A

Project Report On

Autonomous vehicle using Raspberry pi and Deep learning

In Partial Fulfilment of Degree for Bachelor of Engineering in Electronics and Communication Engineering
Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur.

Submitted by

Avinash Kumbhare
Ritik Datey
Pranay Khaire

Guided By

Prof. S. A. Chaturvedi

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG
Priyadarshini Institute of Engineering & Technology,
Hingna Road, Nagpur (M.S.).

(Session 2020 – 21)

A

Project Report On

Autonomous vehicle using Raspberry pi and Deep learning

In Partial Fulfilment of Degree for Bachelor of Engineering in

Electronics and Communication Engineering

Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur.

Submitted by

Avinash Kumbhare
Ritik Datey
Pranay Khaire

Guided By

Prof. S. A. Chaturvedi

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG
Priyadarshini Institute of Engineering & Technology,
Hingna Road, Nagpur (M.S.).

(Session 2020 – 21)

DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG
Priyadarshini Institute Of Engineering & Technology,
Hingna Road, Nagpur

CERTIFICATE

This is to certify that this project contains the bonafide work of **Mr.Avinash Kumbhare, Mr. Ritik Datey, Mr. Pranay Khaire** during the academic session 2020 – 2021 in partial fulfilment of the requirements for the award of Bachelor degree of Engineering in DEPARTMENT OF ELECTRONICS & COMMUNICATION ENGG of the Rashtrasant Tukdoji Maharaj Nagpur University, Nagpur.

Date: _____

Prof. S. A. Chaturvedi
Head, E & C Engg, Project Guide

Dr. G. M. Asutkar
Vice Principal

Dr. V. M. Nanoti
Principal

ACKNOWLEDGEMENT

The architecture of success stands on strong foundation made up of strenuous hard work, determination, presence of mind and above all timely advice and guidance from the learned and experienced people. It is our esteemed pleasure to present the completion of our project "**Autonomous vehicle using Raspberry pi and Deep learning**".

Stressing on these words we would like to express our deep sense of gratitude to **Prof. S. A. Chaturvedi** our respected **Project Guide** under whom our project has seen the light of success. He/She has been a constant source of inspiration to carry us smoothly throughout the project. It has been privilege and pleasure working under his/her guidance and has been an experience, which we shall cherish for a long time.

It is our pleasure to thank **Prof. S. A. Chaturvedi,** for her esteemed guidance, constant encouragement and valuable advice during the course of project

We are thankful to **Dr. V.M.Nanoti, Principal** and **Dr. G. M. Asutkar**, **Vice-Principal,** with their efficient management and good administration gave us the environment conductive to the development of our project. We are especially thankful to the librarians for making us available necessary books and reference materials at all required hours.

Last but not the least; we are thankful to all the staff members of the college who directly or indirectly helped us for the successful completion of the project.

Project Team:-

- Avinash Kumbhare
- Ritik Datey
- Pranay Khaire

INDEX

CHAPTER 01

ABSTRACT

ABSTRACT

The self-driving vehicle or driverless vehicle can be alluded to as an autonomous vehicle in basic language. This vehicle is equipped for detecting the environment and fulfilling human transportation without any human input. It's anything but a major advance in propelling future innovation. Autonomous vehicles can sense their environmental factors with cameras, radar, lidar and GPS. Advanced control systems can interpret and decipher the data to monitor their position despite the fact that the conditions change. Among the benefits of autonomous vehicles include less car accidents, better reliability, and reduced traffic as well as insurance costs.

Organizations, such as Google, Volvo, Mercedes-Benz and Audi are the front sprinters in making the autonomous vehicle a reality.

The turn of events and extension of the area in Indian conditions is likewise worth considering.

Self-driving cars are autonomous vehicles that can drive by themselves without any human interference and has the potential to mark the technological revolution of the next decade. This work presents the development of a low-cost prototype of a miniature self-driving car model using simple and easily available technologies. The objective of the work is to avoid accidents caused due to driver faults. In this prototype, Raspberry Pi controller and H-bridge drives two DC motors to realize vehicle automation. Technologies such as GPS to access location, sonar sensors for obstacle detection and avoidance, image processing for pedestrian detection, computer vision for processing images and machine learning for intelligent systems have been deployed.

CHAPTER 02

INTRODUCTION

INTRODUCTION

Rushing around, trying to get errands done, thinking about the things to be bought from the nearest grocery store has become a part of our daily schedule. Driver error is one of the most common causes of traffic accidents, and with cell phones, in car entertainment systems, more traffic and more complicated road systems, it isn't likely to go away. With the number of accidents increasing day by day, it has become important to take over the human errors and help mankind. All of this could come to an end with self-driving cars which just need to know the destination and then let the passengers continue with their work. This will avoid not only accidents but also bring a self-relief for minor day to day driving activities for small items.

Vehicle automation was originally envisioned as early as in 1918. The initial phase of research and development was jointly initiated by General motors and Radio Corporation of America. From 1964 to 2003, several other R&D programs were operational in the US,Europe and Japan to develop automated bus and truck platoons, video image processing for driving .

Driving requires a variety of functions, including localization, perception, planning, control, and management. Information acquisition is prerequisite to localization and perception. If all these functions are available in a vehicle, then it could be termed as an autonomous vehicle.

With the growing needs of convenience, technology now tries to seek automation in every aspect possible. Here we propose an autonomous vehicle which would eradicate human intervention in the field of driving. The car would be able to detect track, avoid obstacles and detect traffic signs.

This would ensure safer, easier, updated and more convenient mobility, hence proving to be a revolutionary step in the field of automation

CHAPTER 03

LITERATURE SURVEY

- **Design & implementation of Autonomous car using Raspberry pi by Gurjashan Pannu. (INTERNATIONAL JOURNAL OF COMPUTER APPLICATION) (researchgate.net) March 2015**
  - In this paper, a method to make a self driving robot car is presented. The different hardware components and their assembly is clearly described. A novel method to determine the uneven, marked or unmarked road edges is explained in details relying upon OpenCV. Using ultrasonic sensors, the collisions with obstacles are avoided. The algorithm mentioned in the paper has been successfully implemented on a small autonomous car.

- **Real time lane detection and motion planning in raspberry pi & arduino for an autonomous vehicle prototype by alfa rossi (arxiv.org) 20 Sep 2020.**
  This paper discusses a vehicle prototype that recognizes streets' lanes and plans its motion accordingly without any human input. Pi Camera 1.3 captures real-time video, which is then processed by Raspberry-Pi 3.0 Model B. The image processing algorithms are written in Python 3.7.4 with OpenCV 4.2. Arduino Uno is utilized to control the PID algorithm that controls the motor controller, which in turn controls the wheels. Algorithms that are used to detect the lanes are the Canny edge detection algorithm and Hough transformation. Elementary algebra is used to draw the detected lanes. After detection, the lanes are tracked using the Kalman filter prediction method. Then the midpoint of the two lanes is found, which is the initial steering direction. This initial steering direction is further smoothed by using the Past Accumulation Average Method and Kalman Filter Prediction Method. The prototype was tested in a controlled environment in real-time. Results from comprehensive testing suggest that this prototype can detect road lanes and plan its motion successfully.
- **Raspberry pi 3b model data sheet, available at https://www.raspberrypi.org/**
  The Raspberry Pi is a credit card-sized single-board computer. There are currently five Raspberry Pi models in market i.e. the Model B+, the Model A+, the Model B, the Model A, and the Compute Module (currently only available as part of the Compute Module development kit). All models use the same SoC (System on Chip - combined CPU & GPU), the BCM2835, but other hardware features differ.
  The A and B use the same PCB, whilst the B+ and A+ are a new design but of very similar form factor [17].The Compute Module is an entirely different form factor and cannot be used standalone.
  In this project, we have used the model B Rev 2. It comprises of a 512 MB RAM model with two USB ports and a 10/100 Ethernet controller [17]

