DeepC Car

Minor Project II

Submitted By:
Shubham Kumar Tiwari (16102021)
Aman Sharma (16102243)

Under the supervision of:
Mr. Ankur Bhardwaj

Department of ECE

Jaypee Institute of Information Technology, Noida

May 2019

AKNOWLEDGEMENT

It is our privilege to express our sincerest regards to our project coordinator,Mr. Ankur Bhardwaj for his valuable inputs, guidance, encouragement, whole-hearted cooperation and constructive criticism throughout the duration of our project.

We deeply express our sincere thanks to our Head of Department Prof. ShwetaShrivastavafor encouraging and allowing us to present the project on the topic “DeepC Car” at our department premises for the partial fulfillment of the requirements leading to the award of B-Tech degree. We take this opportunity to thank all our lecturers who have directly or indirectly helped our project.

We are thankful and fortunate enough to get constant support and guidance from all Teaching Staff of Department of ECE who helped us in successfully completing our project work.

We pay our respects and love to our parents and all other family members for their love and encouragement throughout our career. Last but not the least we express our thanks to our friends for their cooperation and support.

Signature:

Names:

Shubham Kumar Tiwari (16102021)

Aman Sharma (16102243)

ABSTRACT

Over the last years the automotive industry has advanced significantly towards a future without human drivers. Researchers are currently been trying to overcome the technological, political and social challenges involved in making autonomous vehicles mainstream. These vehicles need to be safe, reliable and cost-efficient. Connecting them and creating coordination mechanisms could help achieve these goals.

The idea is to build a laptop as a central processing unit based project to demonstrate a working model of a Deep Neural network (DNNs) [1] based car that is able to control its acceleration and steering angle according to the path and also able to detect various obstacles that come its way. The car has a camera and obstacle detecting sensors mounted on the car. Camera captures images and provides it to the Convolutional Neural Network (CNNs) that processes the frame (image) in real time and produces steering angle and acceleration as the output.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The development of autonomous cars is strongly linked to the need and the drive to enhance safety in vehicles. It turns out that people are pretty bad drivers. In 2013, over 1 million deaths occurred on the roads worldwide. This is the 8th leading cause of death globally, and the number one cause of young people’s death. Global losses due to road traffic injuries are estimated at over half a trillion dollars. Human error accounts for around 90% of all vehicle crashes. The reasons? –poor lookout- excessive speed - inattention- wrong evasive action - internal distraction.

So, what are autonomous cars?

The vehicle fully controls itself, only needing occasional input from the driver; – Complete and full autonomy. Example - latest version of the Google car, which has no steering wheel.

In 2009, Google started to develop Chauffeur, and by now it has driven over 700,000 miles, wholly autonomously. It can detect hundreds of types of objects simultaneously, and it has not caused any accidents yet.

On the other side, autonomous cars will provide personal transportation for millions of disabled or elderly people .Connected vehicles means police could stop them remotely if a criminals inside.

Autonomous vehicles will revolutionize transportation. Self-driving cars will save millions of lives in situations where it is impossible for a person to prevent a car accident. The reaction times and alertness of machine are much better. In addition, long distance cameras and ultrasonic Sensors/LIDARs further equip these cars with super-human abilities. For such reasons, we see many corporations and researchers trying to develop technologies that would allow for a fully autonomous driving experience.

1.1Project Goal

The project focuses on designing autonomous car to improve the standard of living and to help minimize the injuries and death due to high road accident. There are several benefits and perhaps a few drawbacks we could list but the main goal of autonomous driving is to remove the human element from the equation. The vast majority of accidents are caused by human error. There are other factors and there will still be things that go wrong but the number of auto accidents would decrease dramatically if people were not behind the wheel making bad decisions.

Another benefit you don’t hear about much: self-driving cars will never fall asleep at the wheel because they have been awake too many hours or took too many pills and drove anyway. They will never have a few too many drinks, text and drive or be distracted by other people in the car.

CHAPTER 2
BACKGROUND STUDY

The motivation behind the project is to prevent and reduce the damage caused by road accidents by saving their lives.

2.1 Levels of Autonomous Driving

The National Highway and Traffic Safety Administration (NHTSA) defines
five levels of vehicle autonomy [2].

• ***no automation*** *(level 0)* — The human driver does all the driving.

• ***driver assistance*** *(level 1)* — The car assists the driver with a single vehicle control such as steering or braking, but not both at the same time.

Example:*lane keeping, cruise control or assisted breaking.*

• ***partial automation*** *(level 2)* — The execution of both steering and accelerating/ decelerating is performed by the car. Human driver is still expected to perform all remaining tasks associated with driving and monitoring of the driving environment.

Example:*Tesla Autopilot*

• ***conditional automation****(level 3)* — The driving system takes complete control over the entire driving task under special circumstances. Human driver is expected to intervene when required. When the circumstances aren't met, human driver does all the driving.

Example:*Waymo (Google) self-driving car*

• ***high automation*** *(level 4)* — The driving system takes complete control over the entire driving task under special circumstances. The human driver does not need to pay attention or intervene in those situations. If a change of circumstances arises, the car needs to be able to stop and park safely in case the driver does not retake control.

Example:*Waymo announced in 2017 that they are testing level 4 driving[3]*

• ***full automation*** ***(level 5)***— The driving system takes complete control over the entire driving task under **all** circumstances. The human driver does not need to be inside the car.

