ABSTRACT

With increase in movement of goods and raw material being transported to various location. Companies are searching for cheap and cost effective ways to transport goods form one place to other. The already existing software relies on rigid programming which is incapable to handle the continuously changing environment of warehouses or other places. With such a technology the navigating and safety of such autonomous would increase. Not only the prices could be decrease but also more and more people would start developing there interest in machine learning programming. Though machine learning has been with us for about more then 50 years now. But the development in hardware and improvement in networking has made it much more practical in today's world. Also, with the current rapid economic growth, vehicle ownership is fast increasing, accompanied by more than one million traffic accidents per year worldwide. According to statistics, about 89.8% of accidents are caused by driver's wrong decision-making. Self-driving vehicles aim to eliminate human driver error leading to fewer injuries and fatalities. The project aims to build a miniature vision autonomous car prototype using Raspberry Pi as a processing chip. An HD camera along with an Distance sensor is used to provide necessary data from the real world to the car. The car is capable of reaching the given destination safely and intelligently thus avoiding the risk of human errors. Concepts like Machine Learning, Artificial Intelligence and Image Processing are used to generate a prospective of surrounding in model.

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 Project Introduction

Online shopping and improvement in automation in industrial work has lead to increase in movement of products and raw material. This increase in movement of goods and raw material being transported to various location. Companies are searching for cheap and cost effective ways to transport goods form one place to other. The already existing software relies on rigid programming which is incapable to handle the continuously changing environment of warehouses or other places. With such a technology the navigating and safety of such autonomous would increase. Not only the prices could be decrease but also more and more people would start developing there interest in machine learning programming. Though machine learning has been with us for about more then 50 years now. But the development in hardware and improvement in networking has made it much more practical in today’s world.With the growth number of vehicle ownership is fast increasing, accompanied by more than one million traffic accidents per year worldwide. According to statistics, about 89.8% of accidents are caused by driver’s wrong decision-making. Smart Driving system for vehicles aim to eliminate human driver error leading to fewer injuries and fatalities. The project aims to build a miniature vision of smart car prototype using Raspberry Pi. Camera provides live feed to microprocessor which can then be processed to identify cars, bikes, or other obstacle. The car is capable of reaching the given destination safely and intelligently thus avoiding the risk of human errors. Concepts like Machine Learning, Artificial Intelligence and Image Processing are used to generate a prospective of surrounding in model Its works by using the training data provided and generated previously. Then computing the data and training the rc car to fallow the same path that was previously used. The main aim of this project is to introduce a technology that can be user friendly, open-source, cost effective and scope of growing further in an more effective tool. The parts used are basic and easily available in market. Further more, the programming is done in python which is an highly efficient language. OpenCv is the backbone of this project as the neural network which is in play is library provided in it. Hence, even after neural network being such an complicated subjects but still the simplicity of the code remains. The learning process is also kept linear to make the complexity of programming as low as possible.

1.2 Purpose

The purpose for this “Smart RC Car” is to reduce cost and risk associated with autonomous system for transportation of goods. Additional this would also reduce the efforts which are put to make a line follower robot. The cost is kept less as the parts used are easily accessible and the software used are open source and easy to use. Also one more benefit is that it would increase the peoples involvement in Artificial Indulgence and Machine Learning Programming. This would increase the gathering rate and the rate at which improvement are done in this sector.

As the previously existing concept is limited to big companies and research centers. Also, They are equipped with all the modern and complicated circuits and programming concepts. But to its contrast, the method and other resources which are used in this project are much cheap, therefore more vastly accessible. This is more focused on making technology accessible to people, thereby increasing the involvement and further increasing the pace of development. The project consists of continuous data input which need to be processed to keep the passenger safe from accidents. This is done by saving the video feed and other input data and sensor data in an format with can then be given to neural network.

The project is used truck drivers, Long route drivers, commuters, etc for traveling without experiencing the stress of driving. This can be achieved by interconnecting all the sensors into microprocessor. Using Artificial Intelligence, we can then further make decision about driving.

Furthermore, it can detect obstacles, stop sign and red light. These features help the driver in driving responsibly and safely. The data gathered form various verticals can also be used for crime investigation or any other application. This is result of the data which was previously been saved to generate training data.

Our purpose to build this system is:

- To reduce the cost and accessibility of such a system.
- Reduce cost of transporting good and simplify transportation system.
- To make availability of artificial intelligence more access-able and to promote research in this field.
- To provide all the software in an open source manor so that more developers can make modification in present technology.
- To reduce accidents on road by implementing an unambiguous way of transport.
- To eliminate the stress of long root driving.
- Furthermore, it has potential of reducing the cost of transport that commuters have to pay.

1.3 Scope

The scope of this project is in reducing the cost of manufacturing self-driving car. As there is an increase in movement of goods and raw material being transported to various location. Companies are searching for cheap and cost effective ways to transport goods form one place to other. The already existing software relies on rigid programming which is incapable to handle the continuously changing environment of warehouses or other places.This project will be helpful reduce accidents and casualties due to mistakes made by driver.

System can do:

- **User Friendly:**

  User can easily get the information which they want and easily use it. As smart rc car can be trained according to users requirement and can be modified and scaled easily according to demand.

- **Flexibility:**

  Ability to handle different situation that might occur. The input form different sensors are taken and route is decided accordingly. The neural network can work fine even in case of small changes. Additionally it gives flexibility to modify not only neural network structure but also to to add more sensors if required.

- **Authentication Users:**

  The raspberry pi login makes sure that only authenticated users can access network to interact with the programming aspect of car. This provides surety that no intruder could alter code which might change the working of model.

- **Ease of use:**

  Processes and design features which avoid the problems such as confusions, time waste, miss management of resources and other information’s, unclear system structure and excessive cost to the learner.