- **Pi camera module data sheet, available at https://cdn.sparkfun.com/datasheets/**
  It is the camera shipped along with Raspberry Pi [. Pi camera module is also available to which can be used to take high-definition videos as well as still photographs.
- **For image processing, Opencv.org**
  It (Open Source Computer Vision) is a library of programming functions mainly aimed at real-time computer vision.
  It has over 2500optimized algorithms, including both a set of classical algorithms and the state of the art algorithms in Computer Vision, which can be used for image processing, detection and face recognition, object identification, classification actions, traces, and other functions [21]. This library allows these features be implemented on computers with relative ease, provide a simple computer vision infrastructure to prototype quickly sophisticated applications [20, 21].
  The library is used extensively by companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota, and startups are Applied Minds, Video Surf and Zeitera. It is also used by many research groups and the government [21].
  It is based on C++ but wrappers are available in python as well. In our project is used to detect the roads and guide the car on unknown roads [21].

CHAPTER 04

DESIGN OF PROPOSED WORK

DESIGN OF PROPOSED WORK

Here we have Raspi Cam v2 that sends video streams to Raspberry pi 3b as shown in fig.1. Then the captured stream is processed frame by frame to detect the lanes on the track. The Raspberry pi sends instructions accordingly to Arduino to drive the vehicle on track.
When a car is placed on track then it starts to move and stays in the middle of the lane until it reaches the stop sign or if it detects lane turn then it makes a turn depending on track.

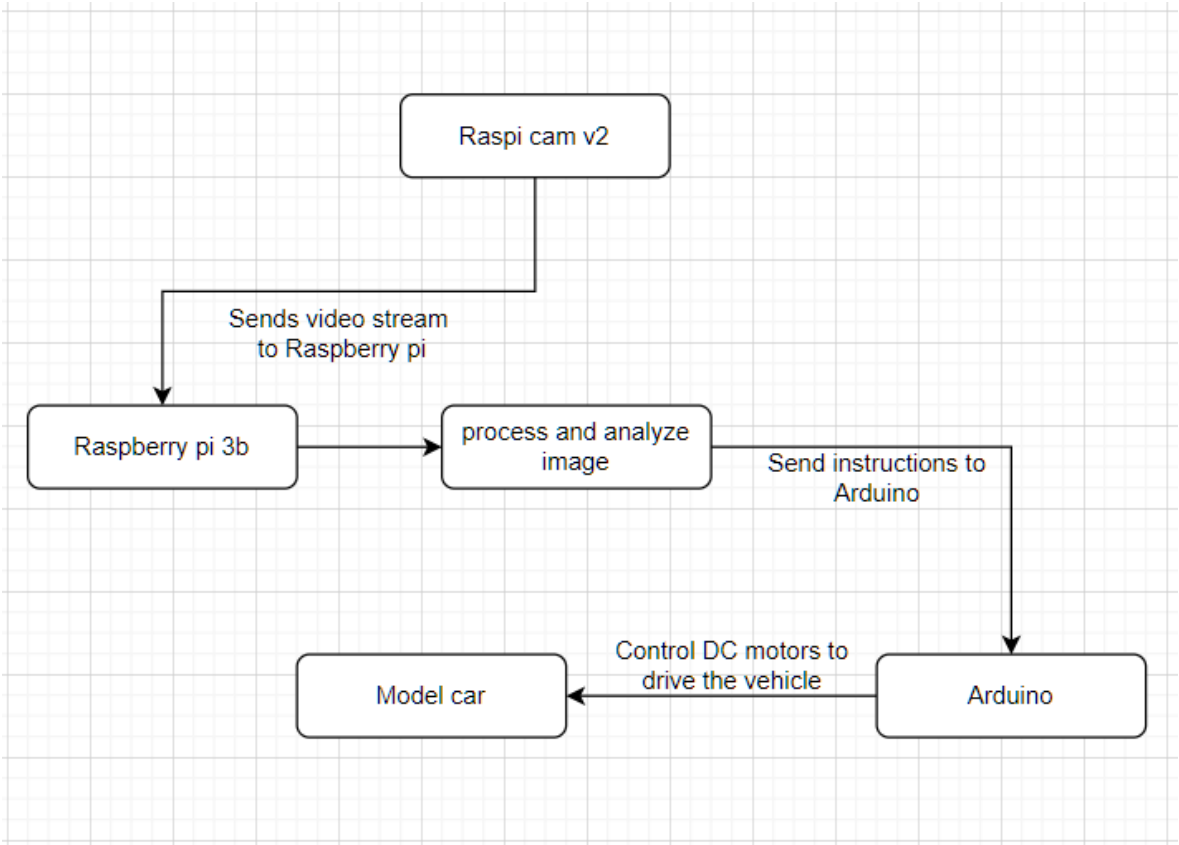

Fig. 1 Design of proposed work

We are using raspberry pi 3b because it is capable of processing upto 10 frames per second, which is sufficient for our project. The raspicam captures the video stream at 360p resolution, the track and lanes are visible at this resolution. The Raspberry pi should be able to send instructions to the arduino. The arduino sends instructions to the L298N motor driver via EN pins of the motor driver. The 12v adapter is used to provide power to the motor driver. There is a 2.5v voltage drop across the output pins of the driver, so we are using a 12v supply, the DC motors need around 10v power.

Assemble robot body by assembling the chassis and wheels of robot car, solder the motors and fix them in the chassis, fix raspberry pi, Arduino Uno, motor driver and make the circuit connections as per circuit diagram. Prepare the sample track for the robot to run. Bundle operating system to raspberry pi along with the Raspbian OS and it to a 16GB Micro SD card and put it to the raspberry pi. Using ethernet cable connect the raspberry pi to the computer and enable the Wi-Fi function of pi so that it can connect and access the Pi using Wi-Fi and VNC viewer which is installed on pi OS. Install open CV library in raspberry pi and access the pi using your pc and Wi-Fi and use the commands in the command terminal of raspberry pi to install OpenCV library. Link the open cv library to Geaney editor (C++ compiler software of Raspberry Pi).
Setup Raspberry pi camera and now adjust the camera position for best view and fix it on the robot car.
1. Process the camera output (video) in the open cv library.
2. video out of camera and set a region of interest (the portion of video that we want to process to find the lanes of road).
3. Convert region of interest to perspective warp (bird eye view).
4. Threshold operation- Use threshold operation to the color image to black and white image in order we can find the lanes.
5. Find the tracks using Canny edge detection.
6. Find lanes a from track and detect left and right side deviation of track.
7. Lane end detection.
8. Stop sign detection- We capture positive samples and negative samples of Stop sign and load it in the Cascade trainer software the output of this software is used and the trainer data for our machine learning algorithm.
9. Obstacle detection and avoidance.
10. The same method is used to detect traffic lights.
11. Control the Arduino. From the above steps the proposed system can find the left and right deviation of track now we need to run the robot as per the conditions.
12. If the robot is in the center of the track go Forward.
13. If the track is deviated to the left, steer to Left.
14. If the track is deviated to right steer to Right.
15. If the lane end is detected, take a U turn.
16. If an obstacle is detected follow stop, left, forward, right, then forward.
17. If the traffic signal is red Stop.
18. If the traffic signal is green, Forward.
19. We use 4 GPIO pins of raspberry pi to send decimal values to Arduino in the form of binary data. In the Arduino these binary data‟s are received through its I/O pins and convert it to decimal and perform the movement operations.