Example:*JohnyCab from Total Recall*

2.2Literature Survey

The first big breakthrough in the use of end-to-end deep learning for self-driving cars is dated to 1989 and the accomplishments of Pomerleau, who built the Autonomous Land Vehicle in a Neural Network (ALVINN)[4]. He used a simple feed forward neural network with a single hidden layer of 29 units. This research shows that it is possible to use end-to-end neural nets to drive a car. In 2004, the Defence Advanced Research Projects Agency (DARPA) came up with a project known as DARPA Autonomous Vehicle (DAVE), which is a 1/10th scale autonomous RC car prototype able to drive in terrain and avoid obstacles. DAVE served as groundwork for the most recent achievement of NVIDIA in the field of self-driving cars – the DAVE-2. In the paper *End-To-End Learning for Self-Driving Cars*, NVIDIA published a state-of-the-art network architecture that benefits from the modern convolutions and processing power of present-day GPUs. Their car prototype was able to drive on highways and in simple traffic on local flat roads.

Modern self-driving car companies do not use end-to-end learning as it is infeasible to collect enough training data, to cover all possible scenarios in a real world driving experience. In 2016, in a regime where extremely high accuracy is necessary, the amount of data required to train an end-to-end system grows exponentially compared to a modular system. In a modular approach, the system is broken down into sub-modules with different responsibilities as pedestrian detection or path planning. Each module is then trained either using machine learning or manual engineering, depending on empirical success.

2.3Proposed System

This car should be able to drive itself on a flat surface that will mimic a simplified road. The main input of the car will be real time video from a camera, which will be mounted on the top. The system should then output corresponding steering commands and control the car accordingly. Since the camera will be the only input for the controller, the goal of the project will be to teach the car how to steer. We will use ultrasonic sensors to detect obstacles on the road and stop the car accordingly. This will work as a separate module that will not interfere with the process of steering which will be managed exclusively by the neural network.

CHAPTER 3
HARDWARE COMPONENTS

3.1 Autonomous Car Module

The hardware components used are:

- **Arduino UNO**:The Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button.

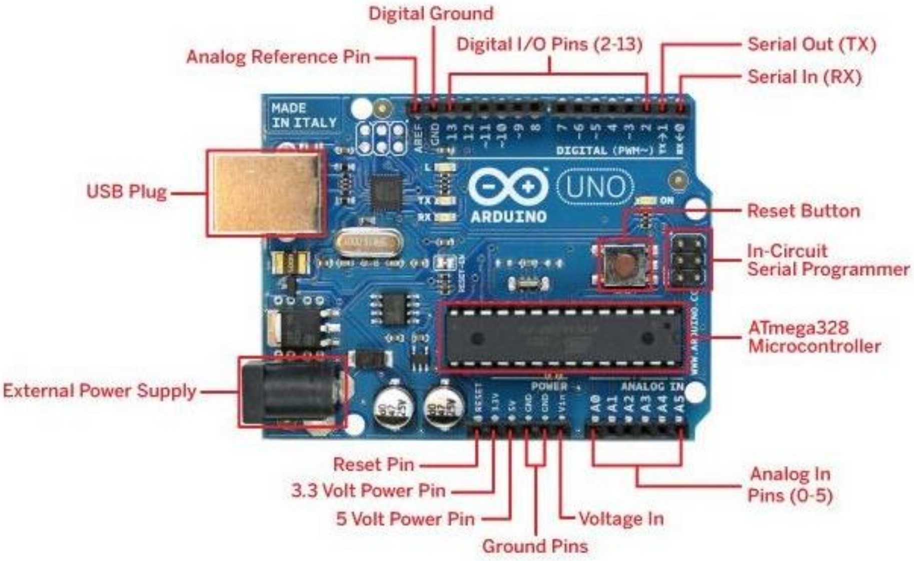


Figure 3.1:Description of Arduino UNO[5]

- **IC 7805**:7805 is a voltage regulator integrated circuit. It is a member of 78xx series of fixed linear voltage regulator ICs. The voltage source in a circuit may have fluctuations and would not give the fixed voltage output. The voltage regulator IC maintains the output voltage at a constant value i.e 5V.

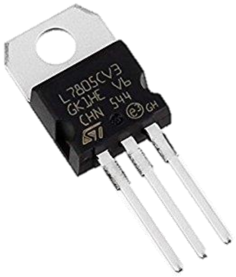


Figure 3.2:IC 7805[6]

- **Ultrasonic Sensors(SR-04)**:It measures distance by using ultrasonic waves. The transmitter head emits an ultrasonic wave and receiver receives the wave reflected back from the target. Ultrasonic Sensors measure the distance to the target by measuring the time between the emission and reception.

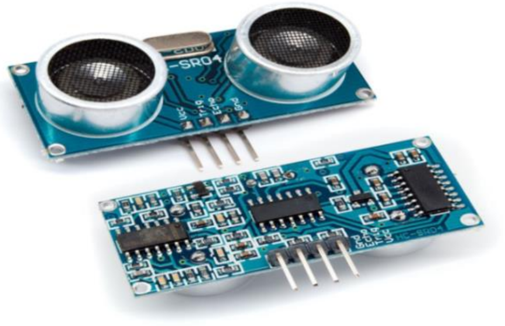


Figure 3.3: Ultrasonic sensor(SR-04)[6]

- **USB Webcam**:It is a video camera that feeds or streams its image in real time to or through a computer to a computer network. When "captured" by the computer, the video stream may be saved, viewed or sent on to other networks travelling through systems such as the internet, and emailed as an attachment. When sent to a remote location, the video stream may be saved, viewed or on sent there.

