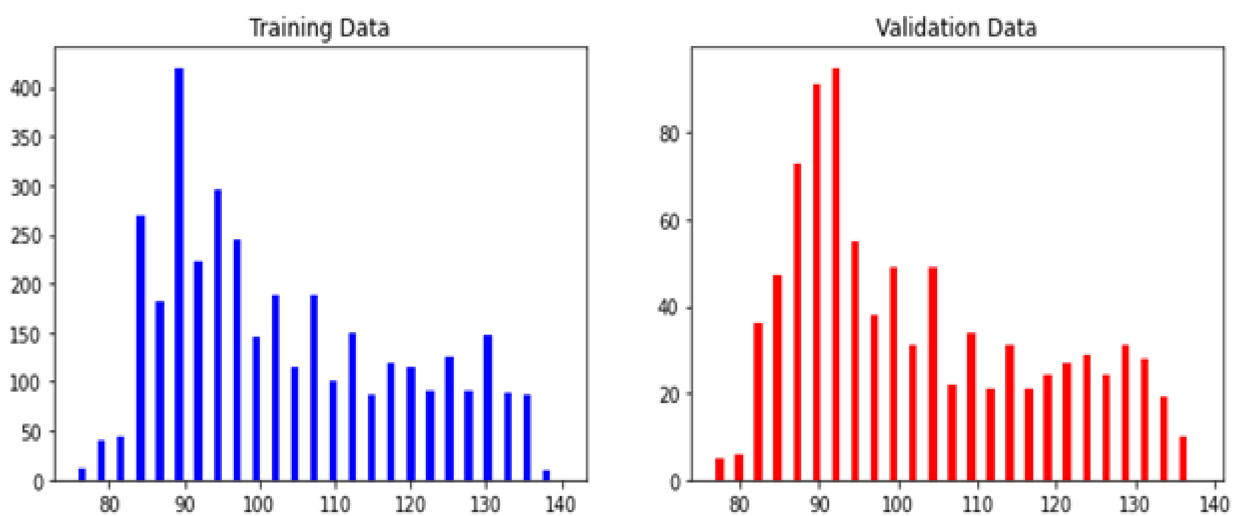

Figure 42: Train and valid data graph

5.4 Image augmentation

The exhibition of deep learning neural networks consistently improves with the proportion of data accessible. Image augmentation is a strategy to artificially make new preparing information from existing preparing information or data. This is done by applying a specific domain method to model from the preparation information that makes new and differing preparing models. Image data augmentation expansion is might be the most striking kind of information growth and remembers making changed renditions of pictures for the preparation dataset that have a spot with a comparative class as the primary picture. Changes incorporate a scope of tasks of picture control or model, zooms, shift, slip and generously more [31].

The intention is to develop or extend the preparation dataset with new doable models. This infers sorts of the training set of the pictures that are going to be seen by the model. For instance a horizontally flipped picture of our track makes sense, because the photo could have been taken from the left or right. A vertically flipped picture of the track would not be appropriate given, that the model is very unlikely to see a photo of an upside down of the track. The decision of the particular data augmentation procedures utilized for a preparation dataset must be picked carefully and inside the setting of the preparation dataset and data of the difficult area. It tends to be valuable to explore different avenues regarding data augmentation techniques in seclusion and in show to check whether they bring about a quantifiable improvement to display execution, maybe with a little model dataset, model, and preparing run. Modern profound learning calculations, for example, the convolutional neural network or CNN, can learn highlights that are invariant to their area in the picture. In any case, growth can additionally help in this change invariant way to deal with learning and can help the model in learning highlights that are likewise invariant to changes, for example, left-to-right to top to down requesting, light levels in photos, and then some. Image data augmentation is typically only applied to the training dataset, and not to the validation or test dataset. We have used different ways of image augmentation which are flipping, introducing Gaussian blur, adjusting the image brightness, and panning out a smaller image from the center.