CHAPTER 05

HARDWARE & SOFTWARE REQUIREMENTS

Hardware :

- **Raspberry pi 3b :**

  The Raspberry pi is a low cost, credit-card sized computer. It is a capable device which has enough compute power to drive our autonomous vehicle. It has Quad core 1.2GHz 64bit CPU with 1GB RAM and wireless LAN.
  Raspberry pi 3b can stream and process around 10 fps, which is enough to drive the vehicle on track.

Fig. 2 Raspberry pi 3B

Raspberry pi 3b Specification :

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

- **Arduino UNO :**

  The Arduino Uno is a microcontroller board based on the ATmega328. It has 20 digital input/output pins (of which 6 can be used as PWM outputs and 6 can be used as analog inputs), a 16 MHz resonator, a USB connection, a power jack, an in-circuit system programming (ICSP) header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Uno differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features an ATmega16U2 programmed as a USB-to-serial converter.

Fig.3 Arduino UNO

We are using Arduino UNO as a slave device which takes instructions from raspberry pi (main device) and sends the required instruction to the L298N motor driver to drive DC motors.

Specification of Arduino uno :

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Inout Voltage (limit): 6-20V
- Digital I/O Pins: 14 (of which 6 provide PWM output)
- PWM Digital I/O Pins: 6
- Analog Input Pins: 6
- DC Current per I/O Pin: 20 mA
- DC current for 3.3V Pin: 50 mA

- **L298N H-Bridge motor driver :**

  The L298N is a dual H-bridge motor driver which allows speed and direction control of DC motors at the same time. The module can drive DC motors that have voltages between 5 to 35V with peak current up to 2A.

  Let's take a closer look at the pinout of L298N module and explain how it works. The module has two screw terminal blocks for the motor A and B, and another screw terminal block for the Ground pin, the VCC for motor and a 5V pin which can either be an input or output.

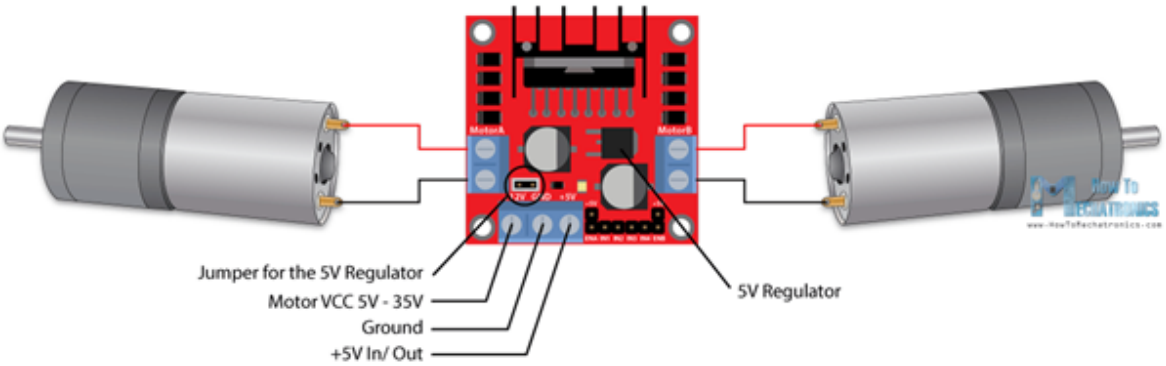

Fig. 4 L298N H-Bridge

- **RaspiCam v2 :**

  The v2 camera module has a 5MP sensor. The camera module is used to take video streams as well as still photos. The camera works well with Raspberry Pi 3b.
  To increase the image processing speed, we will be capturing video streams at 360p resolution only.

Fig. 5 Raspi Cam v4

- **Car chassis kit :**

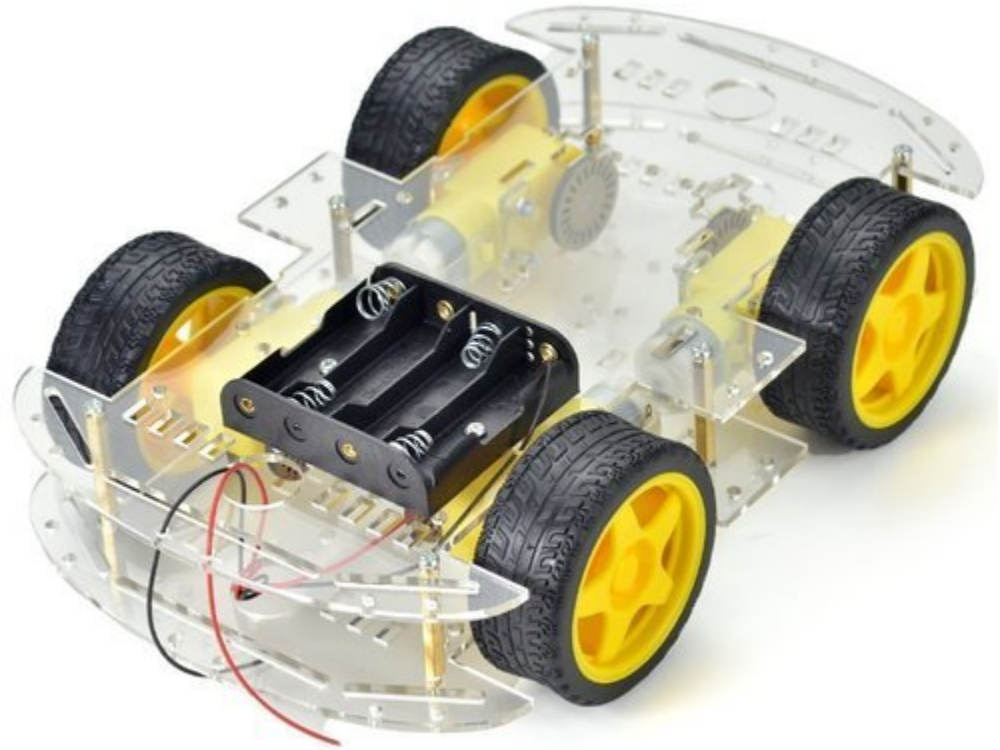

Fig. 6 Car chassis kit

- **12V Power adapter**
- **9V batteries**
- **Jumper wires**

Software :

- **Raspbian OS** :
  Raspberry Pi os (Raspbian) is a Debian-based operating system for Raspberry Pi. Since 2015, it has been officially provided by the Raspberry Pi Foundation as the primary operating system for the Raspberry pi family of compact single-board computers.
- **OpenCV** :
  OpenCV (Open source Computer Vision Library) is an open source computer vision and machine learning software library. The library has more than 2500 optimized algorithms, which includes a comprehensive set of machine learning algorithms. These algorithms can be used to detect objects, classify images and recognize what's in the image.
- **Arduino IDE :**
  The open source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on windows, mac, and linux.
- **Programming languages used :**
  C++, Python.