Figure 3.4:USB Webcam[8]

- **DC Motor:** These motors have been used for driving the tyres using arduino.

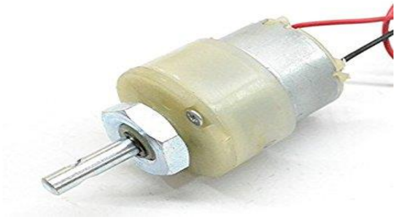


Fig 3.5:DC Motor[9]

CHAPTER 4
DETAILED DESIGN

The principal aim of the project is to do a small scale implementation of a self -driving RC car with "level-3" automation. The car works on the principle of Convolution Neural Networks(CNNs) by taking camera image as the input.

4.1 Autonomous Car Module

The car module uses a CNN model to predict the steering angle of the car so that it is able to follow the road on which it is travelling. On being fed to the model, it generates the required steering angle which helps the car follow the road on which it is moving. Also, ultrasonic sensors are used to prevent the car from hitting the immediate obstacles that come its way. Ultrasonic sensors come along with many advantages. They can be used in dark environments and are hence preferred over IR sensors. Ultrasonic sensors are also not affected by the color or transparency of objects. They are also budget friendly thus making the cost of the car affordable.

4.1.1 Assembling the hardware

The required connections are made using the Arduino and ICs. The ultrasonic sensors are connected according to their pin description. Arduino measures the values from all 5 ultrasonic sensors and then sends these values to the laptop via serial communication. Laptop then predicts the steering angle and also does the decision making based on priorities set in the code to generate a decision based on the ultrasonic sensors data as well as the CNN model.

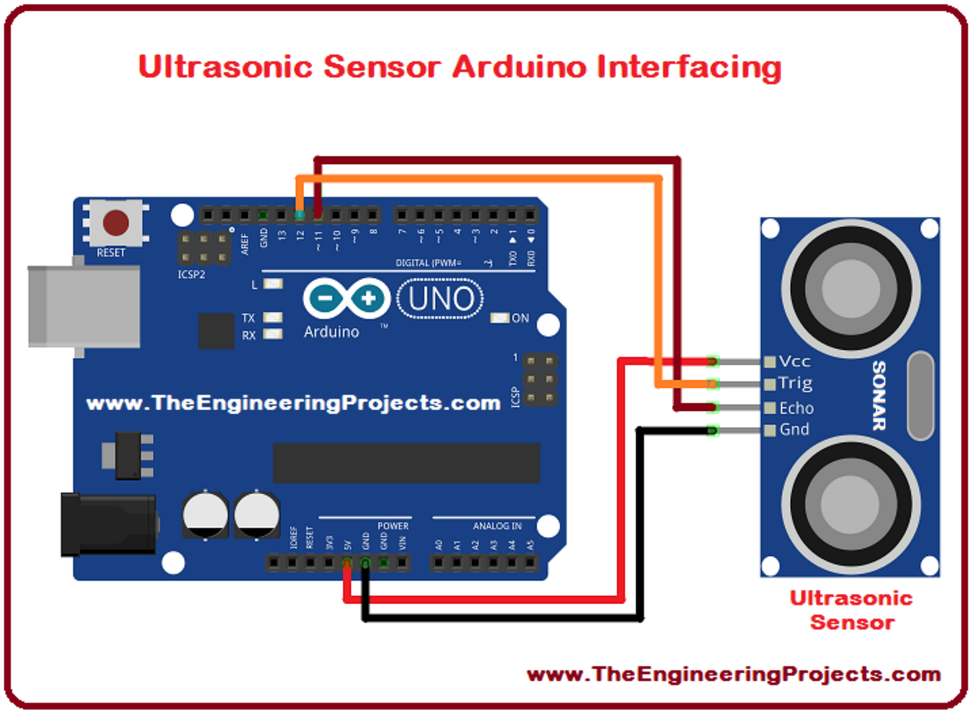
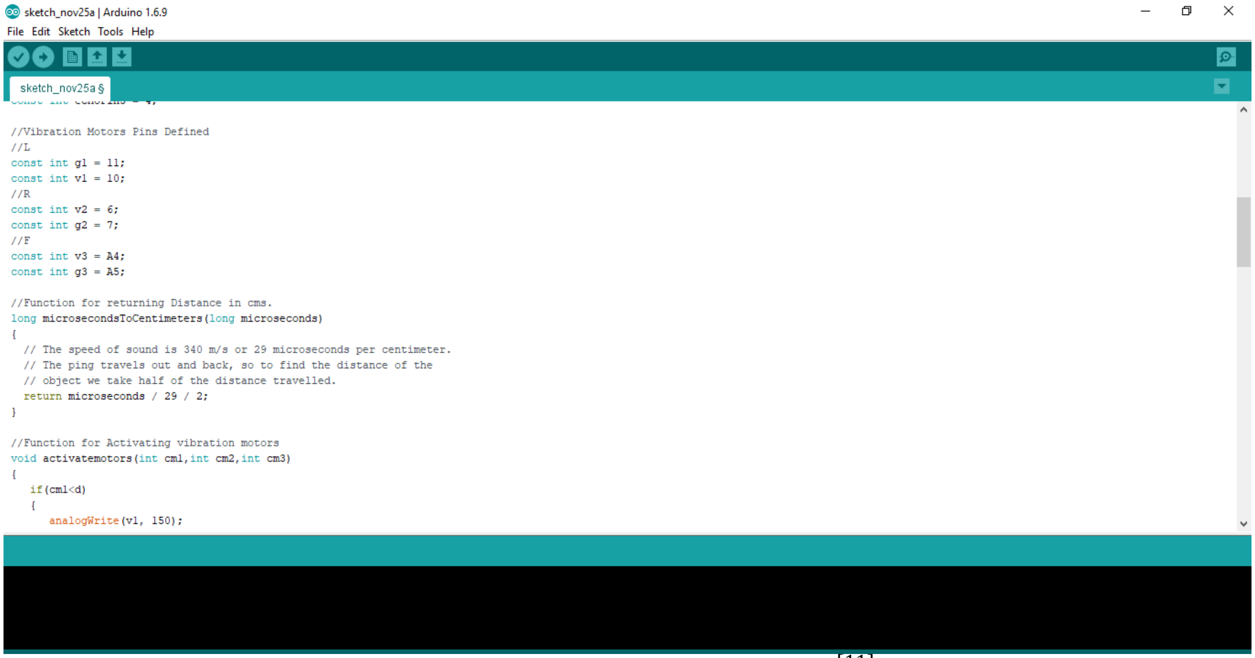