1.4 Technology and Literature Review

Sr.No	Software	Literature Reviewed
1	Miniconda	https://docs.conda.io/en/latest/miniconda.html
2	Python	https://www.w3schools.com/python/ https://www.tutorialspoint.com/python/ https://docs.python.org/3/tutorial/
3	Raspbian OS	https://www.raspberrypi.org/downloads/raspbian/
4	NumPy	http://www.numpy.org/ https://www.tutorialspoint.com/numpy
5	OpenCV	https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html https://docs.opencv.org/2.4/doc/tutorials/tutorials.html https://www.tutorialspoint.com/opencv/
6	Machine Learning Library in OpenCV	https://docs.opencv.org/3.4.1/dc/dd6/ml_intro.html https://www.learnopencv.com/tag/deep-learning/

Table 1.1 Technologies & Reviews

2. SYSTEM REQUIREMENT SPECIFICATION

2.1 User Characteristics

- Check IP Address of PC and Raspberry Pi
- Configure Raspberry Pi with PC
- Login into Raspberry Pi
- Train Model
- Check Accuracy and Precession

2.2 Hardware and Software Requirement

Software Requirement:

Sr.No	Software	Description
1	Miniconda	Platform to run Linux
2	Python	Programming Language
3	Raspbian OS	Raspberry Pi Operating system
4	NumPy	Provides additional features for array
5	OpenCV	Provides function and methods to perform Image Processing
6	Machine Learning Library in OpenCV	Provides methods and features for performing Machine Learning and neural network algorithms

Table 2.1 Software Requirement

Hardware Requirement:

Sr.No.	Hardware	Description
1	Raspberry Pi	To send the data to pc where required calculation and manipulation of data is done
2	Pi Camera	To identify the surrounding and provide required data to run machine learning algorithm on
3	Motor Driver	To control motion of motor
4	Lithium Polymer	To power all Components
5	Distance Sensor	To detect obstetrical
6	Accelerometer	To calculate displacement
7	GPS	To detect real time location
8	PC	To run neural network

Table 2.2 Hardware Requirement

2.3 Constraints

- **Hardware Limitations:**

  The hardware limitation is that there must be minimum 2GB RAM in microprocessor without that uploading live feed and sensors would not be possible and above cannot be effectively run without have quad core 1.4 GHz processor or above versions.

- **Software Limitations:**

  In this project python required. Also, it requires effective connection between all the sensors. Software for taking input is required to be developed in python. Further more this input is sent for analysis to Open CV ML.

- **Parallel Operation:**

  At a time more than one sensor data is put into AI for analysing the surrounding.

- **High Order Language Requirement:**

  Front End: Pygames

  Back End: Python, Miniconda and Open CV.

- **Safety & Security Considerations:**

  As our system is providing photo id at a time of registration so the system is secure for every user.