5.4.1 Zooming the frame

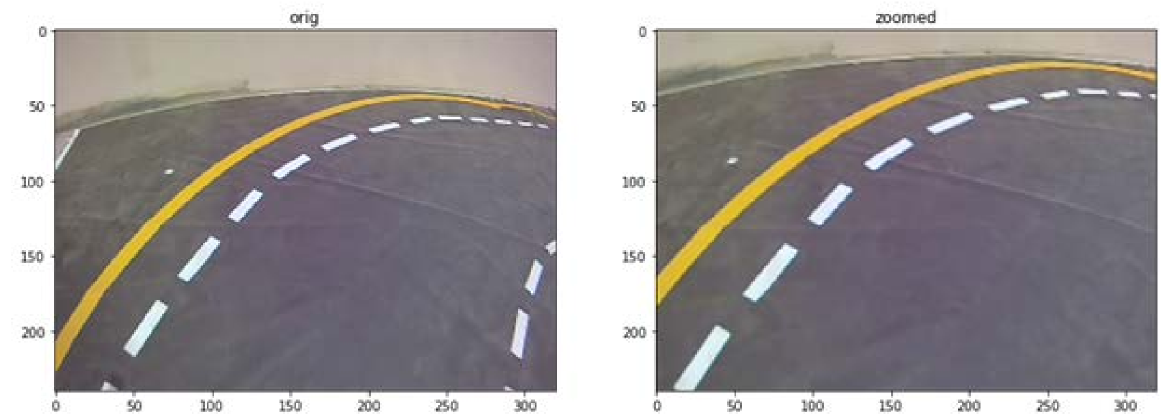

Figure 43: Frame zoomed

As we can see the right side of the frame has been zoomed in, from 100 percent (no zoom) into 130% zoom.

5.4.2 Pan the frame

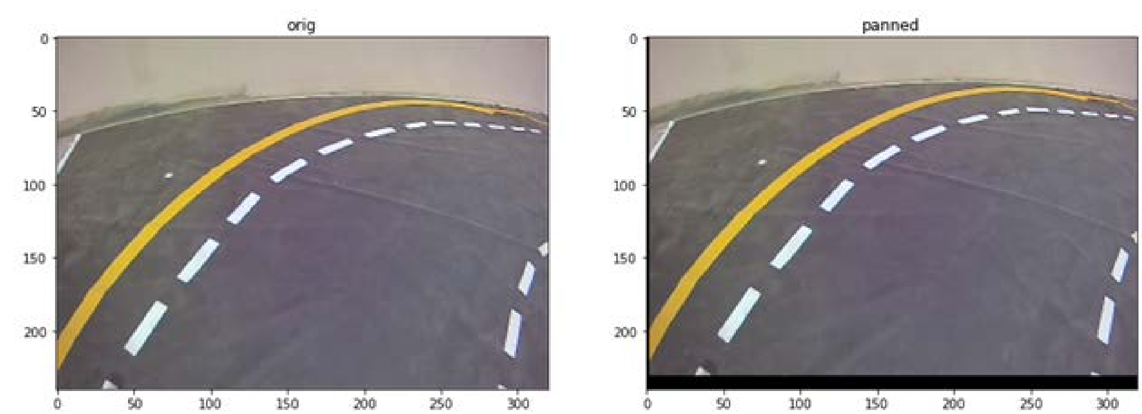

Figure 44: Pan the frame

We have panned the frame on the right hand side from left, top and bottom up to 10 percent.