Ultrasonic Sensor Arduino Interfacing
RESET
ICSP2
DIGITAL (PWM~)
L
TX
RX
UNO
Arduino
ON
ICSP
www.TheEngineeringProjects.com
POWER
ANALOG IN
Vcc
Trig
Echo
Gnd
SONAR
Ultrasonic
Sensor
www.TheEngineeringProjects.com

Figure 4.1:Circuit Diagram of ultrasonic sensor[9]

4.1.2 Arduino UNO Coding

The coding is done on the Arduino IDE software. Ultrasonic sensors are used to measure the distance of obstacles nearby and this data is then sent to laptop for further processing.



```
//Vibration Motors Pins Defined
//L
const int g1 = 11;
const int v1 = 10;
//R
const int v2 = 6;
const int g2 = 7;
//F
const int v3 = A4;
const int g3 = A5;

//Function for returning Distance in cms.
long microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}

//Function for Activating vibration motors
void activatemotors(int cm1,int cm2,int cm3)
{
   if(cm1<d)
   {
     analogWrite(v1, 150);
```

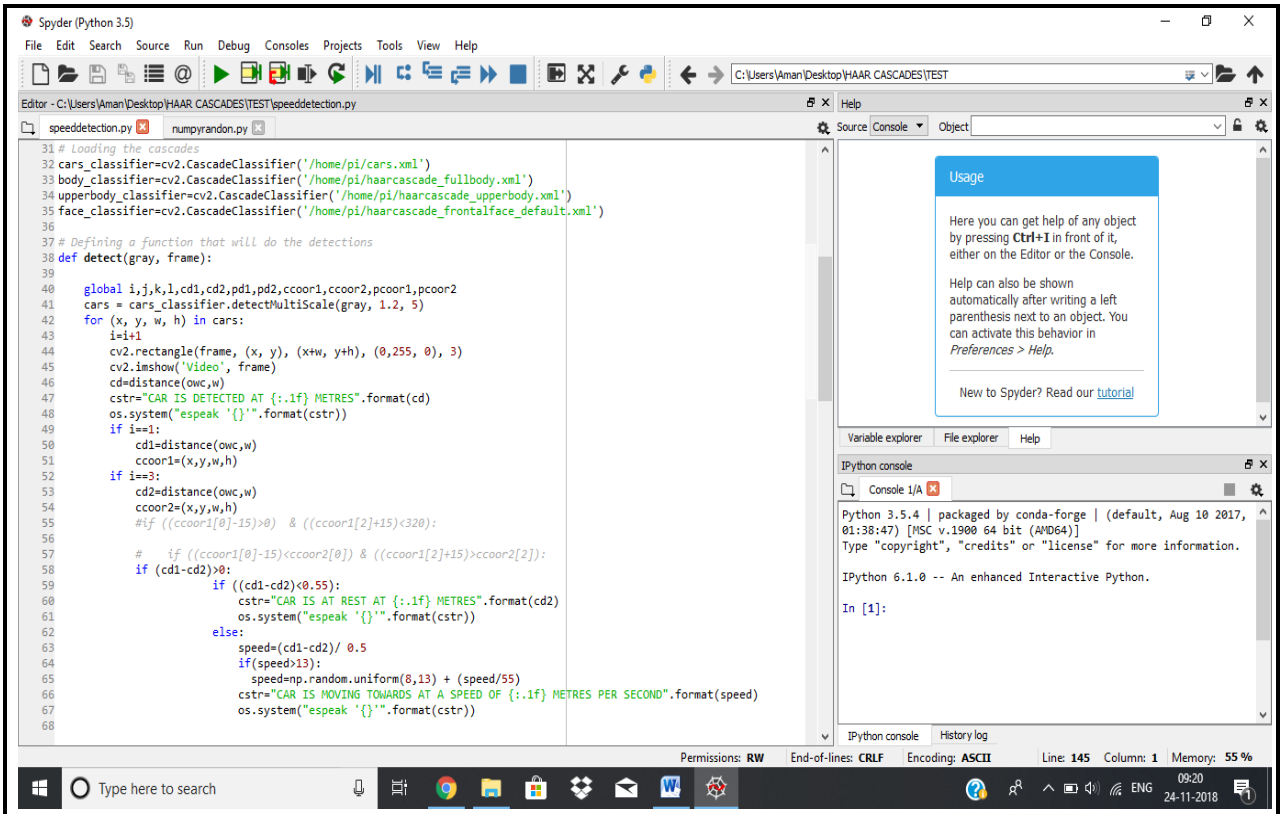
Figure 4.2: Arduino Genuino Software[11]

