A Synopsis on

AI based Self Driving Car

Submitted in partial fulfillment of the requirements
of the degree of

Bachelor of Engineering

in

Information Technology

by

Mahek Jain (17204010)
Vinit Shah (17204014)
Hiral Thadeshwar (17204012)

Prof. Rujata Chaudhari
Prof. Vishal Badgujar

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

2019-2020

CERTIFICATE

This is to certify that the project Synopsis entitled ***"AI based Self Driving Car"*** Submitted by ***"Mahek Jain (17204010),Vinit Shah (17204014),Hiral Thadeshwar (17204012)"*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology.***to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020

Prof. Vishal Badgujar
Co-Guide

Prof. Rujata Chaudhari
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Mahek Jain and 17204010)

(Signature)

(Vinit Shah and 17204014)

(Signature)

(Hiral Thadeshwar and 17204012)

Date:

Abstract

The basic idea behind the project is to develop an automated car that is capable of sensing its environment and moving without any human input. The automation will be achieved by detecting lane marking, signals, obstacles, stop sign using image processing and neural network to react and take decisions such as changing the course of the car, stopping on stop signs and red signal, and moving on green signal. Self-driving car processes inputs, plots a path, and sends instructions to the vehicle's actuators, which control acceleration, braking, and steering. Hard-coded rules, obstacle avoidance algorithms, predictive modeling, and "smart" object discrimination help the software follow traffic rules and navigate obstacles.

a monocular vision-based self-driving car prototype using Deep Neural Network on Raspberry Pi is proposed. Self-driving cars are one of the most increasing interests in recent years as the definitely developing relevant hardware and software technologies toward fully autonomous driving capability with no human intervention. Level-3/4 autonomous vehicles are potentially turning into a reality in near future. Convolutional Neural Networks (CNNs) have been shown to achieve significant performance in various perception and control tasks in comparison to other techniques in the latest years. The key factors behind these impressive results are their ability to learn millions of parameters using a large amount of labeled data. In this work, we concentrate on finding a model that directly maps raw input images to a predicted steering angle as output using a deep neural network. The technical contributions of this work are two-fold.

First, the CNN model parameters were trained by using data collected from vehicle platform built with a 1/10 scale RC car, Raspberry Pi 3 Model B computer and front- facing camera. The training data were road images paired with the time-synchronized steering angle generated by manually driving. Second, road tests the model on Raspberry to drive itself in the outdoor environment around oval-shaped and 8-shaped with traffic sign lined track. The experimental results demonstrate the effectiveness and robustness of autopilot model in lane keeping task. Vehicle's top speed is about 5-6km/h in a wide variety of driving conditions, regardless of whether lane markings are present or not.

Introduction

Human driven cars use technologies to provide safety and detect obstacles and auto stop in various high end cars but none of them works completely driver less. The existing cars does not contain the feature of automation to the extent that car can drive autonomously. There is constant need of drivers without it the car becomes unavailable but with self driving car we can make the availability of car constant on roads. In traditional cars the driver constantly needs to keep check on the signals, road safety signs, obstacles and lane and needs to make decisions accordingly.

Whereas a self-driving car, also known as a robot car, autonomous car, or driverless car, is a vehicle that is capable of sensing its environment and moving with little or no human input. Self-driving cars can detect environments using a variety of techniques such as radar, GPS and computer vision. It will also interpret sensory information to identify appropriate navigational paths, as well as obstacles and relevant signage. Self-driving cars have control systems that are capable of analyzing sensory data to distinguish between different cars on the road. This is very useful in planning a path to the desired destination.

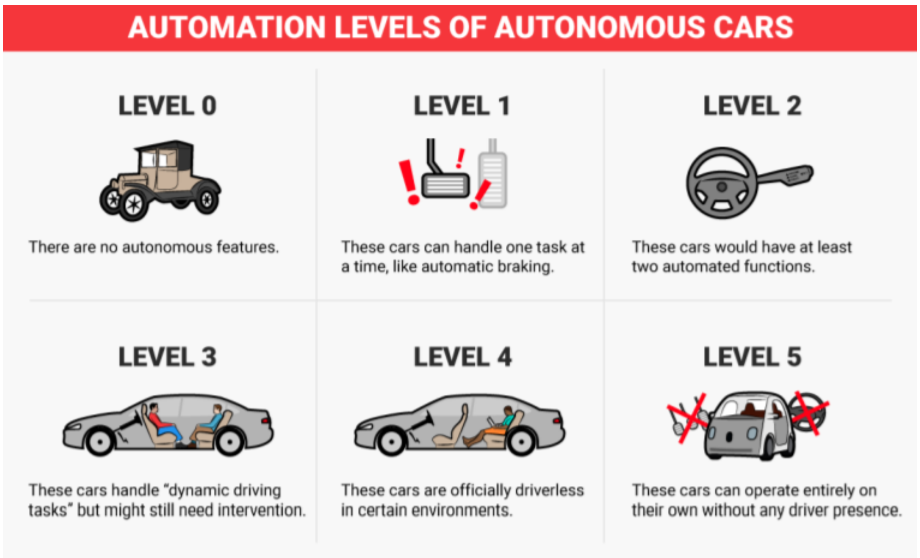


Figure 1: Automation Levels of Autonomous Car

The above diagram shows different automation levels which are described below:

Level 0 (No Automation): The full-time performance by the human driver of all aspects of the dynamic driving task, even when enhanced by warning or intervention systems.
Level 1 (Driver Assistance): The driving mode-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task.
Level 2 (Partial Automation): The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task.
Level 3 (Conditional Automation): The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task with the expectation that the human driver will respond appropriately to a request to intervene.
Level 4 (High Automation): The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task even if a human driver does not respond appropriately to a request to intervene.
Level 5 (Full Automation): The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task under all roadway and environmental conditions that can be managed by a human driver.