2.4 Timeline Chart & Process Model

- **TimeLine Chart**

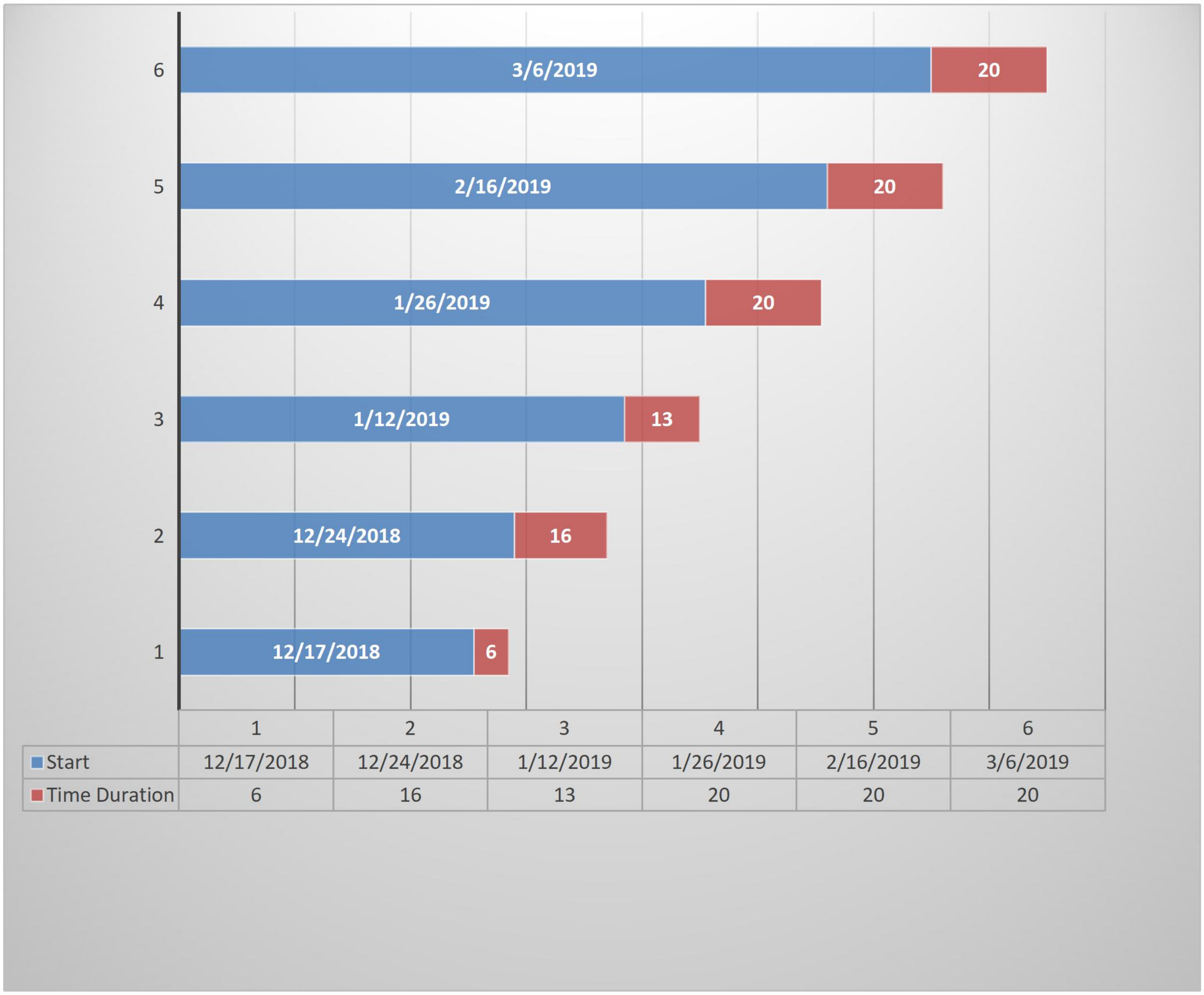

Fig 2.1 Timeline Chart

- **Process Model**

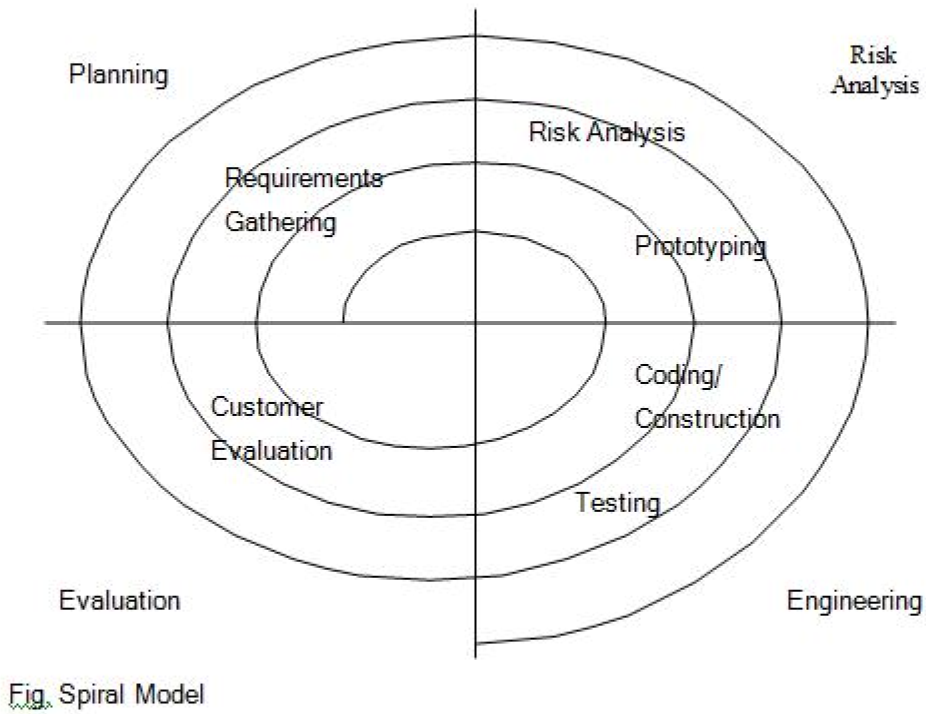

Fig 2.2 Spiral Process Model

- **Why we used spiral process model?**

The spiral model is similar to the incremental model with more emphases placed on risk analysis to identify list of risks and prototyping to address as many risks as possible. Spiral model uses a cyclic approach to incrementally develop the software product while decreasing its degree of risk.

This model uses four phases: Planning, Risk Analysis, Engineering and Evaluation. A software project repeatedly passes through these phases in cycles (called spirals in this model). In Spiral Model, before each cycle to involve stake holders to decide on scope of cycle to arrive at plan. Following this, risk analysis is carried out to arrive at list of risks Stake holders perform concurrent engineering to address these risks through prototyping and before the completion of current cycle. Software is produced in the engineering phase and tested at the end of the phase. The evaluation phase allows the customer to evaluate the output of the project before the project continues to the next spiral

Spiral models have many advantages like high amount of risk analysis so as to reduce rework effort and cost. Spiral models are found very effective for large and mission-critical projects. In addition, we enjoy the benefit of having software that is produced early in the software life cycle.

Spiral model has some disadvantages. Here entire project cost is very high since too many risk analyses and prototyping are involved. Also, spiral model demands highly specific expertise to perform risk analysis. Project’s success is highly dependent on the risk analysis phase and many times customers decide to cancel the project. Spiral model does not work well for smaller projects.

3. SYSTEM ANALYSIS

3.1 Study of Current System

Online shopping and improvement in automation in industrial work has lead to increase in movement of products and raw material. This increase in movement of goods and raw material being transported to various location. Companies are searching for cheap and cost effective ways to transport goods form one place to other. The already existing software relies on rigid programming which is incapable to handle the continuously changing environment of warehouses or other places. With such a technology the navigating and safety of such autonomous would increase. Not only the prices could be decrease but also more and more people would start developing there interest in machine learning programming. Increasing number of vehicles on road also is increasing the probabilities of accident. More then 89% of accidents are caused by mistakes made by humans. By implementing Smart RC Car, we can reduce these accidents and casualties. The present technology is very expensive by using our system this can be reduced. Making the concept access able to everyone. The Reduced cost will also increase the usability of it. Hence, it can be used in driving in city, on highways, and even off roading in future. The growing concept of Artificial Intelligence comes in play here. This increases the possibility of further improvement in system form constant use.

3.2 Problem and Weakness of Current System:

Problems of current system are:

- Multiple users might crash the system.
- Limited products available.
- Hard in maintainability.
- The system is highly complicated making it almost unusable. The cost of implementing such a system is very high.
- Limited options for in driving using AI.

3.3 Requirement of New System

At the present the problem is that people are not aware about availability of technology. The reduces cost can make its access able to everyone. This high use of system can increase the accuracy of system.

This high usage can increase the unambiguous driving trend and hence make road much safer. The increased usage will provide a verity of test cases which can then source to increasing the accuracy of system.

3.4 Feasibility Study

A feasibility study is undertaken to determine the possibility or probability of either improving the existing system or developing a completely new system. It helps to obtain an overview of the problem and to get rough assessment of whether feasible solution exists. This is essential to avoid committing large resources of project and then repent on it later.

- **Need for Feasibility Study:**

The feasibility study is needed to

- Answer the question whether a new system is to be installed or not?
- Determine the potential of the existing system.
- Improve the existing system.
- Know what should be embedded in the new system.
- Define the problems and objective involved in a project.
- Avoid costly repairs at later stage when the system is implemented.
- Avoid crash implementation of a new system.
- Avoid the „Hardware Approach" i.e. getting a computer first and then deciding how to use it.

There are three aspects in feasibility study portion of the preliminary investigation:

- **Technical feasibility**
- **Economic feasibility**
- **Operational feasibility**

- **Technical Feasibility:**

Technical Feasibility determines whether the work for the project be done with the present equipment, current procedures, existing software's technology and available personnel.

If new technology is needed then what alternatives will be needed in the present structure and work ethics?

This will require a close examination of the present system.

The technical feasibility should ask questions related to:

- Adequacy of available technology.
- Adequacy of hardware.
- Adequacy of computer.
- Operating time and support facilities, etc.

Technical feasibility determines whether the technology needed for the proposed system is available and how it can be integrated within the system and technical evaluation must also assess whether the existing system can be upgraded to use the new technology and whether the “Task Management System” has the expertise to use it.

The technical feasibility in the proposed system deals with the technology used in the system. It deals with the hardware and software used in the system whether they are of latest technology or not. It happens that after a system is prepared a new technology arises and the user wants the system based on that technology. Thus, it is important to check the system to be technically feasible.

The minimum memory requirement is 2GB of RAM while 4GB is better to have for better performance. As far as software is concerned installing interfacing modules and other required software.

- **Economic feasibility:**

  Economic feasibility looks at the financial aspects of the project. Economic feasibility concerns with the returns from the investments in a project. It determines whether it is worthwhile to invest the money in the proposed system. It is not worthwhile spending a lot of money on a project for no returns.

  To carry out an economic feasibility for a system, it is necessary to place actual money value against any purchases or activities needed to implement the project.