CHAPTER 06

IMPLEMENTATION OF PROPOSED WORK

Implementation :

The following figures show the working model of our autonomous vehicle.

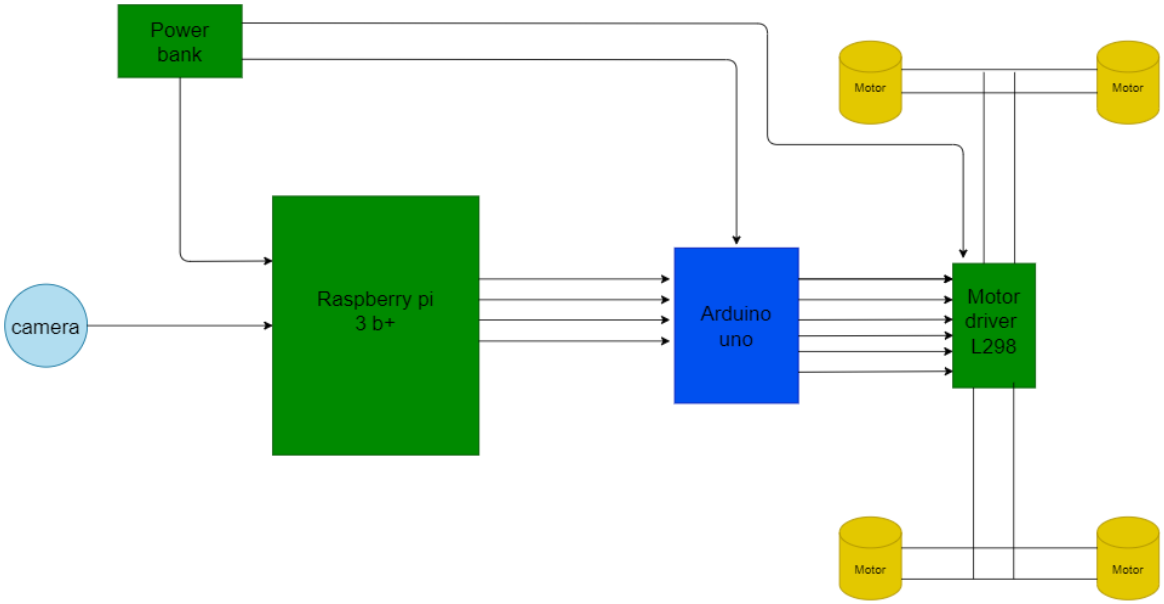

fig. 7 Block diagram

Our prototype model shows some work that we have discussed. Our main focus was to build a miniature version of an autonomous vehicle which finds lanes on track and identifies the environment and follows the lanes.
First we implemented the camera sensors which can detect the surrounding environment and other objects. Then we connected the required hardware components. RaspiCam connected to Raspberry pi via CSE cable, and arduino connected to 4 pins of Raspberry pi via jumper wires. The motor driver (L298N) connected to the arduino PWM pins (3,4,5) and (9,10,11).
The 4 wheels of the chassis are connected to 4 separate motors. The motor driver IC L293D is capable of driving 2 motors simultaneously [22]. The rotation of the wheels is synchronized on the basis of the sides i.e. the left front and left back wheels rotate in sync and right front and right back wheels rotate in sync. Thus the pair of motors on each side is given the same digital input from L293D at any moment. This helps the car in forward, backward movements when both side wheels rotate in the same direction with same speed. The car turns when the left side wheels rotate in opposite direction to those in right [22]. The chassis has two shelves over the wheels separated by 2 inch approx. The IC is fixed on the lower shelf with the help of two 0.5 inch screws. It is permanently connected to the motor wires and necessary jumper wires are drawn from L293D to connect to Raspberry Pi.
.

Truth table to control the Motor :

Pin 1	Pin 2	Pin 7	Function
High	High	Low	Anti-clockwise
High	Low	High	Clockwise
High	High	High	Stop
High	Low	Low	Stop
Low	X	X	Stop

High +5V, Low 0V, X=either high or low

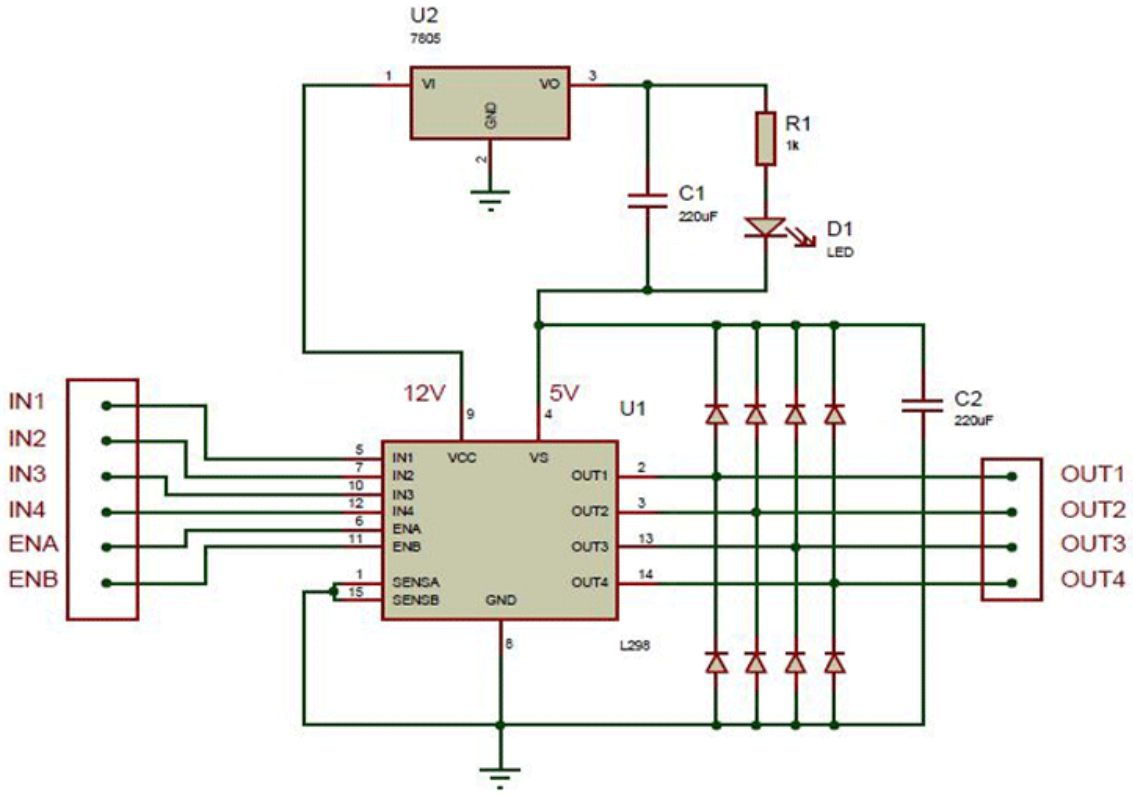

Fig 8. L298n motor driver

The rest of the space on the lower shelf is taken by 8 AA batteries which provide the power to run the motors. To control the motor connected to pin 3 (O1), pin 6 (O2), the pins used are pin 1, pin 2 and pin 7 which are connected to the GPIOs of Raspberry pi via jumper wires.