5.4.3 **Brightness adjustment**

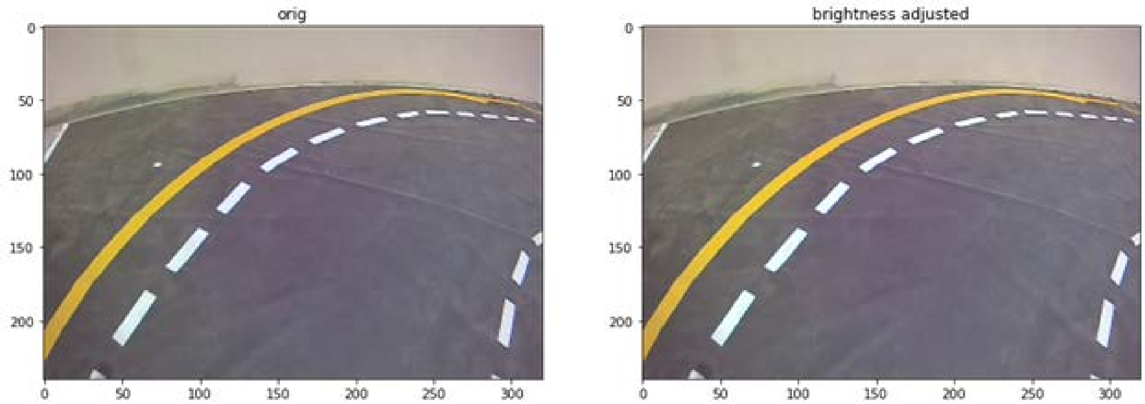

Figure 45: Brightness adjustment

We made brightness adjustments for the data set which can be increased or decreased up to 30%.As we can see on the right hand side, the brightness has been increased on its own.

5.4.4 **Flipping the frame**

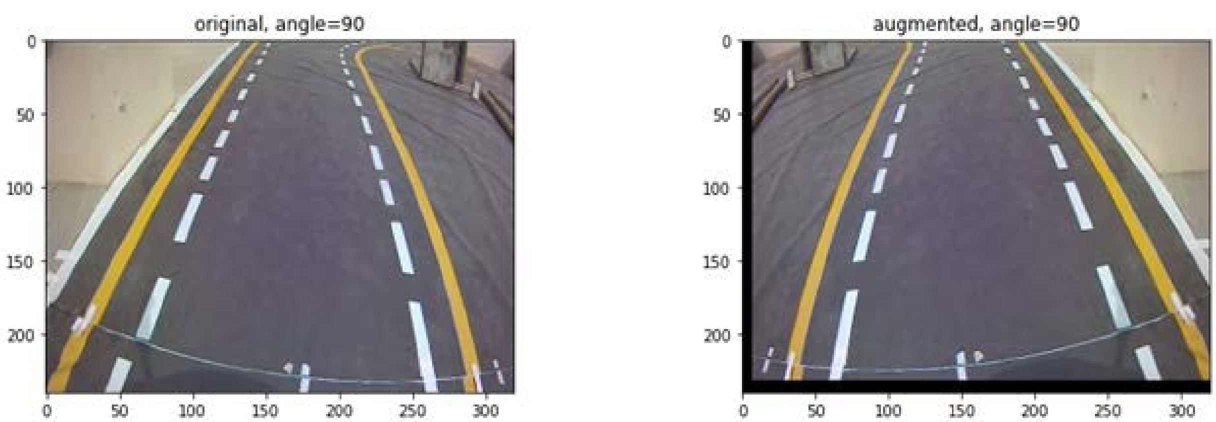

Figure 46: Frame flip

We have flipped the frame horizontally. For example, do a left to right flip, and change the steering angle correspondingly. After doing these techniques separately, we merged them all into one and we got different results as we can see in the figure below.