  The advocate plans to acquire the necessary hardware and software require for the system and there is no hindrance whether economical or otherwise towards its purchase. A brief description of the hardware and software required in the system is given later in the report.

- **Operational feasibility:**

  Operational feasibility covers two aspects. One is the technical performance aspect and other is the acceptance within the "Smart RC Car".

  Operational feasibility determines how the proposed system will fit in the current operations and what, if any job restructuring and retraining may be needed to implement the system.

3.5 Requirements Validations

- Port Number must be correct
- Socket Must be Correct
- Both Client and server socket must match
- Both Client and server socket must be free before running program
- Password to login in raspberry pi must be correct
- Username to login in raspberry pi must be correct
- Ssh must be one
- Wireless access must be on
- VNC viewer must be installed

3.6 System Diagram

3.6.1 Entity Relationship Diagram

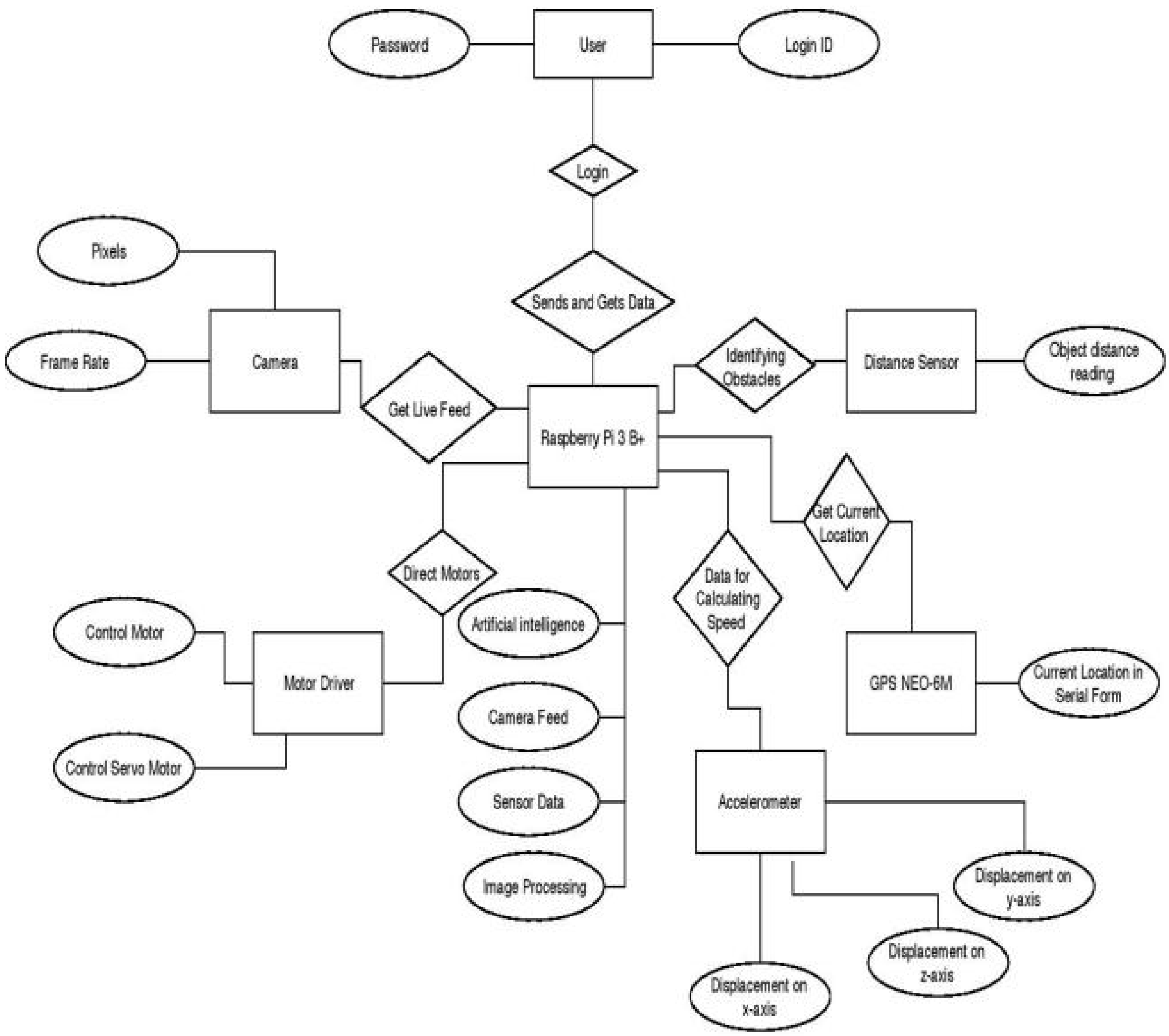

Fig 3.4 Entity Relationship Diagram

4. SYSTEM DESIGN

4.1 System Procedural Design

- **Raspberry pi Noods**

NOOBS is a way to make setting up a Raspberry Pi for the first time **much, much easier**. You won't need network access, and you won't need to download any special imaging software. Just head to the downloads page, grab a copy of the NOOBS zip file, and unpack it onto a freshly formatted 4GB (or larger) SD card. When you boot up for the first time, you'll see a menu prompting you to install one of several operating systems into the free space on the card. The choice means you can boot the Pi with a regular operating system like Raspbian, or with a media-centre specific OS like RaspBMC.

Once you've installed an operating system, your Pi will boot as normal. However, NOOBS stay resident on your card, so by holding shift down during boot you can return to the recovery interface. This allows you to switch to a different operating system, or overwrite a corrupted card with a fresh install of the current one; it also provides a handy tool to let you edit the config.txt configuration file for the currently installed operating system, and even a web browser so you can visit the forums or Google for pointers if you get stuck.

- **Python and its Features**

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Python's features include −

- Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Easy-to-read − Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain − Python's source code is fairly easy-to-maintain.
- A broad standard library − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Interactive Mode − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- Portable − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases − Python provides interfaces to all major commercial databases.
- GUI Programming − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- Scalable − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

- **Open CV**

    - Open CV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing; video capture and analysis including features like face detection and object detection.

Features of OpenCV Library

Using OpenCV library, you can −

- Read and write images
- Capture and save videos
- Process images (filter, transform)
- Perform feature detection
- Detect specific objects such as faces, eyes, cars, in the videos or images.
- Analyse the video, i.e., estimate the motion in it, subtract the background, and track objects in it.
- Open CV was originally developed in C++. In addition to it, Python and Java bindings were provided.
- Open CV runs on various Operating Systems such as windows, Linux, OSx, FreeBSD, Net BSD, Open BSD, etc.