4.1.2 Python coding

- ❖ **ANACONDA PACKAGE (Spyder)**

  Anaconda is a free and open-source distribution of Python and R programming languages for scientific computing that aims to simplify package management and deployment. Package versions are managed by the package management system *conda*.

- ❖ **OPENCV:** OpenCV (Open Source Computer Vision Library) is released under a BSD license and hence it's free for both academic and commercial use. It has C++, Python and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. OpenCV was designed for computational efficiency and with a strong focus on real-time applications. Written in optimized C/C++, the library can take advantage of multi-core processing. Adopted all around the world, OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million. Usage ranges from interactive art, to mines inspection, stitching maps on the web or through advanced robotics. We are making use of this library to capture live stream from webcam and to pre-process the frames received before they are fed to the CNN.



Spyder (Python 3.5)
File Edit Search Source Run Debug Consoles Projects Tools View Help
C:\Users\Aman\Desktop\HAAR CASCADES\TEST
Editor - C:\Users\Aman\Desktop\HAAR CASCADES\TEST\speeddetection.py
speeddetection.py
numpyrandon.py
Help
Source Console Object
Usage
Variable explorer File explorer Help
IPython console
Console 1/A
IPython console History log
Permissions: RW End-of-lines: CRLF Encoding: ASCII Line: 145 Column: 1 Memory: 55 %
Type here to search
ENG 09:20 24-11-2018

```python
# Loading the cascades
cars_classifier=cv2.CascadeClassifier('/home/pi/cars.xml')
body_classifier=cv2.CascadeClassifier('/home/pi/haarcascade_fullbody.xml')
upperbody_classifier=cv2.CascadeClassifier('/home/pi/haarcascade_upperbody.xml')
face_classifier=cv2.CascadeClassifier('/home/pi/haarcascade_frontalface_default.xml')

# Defining a function that will do the detections
def detect(gray, frame):

    global i,j,k,l,cd1,cd2,pd1,pd2,ccoor1,ccoor2,pcoor1,pcoor2
    cars = cars_classifier.detectMultiScale(gray, 1.2, 5)
    for (x, y, w, h) in cars:
        i=i+1
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0,255, 0), 3)
        cv2.imshow('Video', frame)
        cd=distance(owc,w)
        cstr="CAR IS DETECTED AT {:.1f} METRES".format(cd)
        os.system("espeak '{}'".format(cstr))
        if i==1:
            cd1=distance(owc,w)
            ccoor1=(x,y,w,h)
        if i==3:
            cd2=distance(owc,w)
            ccoor2=(x,y,w,h)
            #if ((ccoor1[0]-15)>0)  & ((ccoor1[2]+15)<320):

            #    if ((ccoor1[0]-15)<ccoor2[0]) & ((ccoor1[2]+15)>ccoor2[2]):
            if (cd1-cd2)>0:
                        if ((cd1-cd2)<0.55):
                            cstr="CAR IS AT REST AT {:.1f} METRES".format(cd2)
                            os.system("espeak '{}'".format(cstr))
                        else:
                            speed=(cd1-cd2)/ 0.5
                            if(speed>13):
                              speed=np.random.uniform(8,13) + (speed/55)
                            cstr="CAR IS MOVING TOWARDS AT A SPEED OF {:.1f} METRES PER SECOND".format(speed)
                            os.system("espeak '{}'".format(cstr))

```

Here you can get help of any object by pressing **Ctrl+I** in front of it, either on the Editor or the Console.

Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in *Preferences > Help*.

New to Spyder? Read our tutorial

```
Python 3.5.4 | packaged by conda-forge | (default, Aug 10 2017,
01:38:47) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]:
```

Figure 4.3:Spyder Software[12]

❖ Keras

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Theano. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural-network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. In addition to standard neural networks, Keras has support for convolutional and recurrent neural networks. It supports other common utility layers like dropout, batch normalization, and pooling. It also allows use of distributed training of deep-learning models on clusters of Graphics Processing Units (GPU) and Tensor processing units (TPU).

❖ Nvidia CNN Network Architecture for Self-driving Cars

We train the weights of our network to minimize the mean-squared error between the steering command output by the network, and either the command of the human driver or the adjusted steering command for off-center and rotated images Figure below shows the network architecture[12], which 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 are designed to perform feature extraction, and are chosen empirically through a series of experiments that vary layer configurations. We then 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 final two convolutional layers.