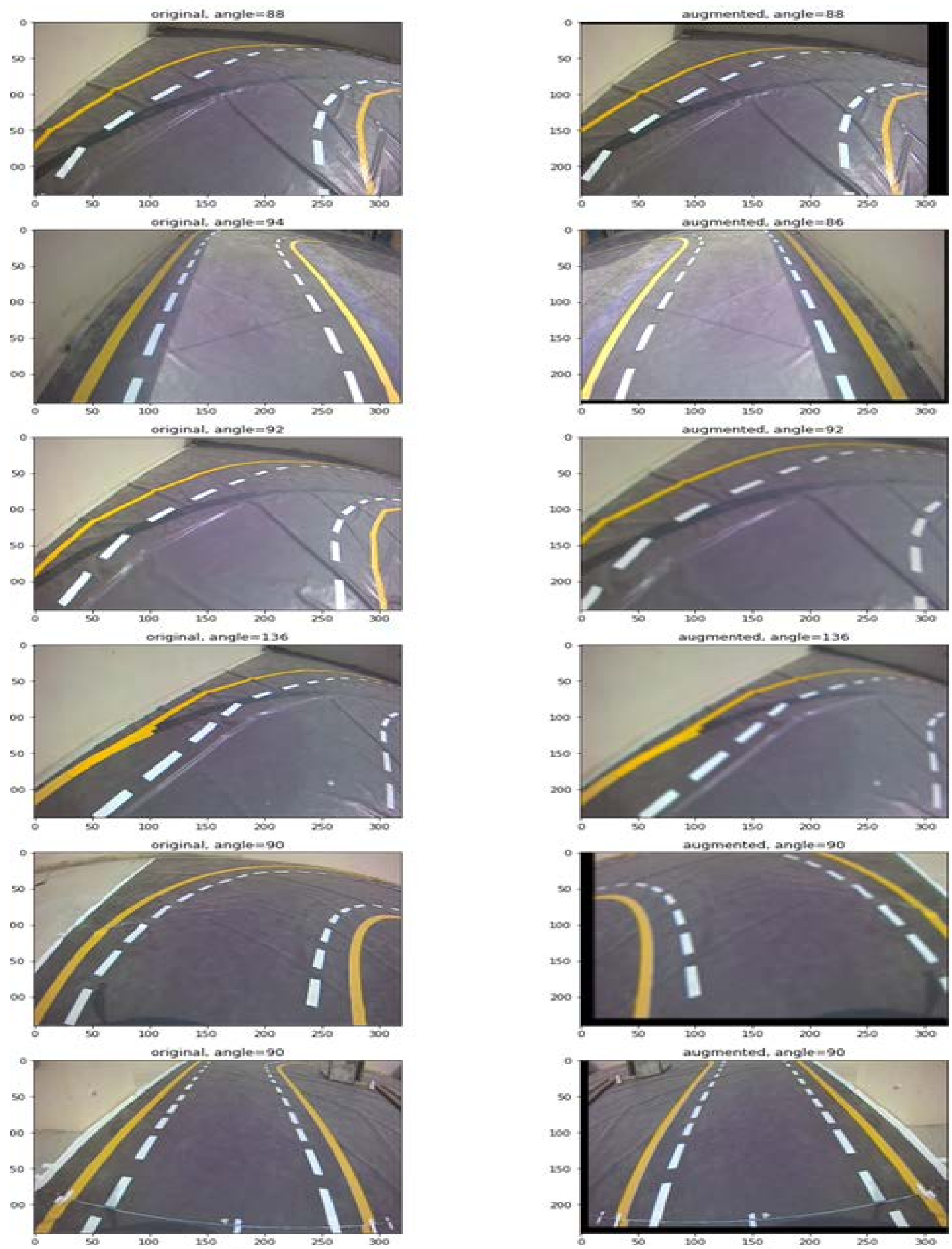

Figure 47: Image data augmentation techniques

5.5 Preprocess Training data for Nvidia model

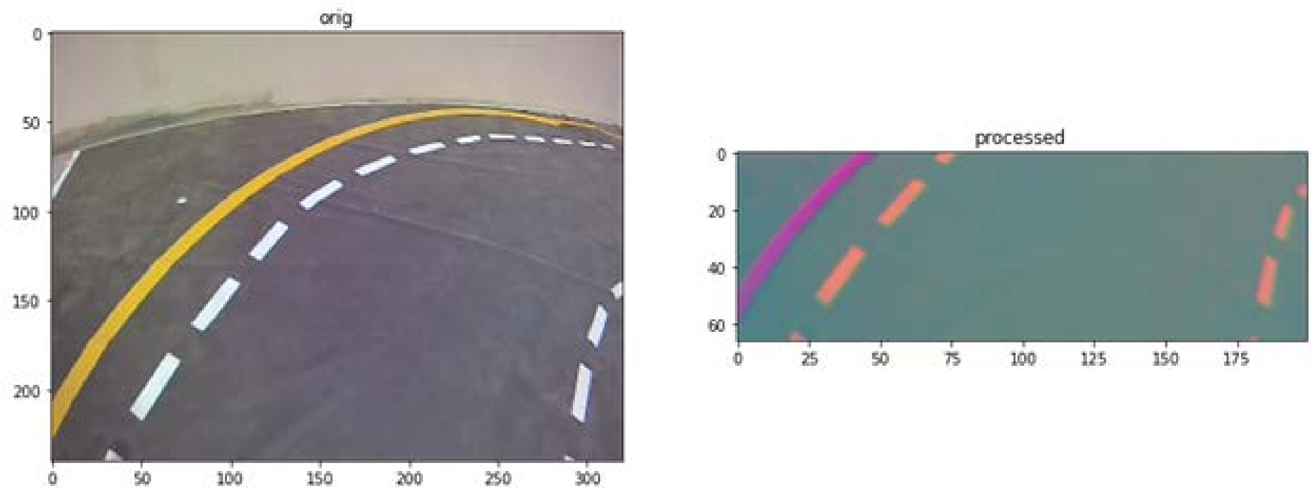

Figure 48: Lane detection using Nvidia Model.

We removed the top half of the image because it is not relevant for lane detection and the Nvidia model said that it is best to use YUV color space. The YUV model defines a color space in terms of one luma component (Y′) and two chrominance components, called U (blue projection) and V (red projection) respectively. The input image size is (200, 66)

5.6 Trained model loss

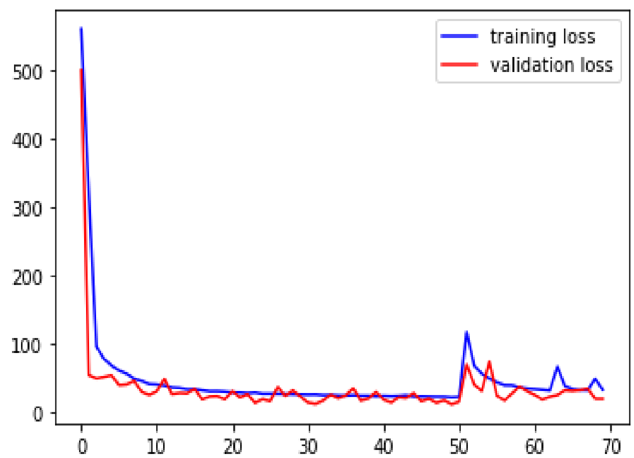

Figure 49: Loss of trained model

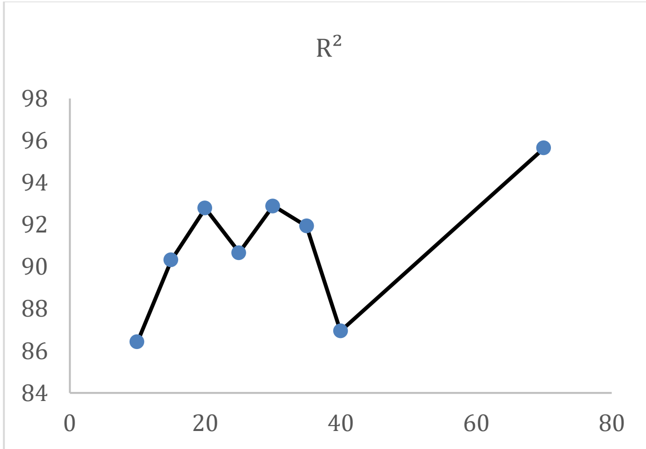

Figure 50: Accuracy of the model

As we can see the R squared of the predicted was around 95% and Mean Squared Errors (MSE) are low, indicating the model is predicting a steering angle every similar to our hand coded land follower, which was used as the model input.

5.7 Object detection

Figure 51: Camera detecting the object and person

Figure 52: Object detection

Chapter 6

Conclusion

The final findings of this project results into a low cost, low power and efficient Autonomous Car which can drive autonomously on custom made track without the interference of humans and detect the object as well and stop the car once the object is detected and with the accuracy of 95% along with (MSE) of 12 .

6.1 General improvements

This system can be improved by slightly changing and adding some features suggested below:

- Improving the code for the lane detection for much better results.
- The hardware system for the car can be improved.
- More car decision making techniques can be implemented .For example, when the car is stopped, it can make the decision to turn left or right.
- Steering angle predictions technique can also be improved as well.