- Open CV Library Modules

- Core Functionality

  This module covers the basic data structures such as Scalar, Point, Range, etc., that are used to build Open CV applications. In addition to these, it also includes the multidimensional array **Mat**, which is used to store the images. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.core**.

- Image Processing

  This module covers various image processing operations such as image filtering, geometrical image transformations, color space conversion, histograms, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.imgproc**.

- Video

  This module covers the video analysis concepts such as motion estimation, background subtraction, and object tracking. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.video**.

- Video I/O

  This module explains the video capturing and video codecs using OpenCV library. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.videoio**.

- calib3d

  This module includes algorithms regarding basic multiple-view geometry algorithms, single and stereo camera calibration, object pose estimation, stereo correspondence and elements of 3D reconstruction. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.calib3d**.

- features2d

  This module includes the concepts of feature detection and description. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.features2d**.

- Objdetect

  This module includes the detection of objects and instances of the predefined classes such as faces, eyes, mugs, people, cars, etc. In the Java library of OpenCV, this module is included as a package with the name **org.opencv.objdetect**.

- Highgui

  This is an easy-to-use interface with simple UI capabilities. In the Java library of OpenCV, the features of this module is included in two different packages namely, **org.opencv.imgcodecs** and **org.opencv.videoio**.

4.2 Flow of Data

4.2.1 Collect Training Data

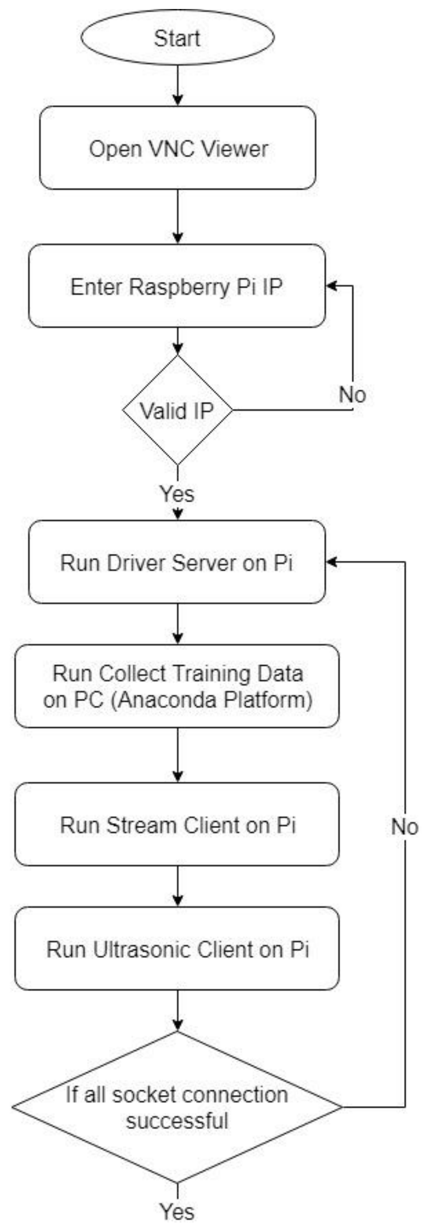

Fig 4.1 Collect Training Data Flowchart

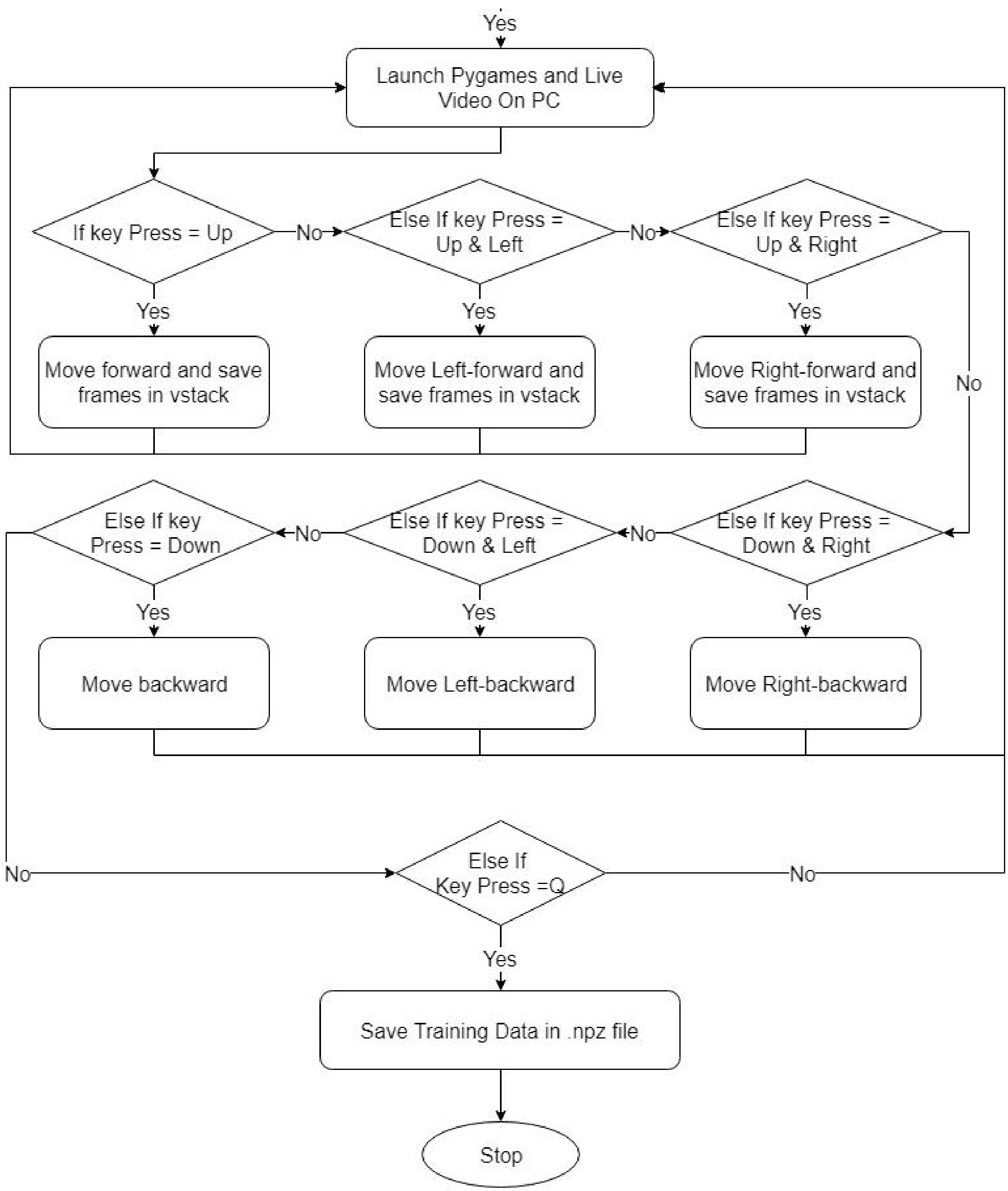

Fig 4.1 Collect Training Data Flowchart

4.2.2 Training Model

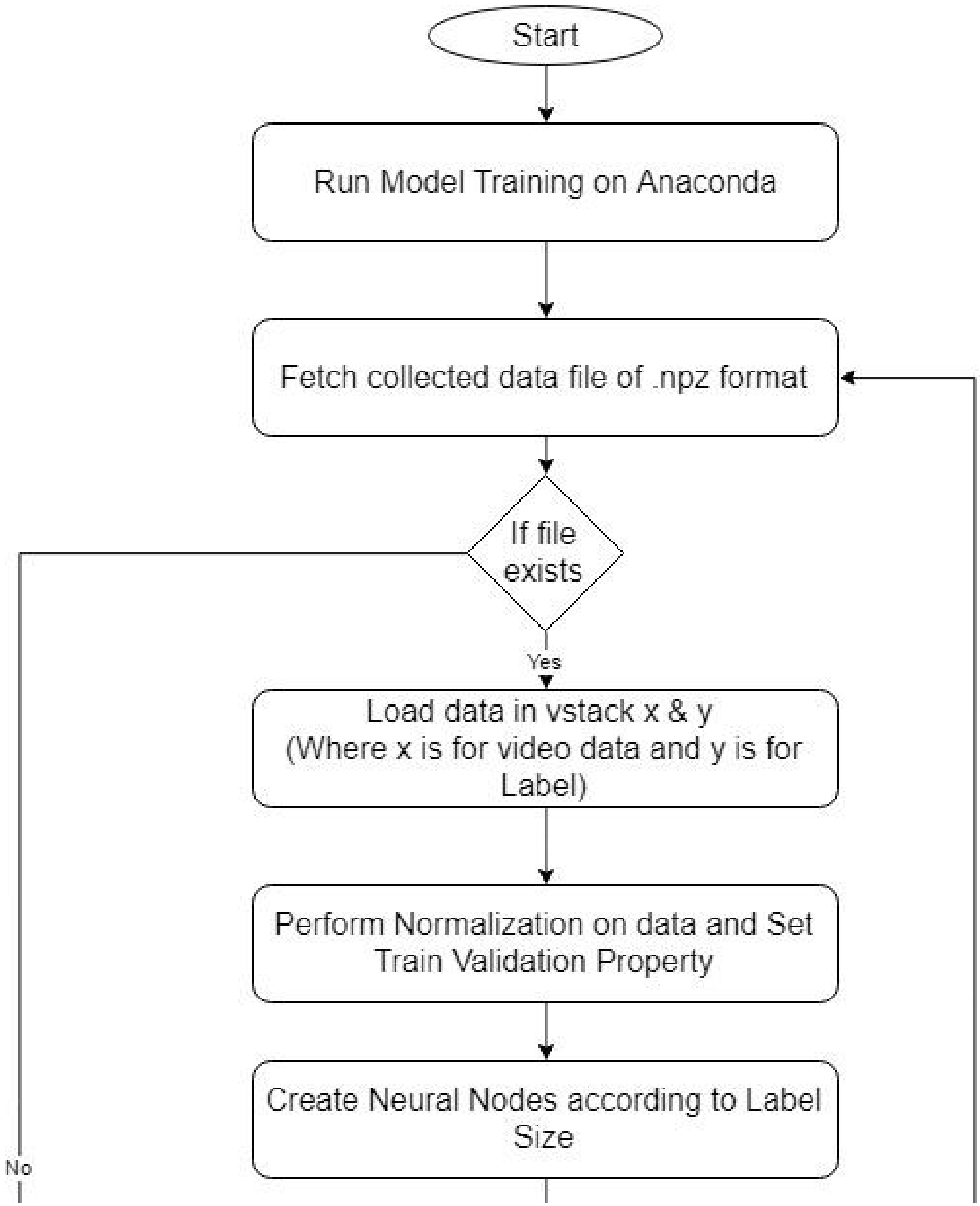

Fig 4.2 Model Training Flowchart