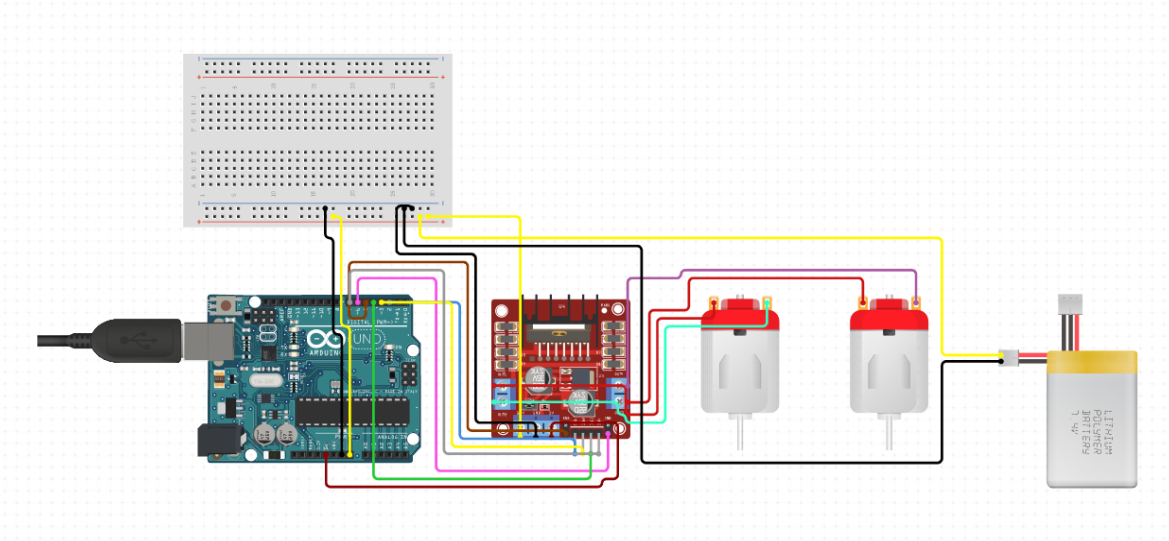

Fig. 9 Circuit diagram

Hardware Connections :

- First RaspiCam is connected to Raspberry pi via CSE cable.
- The arduino is connected to Raspberry pi via 4 gpio pins (21,22,23,24). The instruction will be a decimal number. for eg. [(0010 (1 in binary) to arduino] it means the vehicle has to move forward. There are 9 such instructions we programmed, which will be discussed later.
- The arduino sends instructions to the L298N motor driver. L298N has 6 pins, 2 for enabling and disabling motors, 2 for speed control, 2 for direction control. The 6 PWM pins (3,4,5) and (9,10,11) from the arduino are connected to L298N.
- 4 DC motors are connected to output pins of the L298N.

The raspberry pi case is glued on the top shelf along with the L shaped aluminum strip. The pi is fit in the case and the aluminum strip gives the support to the camera fit on the servo motor. The Wi-Fi dongle is attached to the USB port in Raspberry Pi in order to connect to it wirelessly. The complete connection of the raspberry pi with motor controller L293D can be found in fig.. Since raspberry pi needs its own IP, it needs to be connected to a Wi-Fi router or Hotspot [9]. For the same we need to make some changes in the field specified so as to make raspberry pi recognize the router every time it boots up.

Working Model :

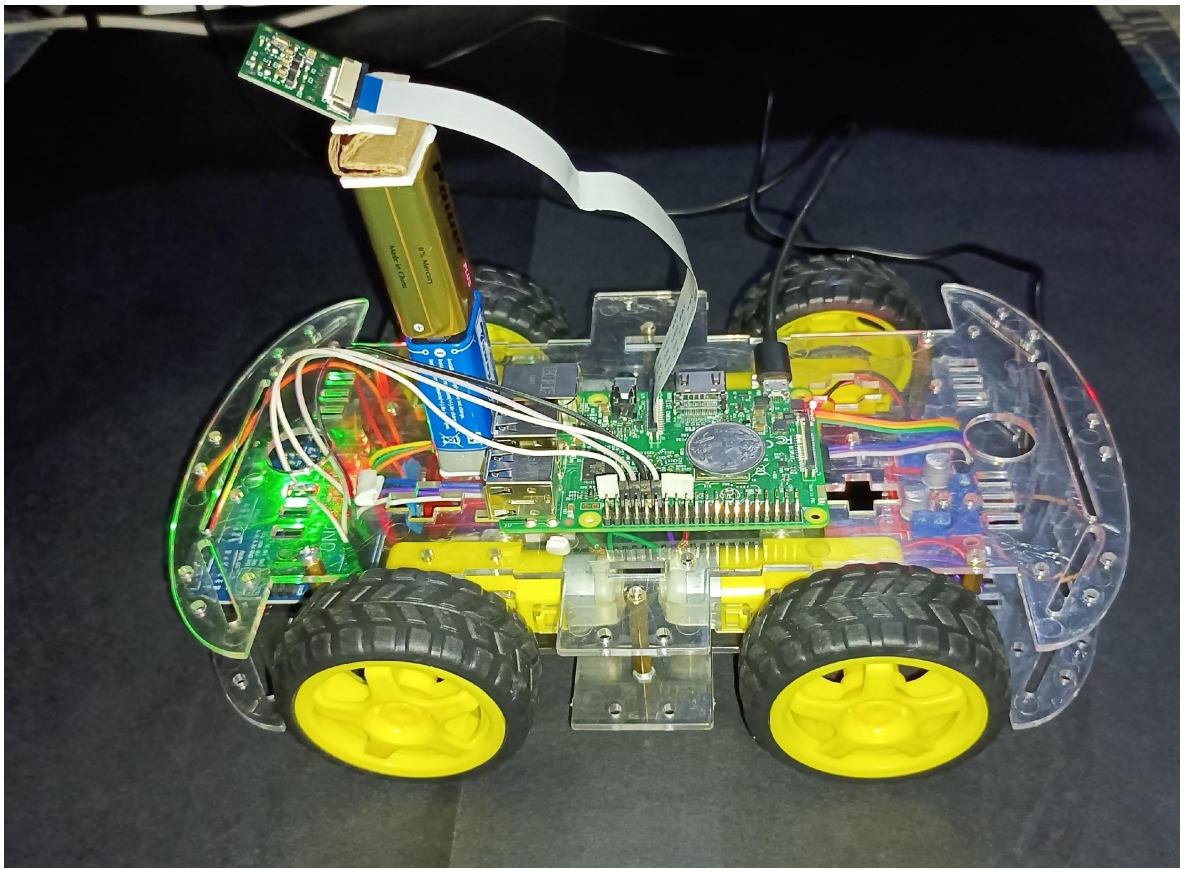

Fig. 10 Working model

The Raspberry-pi is the main core controller which will be attached to the vehicle. The module of pi-camera will be attached on the top of the prototype. The LN298 IC is used for the movement of the prototype. Computer Vision is utilized to find objects in front of the prototype and take necessary movements. When there is any object in front of the prototype and is in a measurable distance from the prototype. The raspberry-pi gives instructions to the Arduino UNO and it gives orders to the L298N Integrated Circuits to halt giving power to the tyres and therefore halts the movement of the prototype contingent on the nearness of the object. The displacement calculated is also shown on the output window of the program. The following process is finding lanes and road traffic signs.