6.2 Future recommendations

Some future work can be done to advance this project taking artificial intelligence and electronics technology to the next level.

- A self- parking system can be implemented in the project to park on its own.
- GPS systems can help the car to find nearby parking systems with the help of the android application.

- Implementing the project on a real car.

6.3 <u>Difficulties Faced</u>

- Steering wheel mechanism was a major problem because different mechanisms were used and failed many times.
- Collecting data sets for the car.
- Training thousands of datasets and filtering out the wrong predicted angles.
- Power management for the Jetson Nano and motors.
- Choosing the camera for the car because the previous used was raspberry pi cam which had an angle of view of not more than 60 degrees and it was causing many problems for the car to see both lanes .For this reason , wide angle camera was used which had an angle of view more than 160 degrees.
- Choosing a servo motor for the car and placement of camera on the car
- Sometimes the algorithm could not detect traffic light because the light we used for traffic signal, is not bright enough to be detected in bright day light

  .

References

[1] americans-spend-average-17600-minutes-driving-year

[2] Self –Driving cars and Rural Areas: The potential for Symbiotic Relationship

[3] Why Develop Self-Driving Cars?

[4]Nieuwenhuijsen, J., Almeida Correia, G.H., Milakis, D., et al.: ‘Towards a quantitative method to analyze the long-term innovation diffusion of automated vehicles technology using system dynamics’, Transp. Res. C, Emerg. Technol., 2018, 86, pp. 300–327.

[5]Milakis, D., van Arem, B., van Wee, B.: ‘The ripple effect of automated driving’. Proc. BIVEC-GIBET Transport Research Day, Eindoven, TheNetherlands, May 2015, pp. 1–3

[6]Milakis, D., van Arem, B., van Wee, B.: ‘Policy and society related implications of automated driving: a review of literature and direction for future research’, J. Intell. Transp. Syst., 2017, 21, (4), pp. 324–348

[7] Environment self-driving-cars-could-impact-environment

[8] 3 Amazing Benefits of Driverless Cars That You May Have Never Imagined

[9] Ucsusa org resources self-driving-cars-101

[10] driverless-cars-pros-and-cons

[11] Traffic Survey by AAA Foundation: https://aaafoundation.org/2018-traffic-safety-culture-index/

[12] 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

[13] History of Autonomous Driving

[14] The Road to Driverless Cars: 1925 –2025

[15] End to End Learning for Self-Driving Cars, NVIDIA Corporation, Holmdel, NJ 07735

[16] driving-advice/the-future-of-autonomous-vehicles

[17] C. B. S. T. Molina, J. R. d. Almeida, L. F. Vismari, R. I. R. González, J. K. Naufal and J. Camargo, "Assuring Fully Autonomous Vehicles Safety by Design: The Autonomous Vehicle Control (AVC) Module Strategy," *2017 47th Annual IEEE/IFIP International Conference on*

Dependable Systems and Networks Workshops (DSN-W), Denver, CO, 2017, pp. 16-21, doi: 10.1109/DSN-W.2017.14. Failed many times.

[18] Thierer, A.D., Hagemann, R.: ‘Removing roadblocks to intelligent vehicles and driverless cars’, Wake Forest J. Law Policy, 2015, 5, pp. 339–391

[19]tA. R. Fayjie, S. Hossain, D. Oualid and D. Lee, "Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment," *2018 15th International Conference on Ubiquitous Robots (UR)*, Honolulu, HI, 2018, pp. 896-901, doi: 10.1109/URAI.2018.8441797.

[20] DeepPicar: A Low-cost Deep Neural Network-based Autonomous Car, Michael G. Bechtel, Elise McEllhiney, Minje Kim, Heechul Yun, University of Kansas, USA.