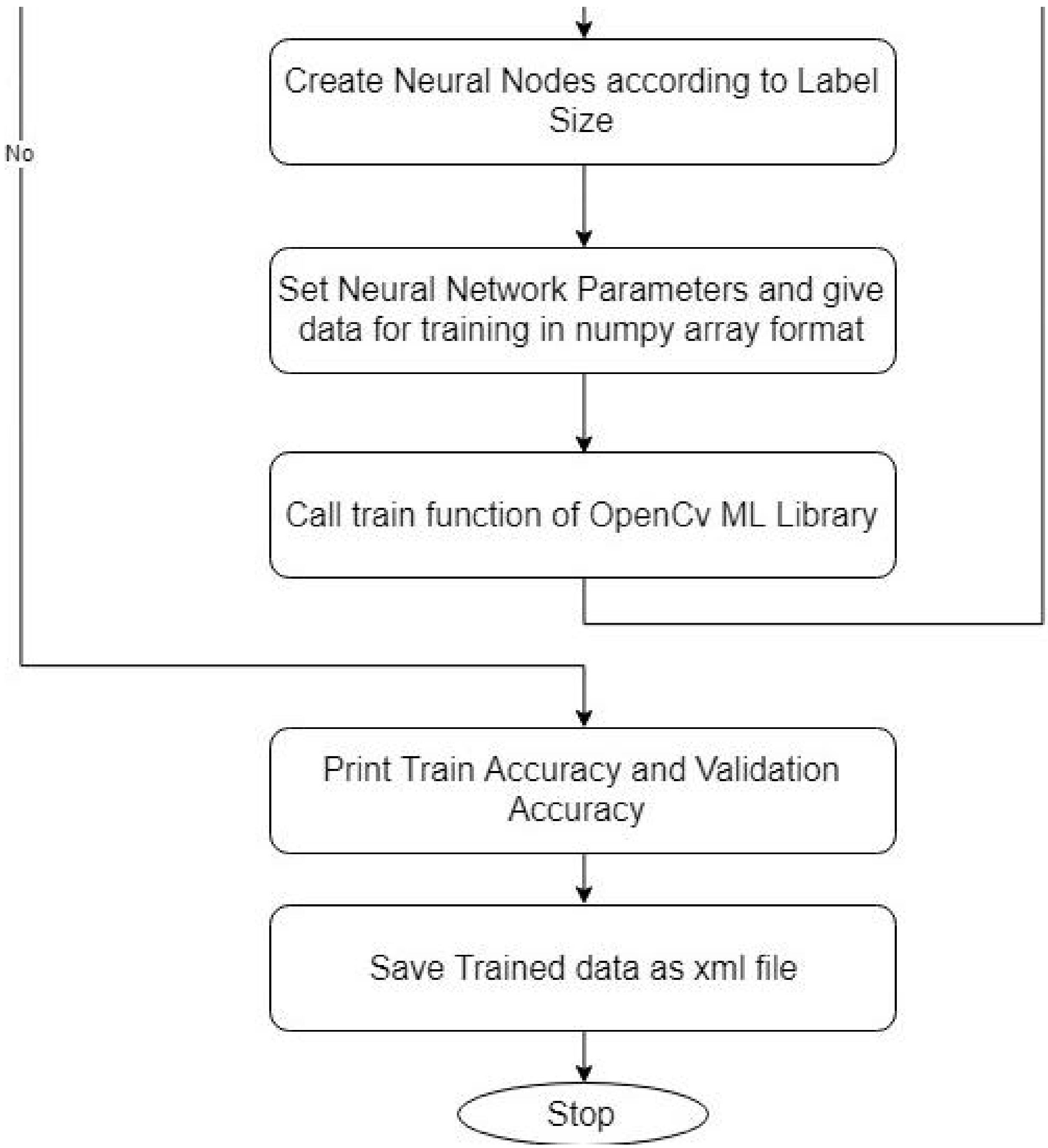

Fig 4.2 Model Training Flowchart

4.2.3 Self Driving

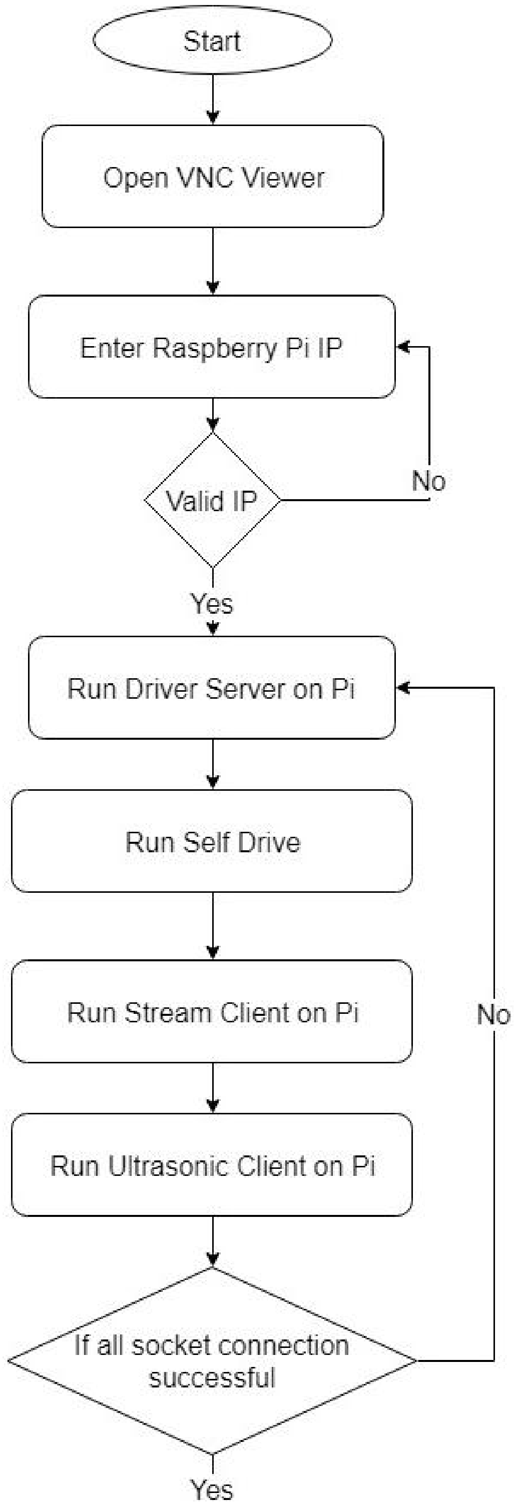

Fig 4.3 Self Driving Flowchart

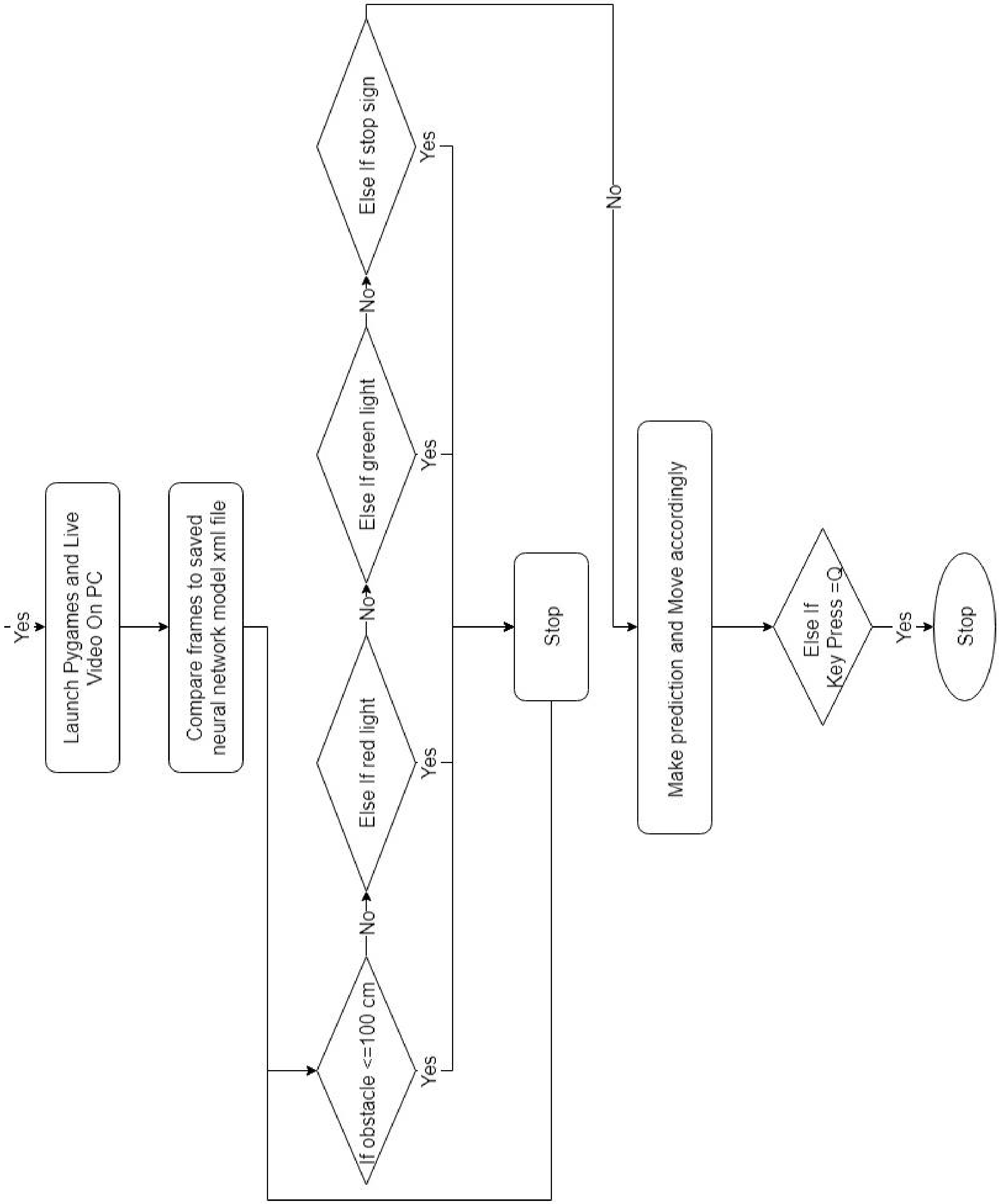

Fig 4.3 Self Driving Flowchart

5. IMPLEMENTATION AND PLANNING DESIGN

5.1 Implementation Environment

Computer

- OS Name: Windows 10
- OS Manufacturer: Microsoft Corporation
- System Type: 64-bit Operating System
- Processor: 5th generation Intel Core i5
- Windows Directory: c:\WINDOWS
- System Directory: c:\\\WINDOWS\system32
- BOOT Device:\Device\Hard disk\Volume32
- Locale: India Time Zone: (GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
- TotalPhysicalMemory:4.00 GB

Raspberry Pi 3 B+

- 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)

5.2 Module Specification

- Collecting Driving Data
- Applying Machine Learning Algorithm
- Pattern recognition
- Auto Driving
- Obstacle Detection
- Stop Sign Detection
- Red Light Detection
- Collecting Data only on click event
- Saving frames with instructions given

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 (320×120) 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).

5.3 Security Features

The security of your Raspberry Pi is important. Gaps in security leave your Raspberry Pi open to hackers who can then use it without your permission.

What level of security you need depends on how you wish to use your Raspberry Pi. For example, if you are simply using your Raspberry Pi on your home network, behind a router with a firewall, then it is already quite secure by default.

However, if you wish to expose your Raspberry Pi directly to the internet, either with a direct connection (unlikely) or by letting certain protocols through your router firewall (e.g. SSH), then you need to make some basic security changes.

Even if you are hidden behind a firewall, it is sensible to take security seriously. This documentation will describe some ways of improving the security of your Raspberry Pi. Please note, though, that it is not exhaustive.

5.4 Process of Implementation

test/

- rc_control_test.py : RC car control with keyboard
- stream_server_test.py : video streaming from Pi to computer