The advantages of a self driving car over human driven car can be:

1. Safety

Driving safety experts predict that once driverless technology has been fully developed, traffic collisions (and resulting deaths and injuries and costs), caused by human error, such as delayed reaction time, tailgating, rubbernecking, and other forms of distracted or aggressive driving should be substantially reduced.

2. Welfare

Automated cars could reduce labor costs relieve travelers from driving and navigation chores, thereby replacing behind-the-wheel commuting hours with more time for leisure or work and also would lift constraints on occupant ability to drive, distracted and texting while driving, intoxicated, prone to seizures, or otherwise impaired. For the young, the elderly, people with disabilities, and low income citizens, automated cars could provide enhanced mobility.

3. Traffic

Additional advantages could include higher speed limits smoother rides and increased roadway capacity and minimized traffic congestion, due to decreased need for safety gaps and higher speeds.

Objectives

The objectives behind developing this project are:

1. To have a driverless car.

Cars without driver will be helpfull to humans to a great extent,human mind can get diverted but the system has predefined set of rules to act in various situation which will reduce accidents.

2. To provide user convenience.

User just needs to enter the destination and he/she is done, user can sit back do his/her work or simply can rest for sometime instead of driving.

3. To make service available 24*7.

Human need rest so no one can drive for hours without rest, the system enables user to rest and travel 24*7 non stop without getting exhausted.

4. If a practical self-driving car is ever mass produced, the first major change would be to ride services.

If the implementation of project gets successful it will be a boon to companies providing ride services(ola,uber,zoom cars etc), as there will be no risk at all, the risk of appointing a good driver, checking driving skills, car safety all this risk will vanish.

5. Instead of relying on human instincts built up over millions of years that ensure selfp-reservation, a computer will be safer.

Humans act differently in same situation so the chance of getting misintercepted increases which cause rise in number of accidents, but if things get computerised there will be same action taken by all cars in same scenerio which decreases accidents and increases safety.

Literature Review

During the development of the project following papers has been referred :

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

-In this paper, they have presented an autonomous car platform based on the softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function.[1] The softmax function acts as sigmoid function by ranging the output while an actual softmax function does not do that. Using deep neural network helps in giving real time output. They have implemented the model on MATLAB simulator before implementing actually. The system uses only one single camera for all inputs and it drives at about 5-6 km/hr whether the lane markings are present or no. This model only detects lane markings and turn signs. It just hovers the car left or right and does not sense signals or stop sign.

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

-They have built an autonomous RC Car that uses Artificial Neural Network (ANN) for control. It describes the theory behind the neural network and autonomous vehicles.[2] Using L298N IC and motor driver a car is made which can be controlled by a microcontroller and then in return sent to the model car. Using cnn helps in only detection of gray scale parts and it ignores the unnecessary data in the detection. The system used is very limited but accurate. Using an embedded pi camera for input and gray scale of images for training in neural network.[3] The system detects lane markings for each direction and does not offer any other functionality other than that.

3. Driverless Car: Autonomous Driving Using Deep Reinforcement Learning In Urban Environment, Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, and Deok-Jin Lee, in 15th International Conference on Ubiquitous Robots (UR) Hawaii Convention Center, Hawai'i, USA, June 27-30, 2018

-In this paper, they have presented a reinforcement-learning based approach with Deep Q Network implemented in autonomous driving.[4] Using lidar sensors it detect objects at a very far distance. The whole system is developed on a simulator depicting actual roads and city streets with traffic. Using fusion of camera and lidar helps in better knowing of the surroundings and all kinds of obstacles. They have implemented a model using lidar(laser sensor) which is a very costly sensor and it is applicable for large scale cars.

4. Object Detection Using Deep Neural Networks, Malay Shah, Prof, Rupal Kapdi, in International Conference on Intelligent Computing and Control Systems ICICCS 2017

-The problem discussed in this article is object detection using deep neural network especially convolution neural networks. Object detection was previously done using only conventional deep convolution neural network whereas using regional based convolution network[5] increases the accuracy and also decreases the time required to complete the program.[6] Training a neural network from scratch takes more time and processing power as it is very difficult to find the dataset of sufficient size and ground truth. Using Regional Convolutional Neural Network(RCNN) helps in finding appropriate regions in image and it enables the system to give real time outputs. This deep neural network is used for image processing, mainly for medical uses like tumor and such where the data set is too complex to detect regions in comparison to a model road environment.