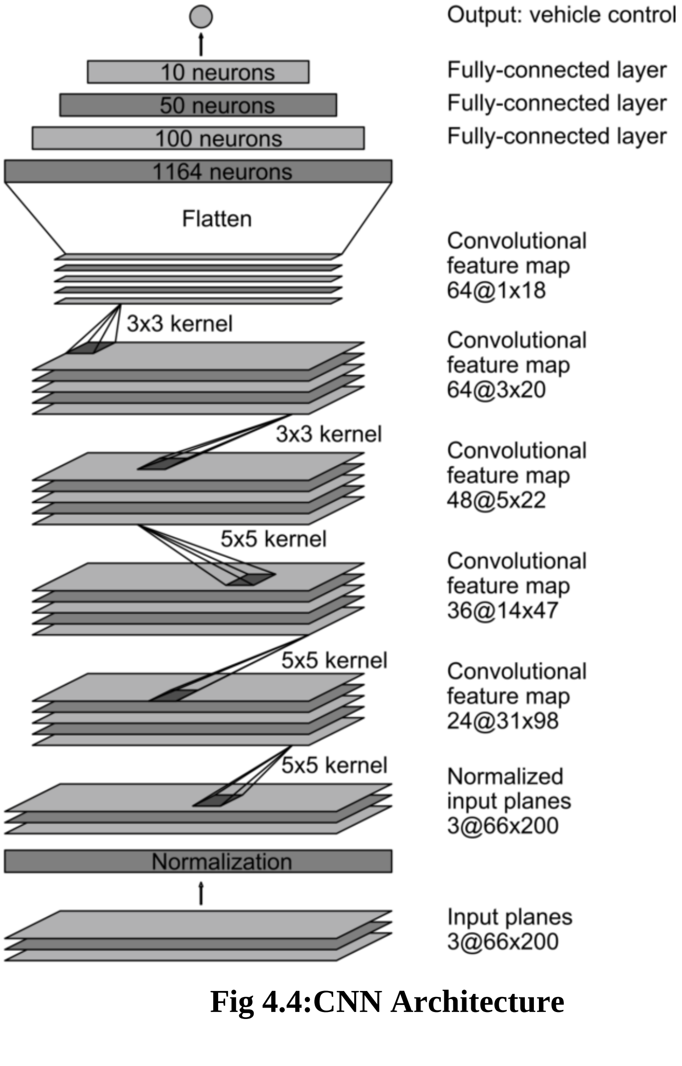


Fig 4.4:CNN Architecture

❖ Udacity Self-driving Car Simulator

This simulator was built for Udacity's Self-Driving Car[13] Nanodegree, to teach students how to train cars how to navigate road courses using deep learning. This is a simulator where we can steer a car around a track for data collection. We use image data and steering angles to train a neural network and then use this model to drive the car autonomously around the track.

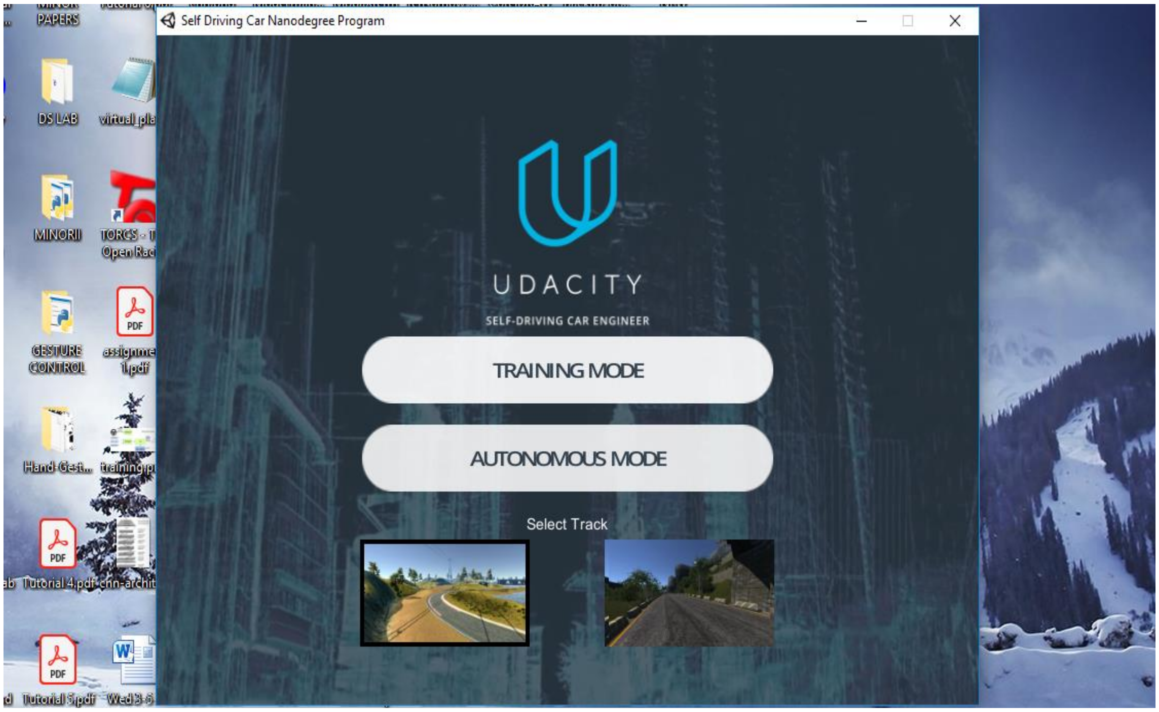


Fig 4.5: Udacity Simulator

❖ **HAAR CASCADES:** A Haar-Cascade is basically a classifier which is used to detect the object for which it has been trained for, from the source. The Haar-Cascade is by superimposing the positive image over a set of negative images. The training is generally done on a server and in various stages. Better results are obtained by using high quality images and increasing the amount of stages for which the classifier is trained.