CHAPTER 07

TESTING & RESULTS

Testing :

Once all the components are connected, the camera starts capturing images. We place the car on track. The camera starts sending video streams to raspberry pi for processing.

Using OpenCV, the images are first converted into grayscale, since we don't need any RGB color to detect lanes and also we want to optimize image processing time. After converting to grayscale, using OpenCV again we then process the image to maximize white and black, since the lanes are white colored and the rest of the track is black, it will be easier to detect lanes on track. Then we perform some operations to detect the lanes, first Perspective transformation, drawing lines on track by finding whitest pixels in the image.
The following image shows the result of OpenCV operation on video stream.

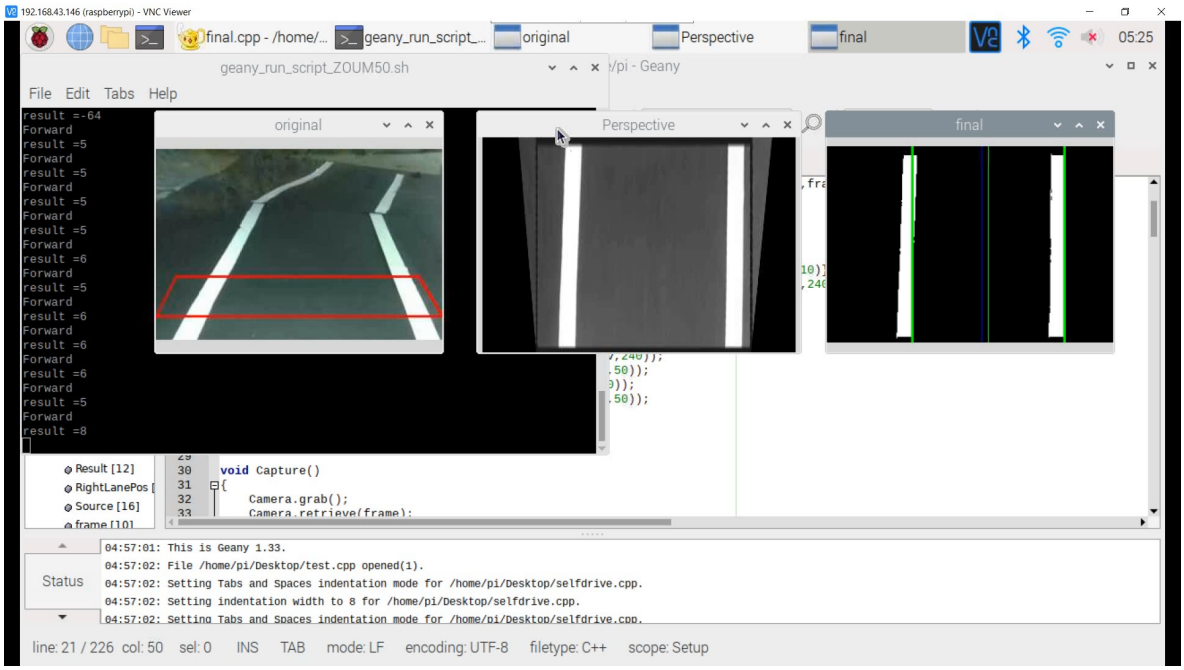

Fig 11. Image processing results

Then we draw two important lines which will determine whether the vehicle is in the middle of the lane or not. The first line will be the center of the frame captured by the camera, and the second line will be the center of the two lane lines drawn by OpenCV.
Now if the vehicle drifts to the left side of the lane, then there will be a negative difference between first and second line. Then the raspberry pi will send this negative difference to the arduino and the arduino will adjust the speed and direction of motors to make this difference zero.
Positive difference means the vehicle is drifted to the right side of the lane. Zero difference between two lines means the vehicle is perfectly in the middle of the lane, this is our objective.

Lane Detection :

Traditionally, lanes could be detected by two approaches namely feature based technique and model based technique. The feature based technique localizes the lanes in the road images by combining the low-level features, such as painted lines or lane edges etc . Accordingly, this technique requires well studied roads having well-painted lines or strong lane edges, otherwise it will fail. Moreover, it has the disadvantage of not imposing any global constraints on the lane edge shapes, this technique may suffer from occlusion or noise. On the other hand, the model-based technique just uses a few parameters to represent the lanes. Assuming the shapes of lanes can be presented by either straight line or parabolic curve, the processing of detecting lanes is approached as the processing of calculating those model parameters. This way, the model-based technique is much more robust against noise and missing data, compared with the feature-based technique. To estimate the parameters of the lane model, the likelihood function, Hough transform, and the chi-square fitting, etc. are applied into the lane detection. However, as most lane models are only focused on certain shapes of road, thus they lack the flexibility to model the arbitrary shape of road. In the proposed algorithm to detect the lanes, a combination of feature and model base is used. In general, this algorithm is valid for all kinds of roads (whether they are marked with white lanes or not).

- **Extract the color range for the road :**

  Extract the appropriate upper and lower range to determine the color of the portion on which the car is standing. This is the primary and most important part of this algorithm. Using this range, a binary image of the current view is created.

- **Define The region of interest :**

  A region of interest is defined starting from the bottom towards upward. As the view is taken from the camera on the car, the road surface closest to the car is at the bottom of the image. Thus the region of interest is defined from the nearest region to the farther distances by moving upwards in the image created. The height of the region of interest is usually not more than half the height of the image. This removes the sky area from the visual field and saves unnecessary computations and is better than the method proposed by H.Dahlkamp. As we move away from the car, the width of the road seems to be narrowing. So, the actual region of interest is in the shape of a trapezium.