5. Self-driving Cars Using CNN and Q-learning, Syed Owais Ali Chishti, Sana Riaz, Muhammad Bilal Zaib, Mohammad Nauman, in 2018 IEEE 21st International Multi-Topic Conference (INMIC)

-They have developed a car which is trained for three different road signs; Stop, No left, and Traffic light using DQN with existing CNN model. These road signs are detected in the environment using OpenCV cascade classifiers. Supervised learning was done using Convolutional Neural Network and we achieved 73perc test and 89perc train accuracy.[7] In addition to this they trained the car with reinforcement learning for three different sign boards; Stop, No left and Traffic light using deep Q-learning technique with CNN model. Currently, the training is done in a constrained environment, many factors of real environment can sway the prediction of model. Lightening effects and weather conditions influences the images they get from cameras which can have a great impact on model prediction.

6. Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino, Aditya Kumar Jain, in Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018)

-Their proposed model takes an image with the help of Pi cam attached with Raspberry Pi on the car. The Raspberry-Pi and the laptop is connected to the same network, the Raspberry Pi sends the image captured which serves as the input image to the Convolutional Neural Network. The image is gray-scaled before passing it to the Neural Network. Upon prediction the model gives one of the four output i.e. left, right, forward or stop. When the result is predicted corresponding Arduino signal is triggered which in turn helps the car to move in a particular direction with the help of its controller.[8] Their car was trained under different combinations of the track i.e. straight, curved, combination of straight and curved and etc. Total of 24 videos were recorded out of which images were extracted.10868 images were extracted and was categorically placed in different folders like left, right, straight and stop. In this paper, a method to make a model of self-driving car is presented. The different hardware components along with software and neural network configuration are clearly described. With the help of Image Processing and Machine Learning a successful model was developed which worked as per expectation. Thus the model was successfully designed, implemented and tested. The car slightly moves out of the track which can be a serious issue if it hits nearby objects if we consider a real car.

Problem Definition

Problem in the existing system is that the existing cars needs effort to be driven and certain amount of time goes in reaching one destination which could be utilised in some other work.

To provide a solution to the above problem we are implementing the project where the existing human driven cars will be replaced by effective self driven cars keeping in mind of various safety conditions. So developing a model scaled car which can be implemented using monocular vision method and technology in an actual car. Implementing machine learning and image processing to gain better knowledge of new technologies. We are creating a prototype to explain the concepts of machine learning which will be helpful to teach the students as it can be presented as the live example of machine learning.

Proposed System Architecture/Working

-This project builds a self-driving RC car using Raspberry Pi, Arduino and open source software.
-Raspberry Pi collects inputs from a camera module and an ultrasonic sensor, and sends data to a computer wirelessly.
-The computer processes input images and sensor data for object detection (stop sign and traffic light) and collision avoidance respectively.
-A neural network model runs on computer and makes predictions for steering based on input images. Predictions are then sent to the Arduino for RC car control.
-The system uses sensor and camera constantly to check for obstacles and safety signs.

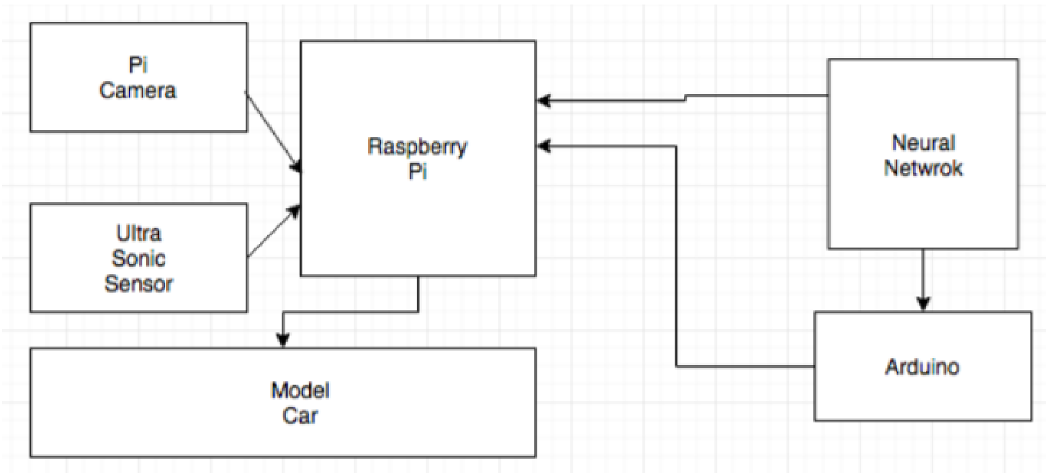


Figure 2: Proposed System Block Diagram

Hardware components used in the project are:

Raspberry Pi 3: SoC: Broadcom BCM2837, CPU: 4 ARM Cortex-A53, 1.2GHz, GPU: Broadcom VideoCore IV, RAM: 1GB LPDDR2 (900 MHz), Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless, Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy, Storage: microSD, GPIO: 40-pin header, populated, Ports: HDMI, 3.5mm analogue audio-video jack, 4 USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI).