- ultrasonic_server_test.py : sensor data streaming from Pi to computer

model_train_test/

- data_test.npz : sample data
- train_predict_test.ipynb : a jupyter notebook that goes through neural network model in OpenCV3

raspberryPi/

- stream_client.py : stream video frames in jpeg format to the host computer
- ultrasonic_client.py : send distance data measured by sensor to the host computer
- driver_server.py : control RC car controller

computer/

- **cascade_xml/**
  - trained cascade classifiers
- **chess_board/**
  - images for calibration, captured by pi camera

- picam_calibration.py: pi camera calibration
- collect_training_data.py: collect images in grayscale, data saved as *.npz
- model.py: neural network model
- model_training.py: model training and validation
- rc_driver_helper.py: helper classes/functions for rc_driver.py
- rc_driver.py: receive data from raspberry pi and drive the RC car based on model prediction
- rc_driver_nn_only.py: simplified rc_driver.py without object detection

1. **Testing:** Flash rc_keyboard_control.ino to Arduino and run rc_control_test.py to drive the RC car with keyboard. Run stream_server_test.py on computer and then run stream_client.py on raspberry pi to test video streaming. Similarly, ultrasonic_server_test.py and ultrasonic_client.py can be used for sensor data streaming testing.

2. **Collect training/validation data:** First run collect_training_data.py and then run stream_client.py on raspberry pi. Press arrow keys to drive the RC car, press 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.

3. **Neural network training:** Run model_training.py to train a neural network model. Please feel free to tune the model architecture/parameters to achieve a better result. After training, model will be saved into newly created **saved_model**folder.

4. **Self-driving in action**: First run rc_driver.py to start the server on the computer (for simplified no object detection version, run rc_driver_nn_only.py instead), and then run stream_client.py and ultrasonic_client.py on raspberry pi.

6. Testing

6.1 Test Plan