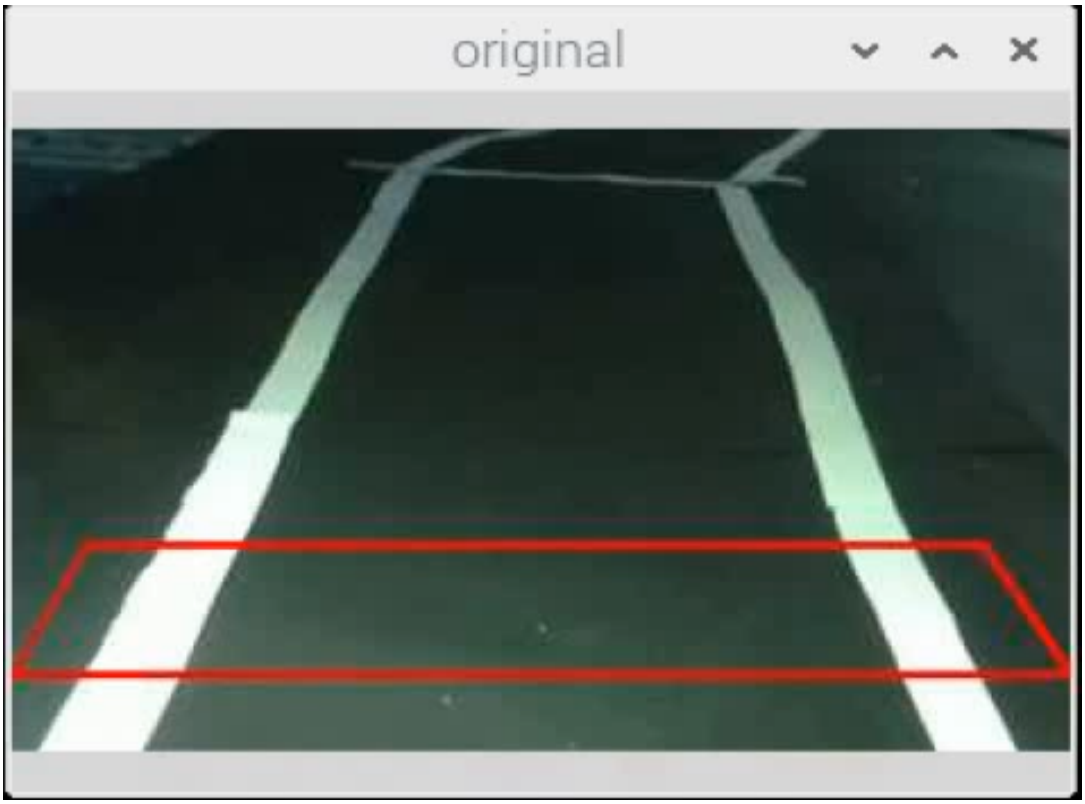

Fig. 12 Region of interest

One way to define an ROI is to use a cv::Rect instance. As the name indicates, it describes a rectangular region by specifying the position of the upper-left corner (the first two parameters of the constructor) and the size of the rectangle (the width and height are given in the last two parameters). In our example, we used the size of the image and the size of the logo in order to determine the position where the logo would cover the bottom-right corner of the image. Obviously, the ROI should always be completely inside the parent image.

- **Convert complex region of interest into simple shape :**

This is an important step in determining the boundary of the road. In the region of interest, the contours are determined. Since the car is on the road, the largest contour which contains the central part of the bottom region is the road. Simplify the shape of this contour using approximations.

Fig. 13 Perspective transformation

In Perspective Transformation, , we can change the perspective of a given image or video for getting better insights about the required information. In Perspective Transformation, we need provide the points on the image from which want to gather information by changing the perspective. We also need to provide the points inside which we want to display our image. Then, we get the perspective transform from the two given set of points and wrap it with the original image.

- **Determine the shape of the road :**

We draw Hough lines on the manipulated contour. A number of Hough lines are obtained along the left and right edges of the road. Out of these, only a few lines represent the actual edge. Rests of the lines are due to the noise (due to irregularities in the road) along the edge. 3.1.5 Filtering the noise The lines along the left edge of the road are tilted towards right and vice versa. So, any line which is tilted towards left and lies entirely in the left half the image is discarded. Similar lines are also discarded from the right half.For the left edge, a line with the smallest angle or having the least positive x intercept or y intercept is chosen as the left edge of the road. Similarly, find the right edge of the road. A general case could be represented as shown in Fig 5.

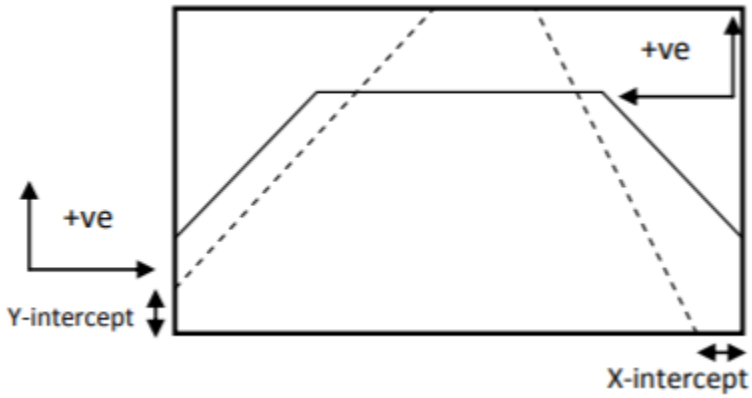

Fig. 14 Conventions for positive intercepts

For the moving car, the change in the direction of the road cannot be abrupt. The edges of the road cannot change the angle in discrete fashion. So, any line which is far away from the line in the previous frame is simply discarded. Another factor called "tolerance" is considered. Basically, it is the count of the continuous frames which could be accepted without being able to determine the edge in the entire above-mentioned process. Its maximum value is 3. If we are not able to determine the edge of the road, tolerance value is decremented by 1. If it reaches 0, we retry to find the new colour range of the road at the run time.

- **Lane Detection :**

Using OpenCV again we then process the image to maximize white and black, since the lanes are white colored and the rest of the track is black, it will be easier to detect lanes on track. Then we perform some operations to detect the lanes, first Perspective transformation, drawing lines on track by finding whitest pixels in the image.

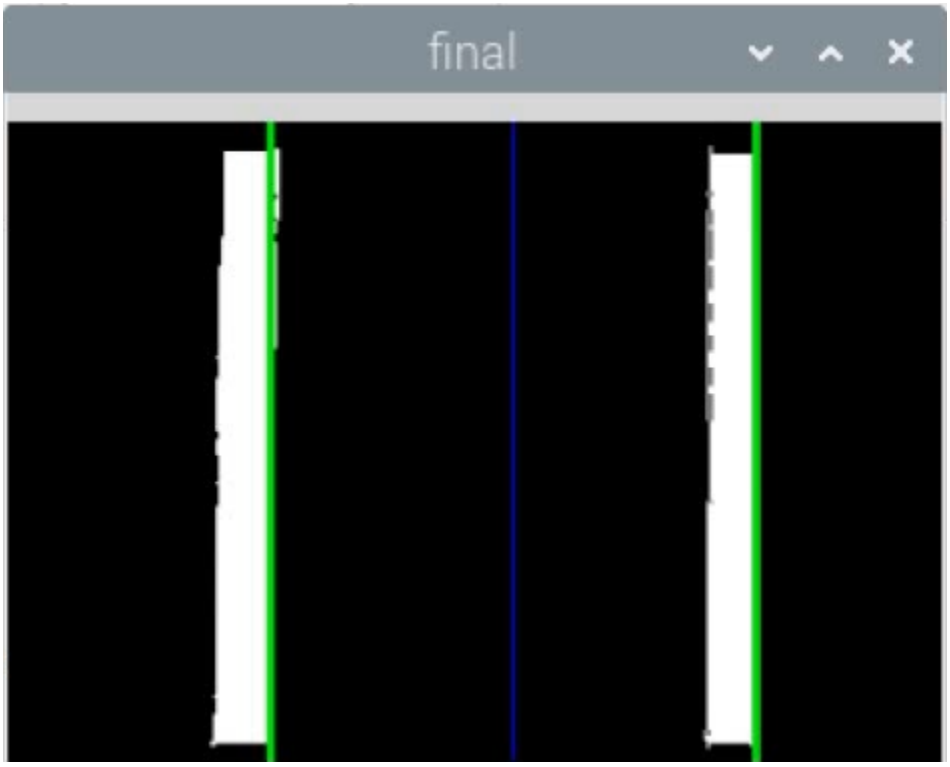

Fig. 15 Lane detection

Then we draw two important lines which will determine whether the vehicle is in the middle of the lane or not. The first line will be the center of the frame captured by the camera, and the second line will be the center of the two lane lines drawn by OpenCV.