[21]318338750_Supervised_Machine_Learning_Algorithms_Classification_and_Comparison

[22] nvidia-jetson-nano-developer-kit-detailed-review

[23learnabout-electronics.org/PSU/psu31.php

[24] servo-motor-types-working-principle-explained

[25] cys-s8218-40kg-digital-metal-gear-servo

[26]https://www.researchgate.net/figure/NVIDIAs-PilotNet-architecture-for-end-to-end-vehicle-control-23_fig1_324409557

[27] aws- deepracer- how-to-build-your’s-own-race-track-at-home-9fa74fea2ca9

[28] recognition-with-deep learning

[29] deep-learning-with-jetson-nano-real-time-object-detection-and-recognition

[30] object-detection-using-ssd-mobilenetv2-using-tensorflow-api-can-detect-any-single-class-from-31a31bbd0691

[31] image-augmentation-for-convolutional-neural-networks-18319e1291c

Plagiarism Report by Turnitin

Supervised Learning Based Autonomous Car

ORIGINALITY REPORT

16%	12%	6%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	3celectrons.com Internet Source	2%
2	towardsdatascience.com Internet Source	1%
3	machinelearningmastery.com Internet Source	1%
4	ktechnics.com Internet Source	1%
5	ruc.udc.es Internet Source	1%
6	Submitted to Universiti Teknologi MARA Student Paper	1%

7	mafiadoc.com Internet Source	1 %
8	Submitted to Engineers Australia Student Paper	1 %
9	www.mynics.in Internet Source	1 %
10	Submitted to University of Wolverhampton Student Paper	<1 %
11	everythingtutorial.com Internet Source	<1 %
12	Submitted to Florida Virtual School Student Paper	<1 %
13	hdl.handle.net Internet Source	<1 %
14	Submitted to Korea National University of Transportation Student Paper	<1 %

Student Paper

15	www.hindawi.com Internet Source	<1%
16	en.wikipedia.org Internet Source	<1%
17	Submitted to The RCY Colleges & Institutes Student Paper	<1%
18	Submitted to St Joseph Engineering College, Mangalore Student Paper	<1%
19	www.engineering.com Internet Source	<1%
20	Submitted to VIT University Student Paper	<1%
21	Submitted to Victoria University Student Paper	<1%
22	Submitted to muetjamshoro Student Paper	<1%

21	Submitted to Victoria University Student Paper	<1%
22	Submitted to muetjamshoro Student Paper	<1%
23	www.slideshare.net Internet Source	<1%
24	blueandgreentomorrow.com Internet Source	<1%
25	Submitted to Liberal Arts and Science Academy Student Paper	<1%
26	Submitted to University of North Texas Student Paper	<1%
27	uk.shopping.com Internet Source	<1%
28	Michael G. Bechtel, Elise Mcellhiney, Minje Kim, Heechul Yun. "DeepPicar: A Low-Cost Deep Neural Network-Based Autonomous Car", 2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems	<1%

Publication

29	Entregado a Higher Education Commission Pakistan el 2011-10-24 Student Paper	<1%
30	Submitted to DeVry, Inc. Student Paper	<1%
31	Submitted to CSU, San Jose State University Student Paper	<1%
32	Submitted to The Robert Gordon University Student Paper	<1%
33	edu.gaitech.hk Internet Source	<1%
34	Submitted to K. J. Somaiya College of Engineering Vidyavihar, Mumbai Student Paper	<1%
35	Submitted to Institute of Technology, Nirma University Student Paper	<1%

39	Shan Ullah, Deok-Hwan Kim. "Benchmarking Jetson Platform for 3D Point-Cloud and Hyper-Spectral Image Classification", 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), 2020 Publication	<1%
40	developer.nvidia.com Internet Source	<1%
41	www.computer.org Internet Source	<1%
42	eprints.utm.my Internet Source	<1%
43	docshare.tips Internet Source	<1%
44	www.cysmodel.com Internet Source	<1%
45	www.pwc.com.au Internet Source	<1%

47	arsalananwer.net Internet Source	<1%
48	Anukriti Singh, Aditi Chandra, Divya Priyadarshni, Nisheeth Joshi. "Self Parking car prototype", 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2018 Publication	<1%
49	Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, Deok-Jin Lee. "Driverless Car: Autonomous Driving Using Deep Reinforcement Learning in Urban Environment", 2018 15th International Conference on Ubiquitous Robots (UR), 2018 Publication	<1%

Exclude quotes	Off	Exclude matches	< 6 words
Exclude bibliography	On		