CHAPTER 5

IMPLEMENTATION

This chapter explains how the system has been implemented, and the decisions that were made during the process. It also shows any difficulties that were encountered, and how they were dealt with.

5.1 Working

The project is intended to build a laptop as central processing unit based project to demonstrate a working model of a Deep Neural network (DNNs) based car that is able to control its acceleration and steering angle according to the path and also able to detect various obstacles that come its way. It will be able to carry out certain actions in order to prevent collisions with these objects. The major emphasis of this project is to make a model with good accuracy that provides reliable output.

The car has a camera and obstacle detecting sensors mounted on the car. Camera captures images and provides it to the Convolutional Neural Network (CNNs) that processes the frame (image) in real time and produces steering angle and acceleration as the output. These output in turn controls the locomotive motors to move the car in desired manner. The obstacle detecting sensors (ultrasonic) prevent the car from hitting the obstacles by providing timely feedback to change its action.

5.2 Advantages

- ***Safety*****:** self-driving cars can reduce traffic accidents, which means there will be fewer deaths and injuries caused by these accidents.
- ***Save Money*****:** as the number of accidents is reduced, less people will be injured and fewer vehicles will be damaged. As a result, insurance rates will also drop.
- ***Independence*****:** at some point, the elderly can no longer drive their own car. However, if they own a self-driving car, they can still go to places without having a driver.
- ***Less Traffic*****:** self-driving cars will not be caught in traffic jams.