Now if the vehicle drifts to the left side of the lane, then there will be a negative difference between first and second line. Then the raspberry pi will send this negative difference to the arduino and the arduino will adjust the speed and direction of motors to make this difference zero.
Positive difference means the vehicle is drifted to the right side of the lane. Zero difference between two lines means the vehicle is perfectly in the middle of the lane, this is our objective.

Results :

The whole detection algorithm was implemented in Raspberry Pi 3B, and it processed close to six frames every second, which is suitable for moderately speeded vehicles. Initially, we sent 3 bytes of data containing our prototype's current position from Raspberry pi to Arduino via UART, which made our system unable to plan its motion in real-time. However, as we changed our approach, which is described in our methodology, and started to send two-byte positional data to the Arduino, the prototype started working fine in real time. The prototype has completed ten indoor tracks equipped with bumps, sudden disappearance of lane markers, and barely visible lane markers, which mimics scenarios faced in real roads. The tracks' cumulative length is 2 meters, and the prototype completed all of those without going out of the lane. Although there were cases of noisy measurements, our prediction-based tracking algorithms helped our prototype to remain on the track.

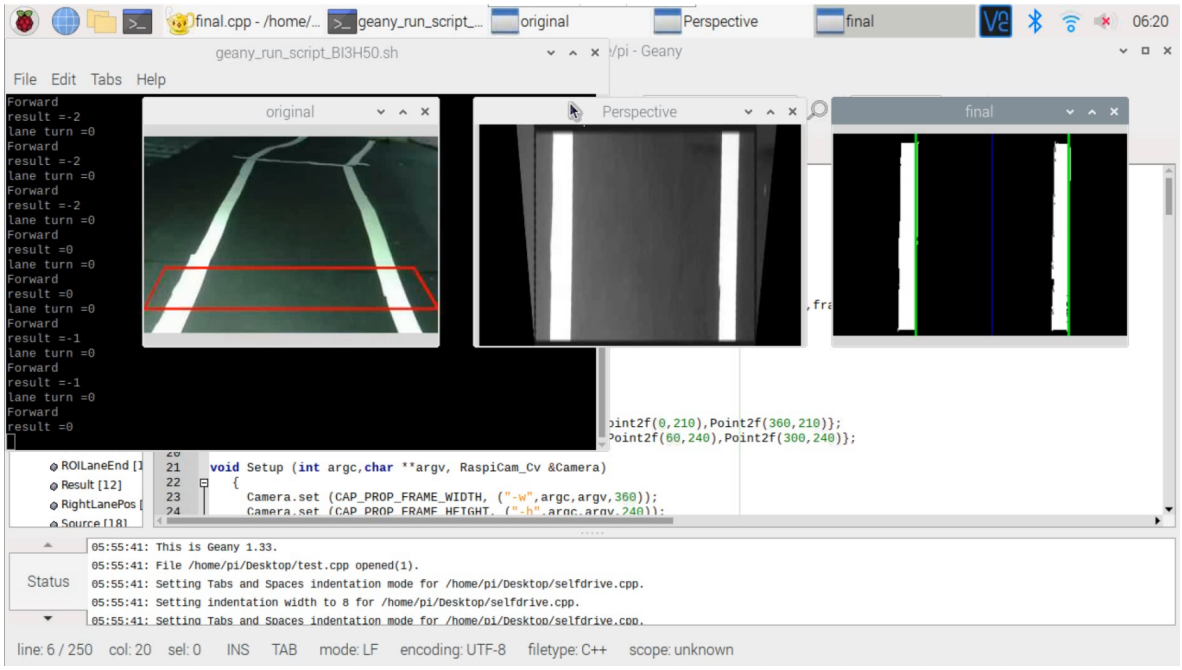

Fig. 16 Final result screen

CHAPTER 08

APPLICATION'S

Applications :

The self driving vehicles have wide range of applications, some of them are :

- **Self driving car** : Driverless cars stand to solve all sorts of problems, like traffic delays and traffic collisions caused by driver error. One of the most significant concerns of self-driving cars is their safety, in theory autonomous systems have a tremendous impact on safety, specifically reducing human errors that are often the cause of accidents.
- **Shipping & Deliveries** : The shipping industry stands to gain a lot from driverless vehicles, especially since the vast majority of companies still rely on human labor to move products from one location to another. Take for instance your standard shipping trailer that travels from state to state in the U.S: with driverless vehicles, these trailers won't have to stop due to driver fatigue, hunger, the call of nature, or other human-centric concerns, which should reduce shipping times by a considerable amount.
- **Public Transportation** : For example, say you take your self-driving car to work, and instead of your vehicle simply sitting in a garage, it goes and starts shuttling other commuters within a predefined area for a fee. With this level of driverless car integration in our transportation system, commuters have access to a near limitless number of commuting options at their disposal, and vehicle owners have access to a new stream of revenue.
- **Emergency Transportation** : Finally, the implementation of driverless vehicles will allow Americans to receive emergency medical care faster than ever before.With traffic congestion reduced (or eliminated) from our streets, emergency vehicles will be able to reach their destinations much more quickly. Additionally, drivers who are having a medical emergency will be able to automatically adjust the course of their vehicle to the nearest E.R without having to worry about operating the vehicle.

CHAPTER 09

LIMITATIONS

Limitations :

- **Security issues** : One of the potential cons about self-driving cars is the possibility of hacking. To have automated cars talk and coordinate with each other, they would need to share the same network protocol if a large number of cars share the same network however they would be susceptible to hacking. Even a small hack could wreak significant damage on busy roads by causing collision sand gridlock traffic.
- **Initial costs** : While self-driving vehicles may produce significant societal cost savings in the long term, the initial cost of automated vehicles may be astronomical.
- **Machine error** : Machine error must be taken into consideration. If the software or any part of the vehicle fails, it can put the driver in more danger. Furthermore, most agree that self-driving vehicles will likely prevent more accidents from happening.

CHAPTER 10

FUTURE SCOPE

Future scope :

The future of Self-driving vehicles in India is not too far. The technology is still new and growing. Indian roads are a bit more challenging as compared to roads in western countries.
The demand for self-driving vehicles is most in India. Additionally, it can possibly prevail here. Be that as it may, the real question is `Are we prepared at this point ?`

In the first place, we need to consider the necessity of autonomous vehicles in india. Indian streets are busy with traffic leading to highly congested roads. The number of road accidents are far greater due to them. These are some pain points which need to be considered.
With the significant growth in the field of Internet of Things, it is safe to say it will pave the way for self-driven vehicles. In India, Ola is one of the many players experimenting with parts of autonomous car technology. Many startups have come who've invested heavily in this technology.

CHAPTER 11

BIBLIOGRAPHY

Bibliography :

- Design & Implementation of Autonomous vehicle using raspberry pi by Gurjashan Pannu, INTERNATIONAL JOURNAL OF COMPUTER APPLICATION, researchgate.net, March 2015.
- Real time Lane detection and motion planning in raspberry pi & Arduino for an autonomous vehicle prototype by Alfa Rossi (arxiv.org) 20 Sep 2020.
- Traffic light detection using tensorflow object detection by Tiagrajah v Janahiraman (2019 IEEE 9th international conference on system engineering and technology) (ICSET) (researchgate.net) October 2019.
- Autonomous object detection and tracking using Raspberry pi by sampa jana.