OpenCV : Decision tree learning, Expectation-maximization algorithm, k-nearest neighbor algorithm, Naive Bayes classifier, Artificial neural networks, Random forest, Support vector machine (SVM), Deep neural networks (DNN).

Arduino Nano: Microcontroller ATmega328, Operating Voltage (logic level): 5 V, Input Voltage (recommended): 7-12 V, Input Voltage (limits): 6-20 V, Digital I/O Pins : 14 (of which 6 provide PWM output), Analog Input Pins: 8, DC Current per I/O Pin: 40 mA, Flash Memory 32 KB (ATmega328) of which 2 KB used by bootloader, SRAM: 2 KB (ATmega328), EEPROM: 1 KB (ATmega328), Clock Speed: 16 MHz, Dimensions: 0.73" x 1.70".

Ultrasonic sensor : Power Supply :+5V DC, Quiescent Current : ¡2mA, Working Current: 15mA, Effectual Angle: ¡15, Ranging Distance : 2cm – 400 cm/1 – 13ft, Resolution : 0.3 cm, Measuring Angle: 30 degree, Trigger Input Pulse width: 10uS, Dimension: 45mm x 20mm x 15mm.

Pi camera : 8 megapixel native resolution sensor-capable of 3280 x 2464 pixel static images, Supports 1080p30, 720p60 and 640x480p90 video, Camera is supported in the latest version of Raspbian, Raspberry Pi's preferred operating system.

Design and Implementation

The system consists of three subsystems: input unit (camera, ultrasonic sensor), processing unit (computer) and RC car control unit.

Input Unit
A Raspberry Pi board (model B+), attached with a pi camera module and an HC-SR04 ultrasonic sensor is used to collect input data. Two client programs run on Raspberry Pi for streaming color video and ultrasonic sensor data to the computer via local Wi-Fi connection. In order to achieve low latency video streaming, video is scaled down to QVGA (320240) resolution.

Processing Unit
The processing unit (computer) handles multiple tasks: receiving data from Raspberry Pi, neural network training and prediction(steering), object detection(stop sign and traffic light), distance measurement(monocular vision), and sending instructions to Arduino through USB connection.

TCP Server
A multithread TCP server program runs on the computer to receive streamed image frames and ultrasonic data from the Raspberry Pi. Image frames are converted to gray scale and are decoded into numpy arrays.

Neural Network
One advantage of using neural network is that once the network is trained, it only needs to load trained parameters afterwards, thus prediction can be very fast. Only lower half of the input image is used for training and prediction purposes. There are 38,400 (320120) nodes in the input layer and 32 nodes in the hidden layer. The number of nodes in the hidden layer is chosen fairly arbitrary. There are four nodes in the output layer where each node corresponds to the steering control instructions: left, right, forward and reverse respectively (though reverse is not used anywhere in this project, it's still included in the output layer).
First each frame is cropped and converted to a numpy array. Then the train image is paired with train label (human input). Finally, all paired image data and labels are saved into a npz file. The neural network is trained in OpenCV using back propagation method. Once training is done, weights are saved into a xml file. To generate predictions, the same neural network is constructed and loaded with the trained xml file.

Object Detection
This project adapted the shape-based approach and used Haar feature-based cascade classifiers for object detection. Since each object requires its own classifier and follows the same process in training and detection, this project only focused on stop sign and traffic light detection.
OpenCV provides a trainer as well as detector. Positive samples (contain target object) were acquired using a cell phone, and were cropped that only desired object is visible. Negative samples (without target object), on the other hand, were collected randomly. In particular, traffic light positive samples contains equal number of red traffic lights and green traffic light. The same negative sample dataset was used for both stop sign and traffic light training. Below shows some positive and negative samples used in this project.

To recognize different states of the traffic light(red, green), some image processing is needed beyond detection. Flowchart below summarizes the traffic light recognition process.
Firstly, trained cascade classifier is used to detect traffic light. The bounding box is considered as a region of interest (ROI). Secondly, Gaussian blur is applied inside the ROI to reduce noises. Thirdly, find the brightest point in the ROI. Finally, red or green states are determined simply based on the position of the brightest spot in the ROI.

Figure 3: Haar cascade classifiers for object detection

Firstly, trained cascade classifier is used to detect traffic light. The bounding box is considered as a region of interest (ROI). Secondly, Gaussian blur is applied inside the ROI to reduce noises. Thirdly, find the brightest point in the ROI. Finally, red or green states are determined simply based on the position of the brightest spot in the ROI.

Distance Measurement
Raspberry Pi can only support one pi camera module. Using two USB web cameras will bring extra weight to the RC car and also seems unpractical. Therefore, monocular vision method is chosen.
This project adapted a geometry model of detecting distance to an object using monocular vision method proposed by Chu, Ji, Guo, Li and Wang (2004).