5.3 Autonomous Car Software Simulation Realization

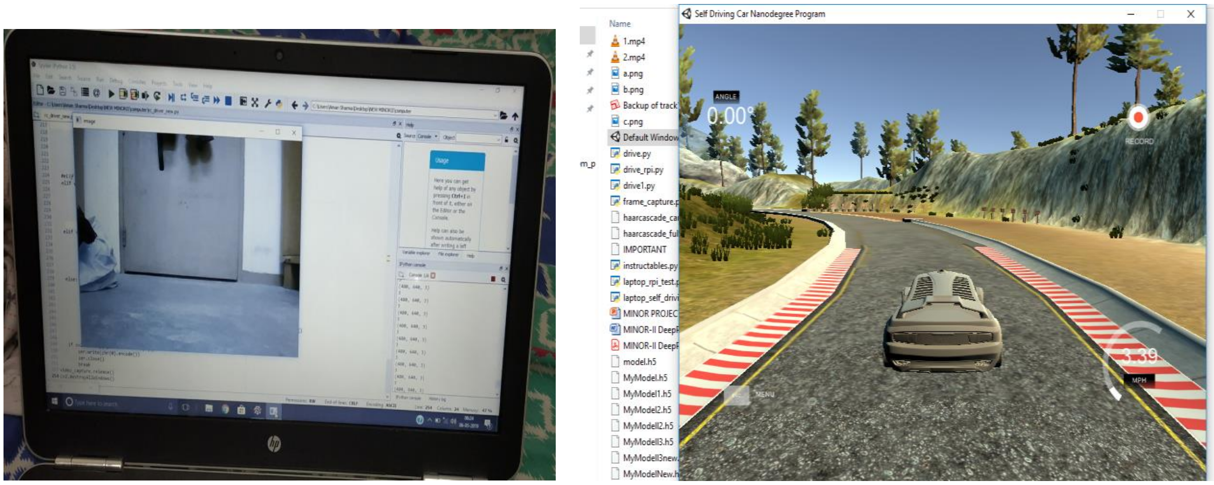


Figure 5.1: Simulation realization of car

5.4 Autonomous Car Realization

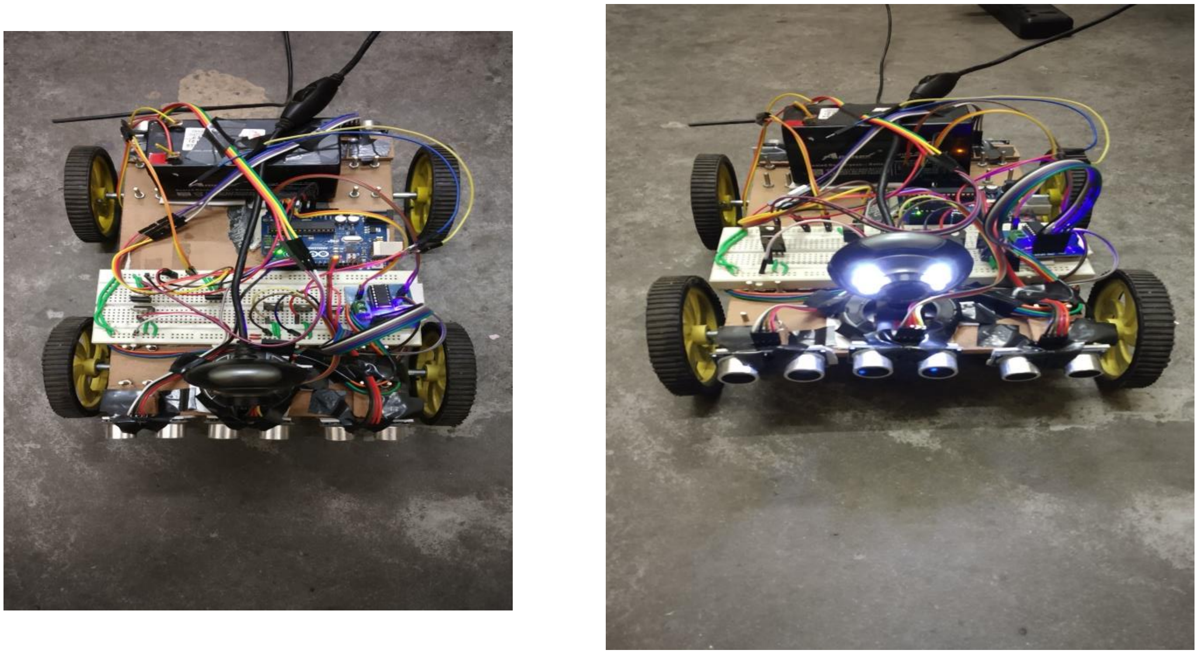


Figure 5.2: Hardware realization of car module

CHAPTER 6
CONCLUSION

The main aim of the project is to make the traffic accident free. In the project described how to use deep convolutional networks to control a prototype of a simplified self-driving car. The simulation was done on Unity by Udacity, we designed a multithreaded system that controls it and provided a light web and command line interface which users can use to interact with the car. Lastly, We equipped the car with autopilot controller based on state-of-the-art convolutional neural network architecture.

The system learned to steer autonomously when running on fixed speed. It can drive on simplified flat roads without traffic only using camera as its input. In the end, the car was able to drive indefinitely on a semi-circular track. The biggest challenges I had to face were the functioning of R-pi due to heavy processing which was replaced by laptop in future.

CHAPTER 7
FUTURE SCOPE

Even though the car works well and does not drive off track, there is still a lot of room for improvement.

- The path it sometimes takes is not ideal. That is caused mainly by the implicit noise in the training data, because I was unable to collect a dataset with optimal paths. One option would be to create a driving simulator and implement path planning algorithms to gather a better driving dataset. That would also simplify the mapping between input images and steering angles, allowing for smaller networks with fewer parameters. Another option might be to use evolution to force the car to take better paths resulting in an increased average speed.

- To make the system more robust, a much larger dataset of different track types and different road conditions would need to be collected. This could be an interesting use-case for generative adversarial networks.

- Change in track with the help of ultrasonic sensors, which detect the car speed and distance and change the lane accordingly.

REFERENCES

[1]Chenyi Chen et al. “Deepdriving: Learning affordance for direct perception in autonomous driving”. In: Proceedings of the IEEE International Conference on Computer Vision. 2015, pp. 2722–2730.(Visited : 14/01/2019)

[2] U. D. Automated Driving Systems - A Vision for Safety. September 2017, [Online; posted September-2017]. Available from : https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/documents/13069a-ads2.0_090617_v9a_tag.pdf (Visited : 14/01/2019)

[3] LLC, W. Waymo Self-Driving Car Project. 2017, [Online]. Available from: https://waymo.com (Visited : 28/01/2019)

[4]Meet ALVINN, the self-driving car from 1989. Available from : https://www.theverge.com/2016/11/27/13752344/alvinn-self-driving-car-1989-cmu-navlab (Visited : 25/02/2019)

[5] ”Arduino Schematic” - https://i1.wp.com/eeeproject.com/wp-content/uploads/2017/06/Arduino-Uno-board-pins-description.jpg?w=629&ssl=1 (Visited: 12/03/2019)

[6] “IC-7805 Image” - https://roboticsdna.in/102-large_default/7805-ic.jpg (Visited : 12/03/2019)

[7] “Ultrasonic Sensor Image” https://www.makerlab-electronics.com/my_uploads/2016/05/ultrasonic-sensor-HCSR041.jpg (Visited : 12/03/2019)

[8] “DC Motor Image” https://www.google.com/search?q=dc+motor&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi3iPGDp4XiAhXBN48KHZ_cAqsQ_AUIDigB&biw=1366&bih=608#imgrc=XC-Xx3NwD_q2YM (Visited : 23/04/2019)

[9]”Circuit diagram to depict working of Car” https://i1.wp.com/www.studentsheart.com/wp-content/uploads/2018/05/hc-sr04-ultrasonic-sensor-arduino-wiring-diagram-1024x529.png?fit=1024%2C529 (Visited : 30/04/2019)

[10]”Arduino Software” https://www.arduino.cc/en/main/software (Visited : 23/03/2019)

[11] Bojarski, M.; Del Testa, D.; et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016. (Visited : 16/04/2019)

[12] Udacity, Inc. Udacity’s Self-Driving Car Simulator. URL: https : //github . com/udacity/self-driving-car-sim (visited on 07/02/2019).