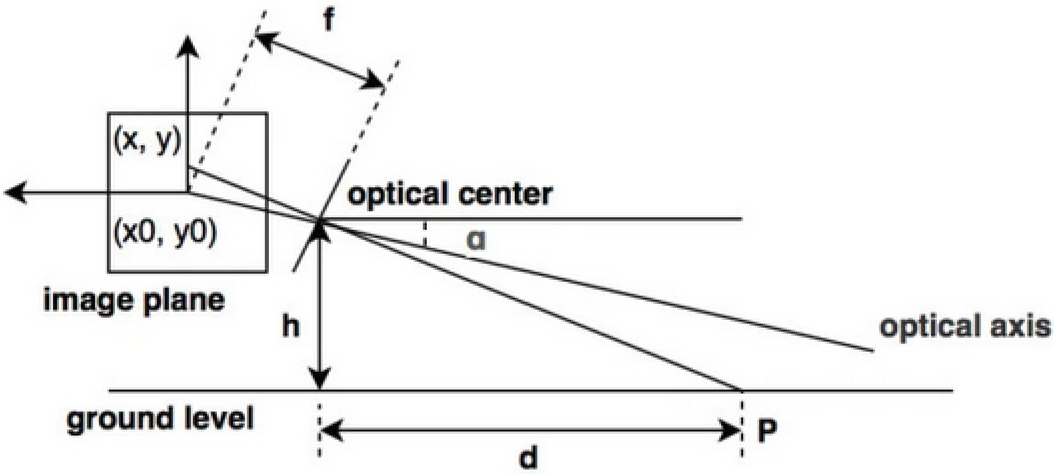


Figure 4: Distance measurement using monocular vision

P is a point on the target object; d is the distance from optical center to the point P. Based on the geometry relationship above, formula (1) shows how to calculate the distance d. In the formula (1), f is the focal length of the camera; is camera tilt angle; h is optical center height; (x0, y0) refers to the intersection point of image plane and optical axis; (x, y) refers to projection of point P on the image plane. Suppose O1 (u0,v0) is the camera coordinate of intersection point of optical axis and image plane, also suppose the physical dimension of a pixel corresponding to x-axis and y-axis on the image plane are dx and dy. Then:

$$d = h/\tan\left(\partial + \arctan\left((y - y_0)/f\right)\right) \tag{1}$$

$$u = \frac{x}{dx} + u_0 \quad v = \frac{y}{dy} + v_0 \tag{2}$$

Let $x_0 = y_0 = 0$, from (1) and (2):

$$d = h/\tan\left(\alpha + \arctan\left((v - v_0)/a_y\right)\right), \qquad (a_y = f/dy) \tag{3}$$

v is the camera coordinates on y-axis and can be returned from the object detection process. All other parameters are camera's intrinsic parameters that can be retrieved from camera matrix. ;ine OpenCV provides functions for camera calibration. Camera matrix for the 5MP pi camera is returned after calibration. Ideally, a(x) and a(y) have the same value. Variance of these two values will result in non-square pixels in the image. The matrix below indicates that the fixed focal length lens on pi camera provides a reasonably good result in handling distortion aspect. Here is an interesting article discussing the focal length of pi camera with stock lens and its equivalent to 35mm camera.

$$\begin{bmatrix} a_x = 331.7 & 0 & u_0 = 161.9 \\ 0 & a_y = 332.3 & v_0 = 119.8 \\ 0 & 0 & 1 \end{bmatrix}$$

The matrix returns values in pixels and h is measured in centimeters. By applying formula (3), the physical distance d is calculated in centimeters.

RC Car Control Unit
The RC car used in this project has an on/off switch type controller. When a button is pressed, the resistance between the relevant chip pin and ground is zero. Thus, an Arduino board is used to simulate button-press actions. Four Arduino pins are chosen to connect four chip pins on the controller, corresponding to forward, reverse, left and right actions respectively. Arduino pins sending LOW signal indicates grounding the chip pins of the controller; on the other hand sending HIGH signal indicates the resistance between chip pins and ground remain unchanged. The Arduino is connected to the computer via USB. The computer outputs commands to Arduino using serial interface, and then the Arduino reads the commands and writes out LOW or HIGH signals, simulating button-press actions to drive the RC car.

Implementation of project

Testing: Flash rc-keyboard-control.ino to Arduino and we run rc-control-test.py to drive the RC car with keyboard. Then we run stream-server-test.py on computer and then we run stream-client.py on raspberry pi to test video streaming. Similarly, ultrasonic-server-test.py and ultrasonic-client.py is used for sensor data streaming testing.

Pi Camera calibration: Took multiple chess board images using pi camera module at various angles and put them into chess-board folder, and run picam-calibration.py and returned parameters from the camera matrix is used in rc-driver.py.

Collect training/validation data: First we run collect-training-data.py and then run stream-client.py on raspberry pi. Using arrow keys we can drive the RC car, q to exit. Frames are saved only when there is a key press action. Once exit, data will be saved into newly created training-data folder.

Design

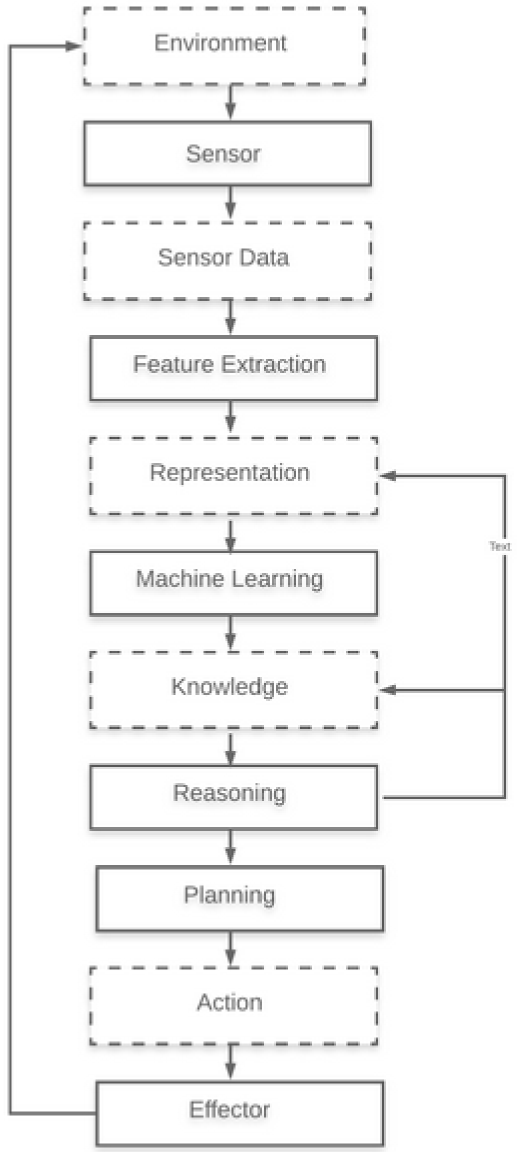


Figure 5: Flow Diagram

Sequence Diagram

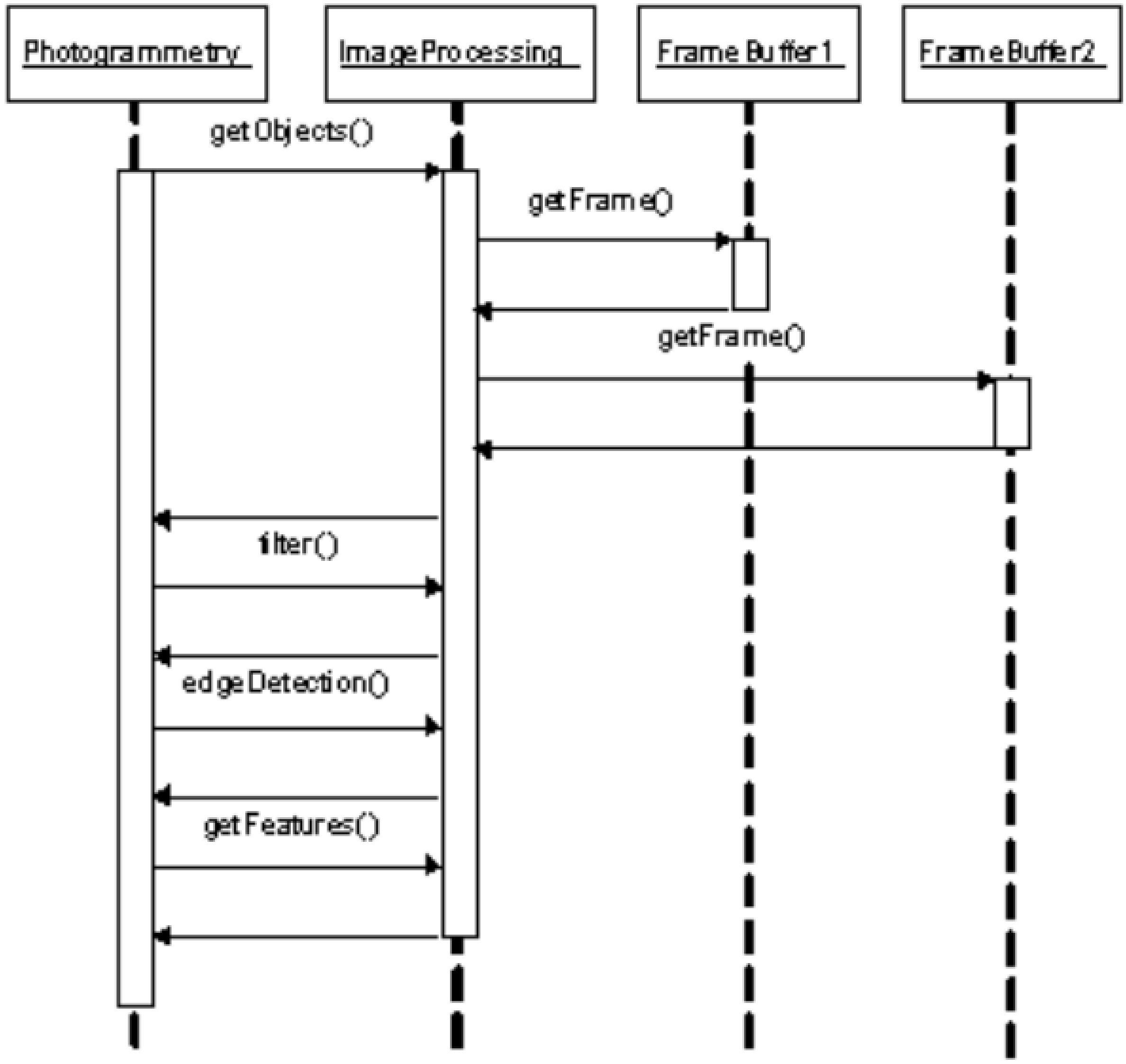


Figure 6: Sequence Diagram for Image Processing

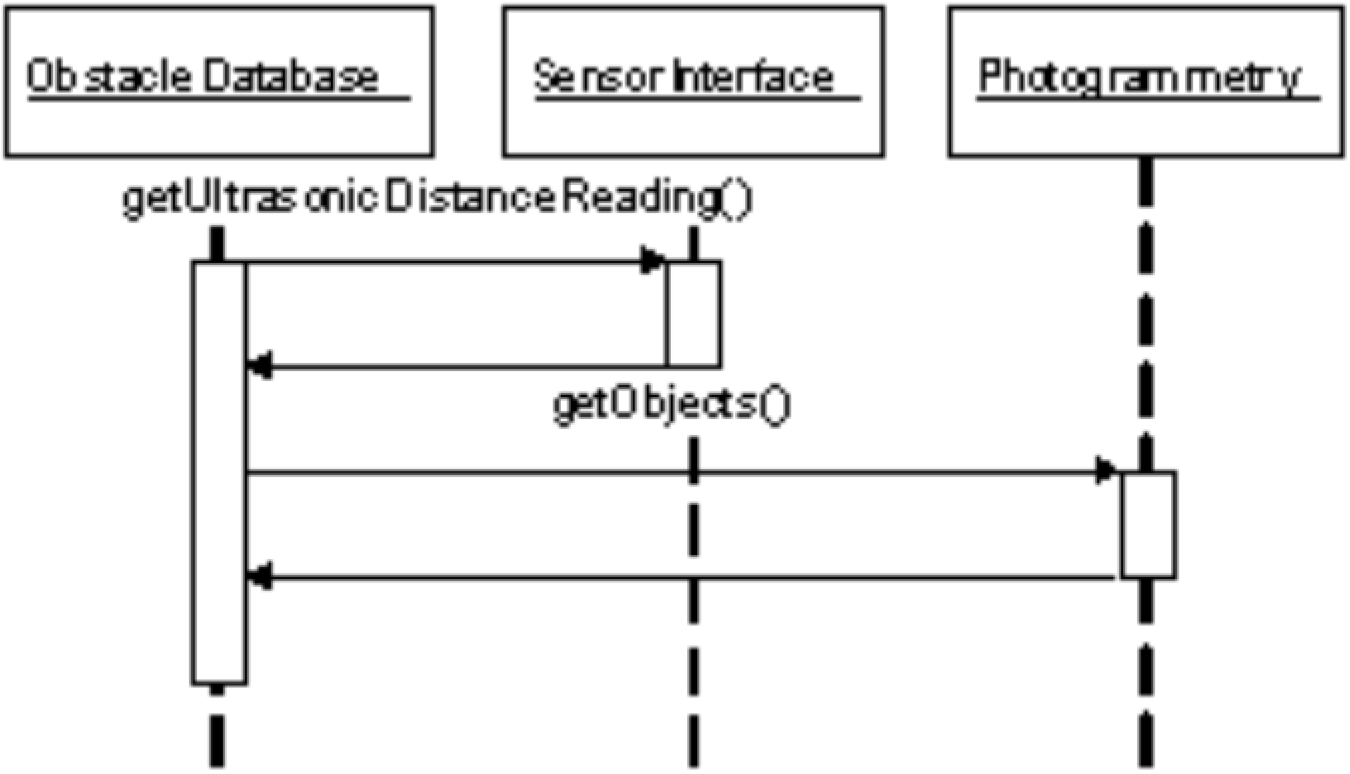


Figure 7: Sequence Diagram for Obstacle Detection

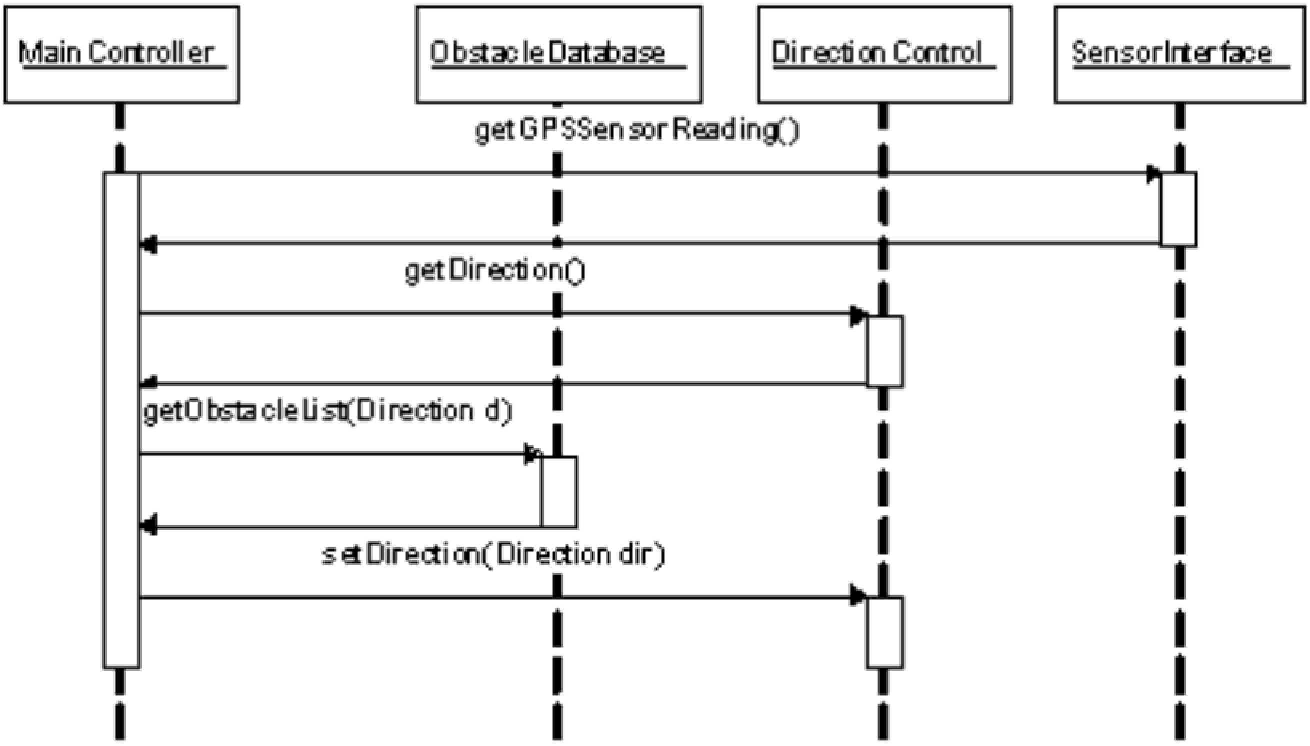
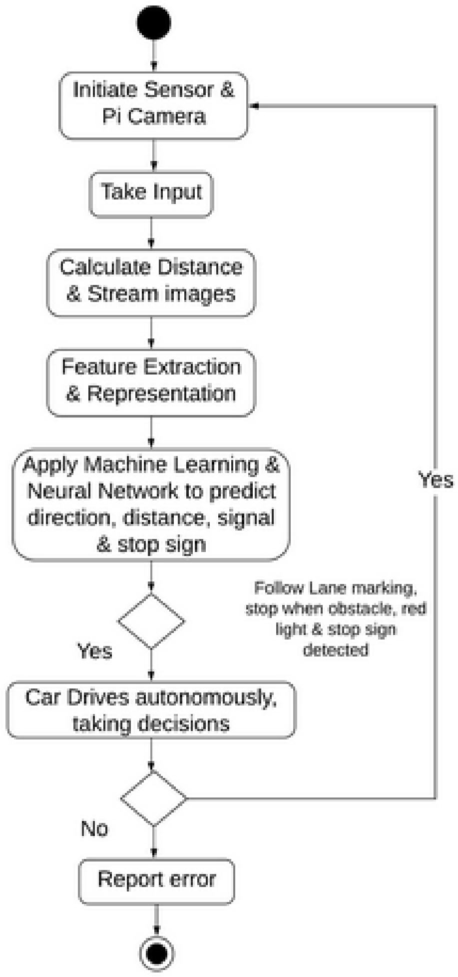


Figure 8: Sequence Diagram for Direction Prediction

Activity Diagram



Summary

The work presented in this report is related to Autonomous Car.

1. Distance calculation using monocular vision.

2. Lane detection, stop sign detection, signal detection using machine learning.

References

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

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

[3] F. Shumaila Mateenuddin, Mrs. V.S. Jahagirdar, "An Android Controlled Mini Rover for real time surveillance using Raspberry Pi 3", in International Journal Of Advanced Research in Engineering Management (IJAREM), 2017.

[4] Abdur R. Fayjie, Sabir Hossain, Doukhi Oualid, and Deok-Jin Lee, "Driverless Car: Autonomous Driving Using Deep Reinforcement Learning In Urban Environment" in 15th International Conference on Ubiquitous Robots (UR) Hawaii Convention Center, Hawai'i, USA, June 27-30, 2018

[5] Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580-587

[6] Malay Shah, Prof, Rupal Kapdi, "Object Detection Using Deep Neural Networks", in International Conference on Intelligent Computing and Control Systems ICICCS 2017

[7] Syed Owais Ali Chishti, Sana Riaz, Muhammad Bilal Zaib, Mohammad Nauman, "Self-driving Cars Using CNN and Q-learning", in 2018 IEEE 21st International Multi-Topic Conference (INMIC)

[8] Aditya Kumar Jain, "Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino", in Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018) IEEE Conference Record 42487; IEEE Xplore ISBN:978-1-5386-0965-1