Testing Planning involves how to plan testing before we are going to start making test suite. First step of testing is to test the System Module by Module that is once the module has been completed we test the module.

Then in second step I have tested all the modules by merging them one by one that are first module is checked then second module is merged with that module and both modules are checked together.

For this I have used both white box testing and black box testing. In white box testing structural testing is done so all the modules are tested one by one and finally when the project is completed black box testing is used to test the whole system together.

6.2 Test Strategy

The development process repeats this testing sub-process a number of times for the following phases:

- Unit Testing
- Integration Testing
- System Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct. System Testing ensures that the system meets its stated design specifications. Acceptance Testing is testing by the users to ascertain whether the system developed is a correct implementation of the Software Requirements Specification.

Testing is carried out in such a hierarchical manner to ensure that each component is correct and the assembly/combination of components is correct. Merely testing a whole system at the end would most likely throw up errors in components that would be very costly to trace and fix.

- **Unit Testing**

  The objective of Unit Testing is to test a unit of code (program or set of programs) using the Unit Test Specifications, after coding is completed. Since the testing will depend on the completeness and correctness of test specifications, it is important to subject these to quality and verification reviews.

- **Integration Testing**

  After testing all the components individually, the components are slowly integrated and tested at each level of integration. That is called integration testing.

- **System Testing**

  Finally, the fully integrated system is tested that is called system testing.

6.3 Test Methods

Involve execution and implementation of the software with test data and examining the outputs of the software and its operational behaviour to check that it is performing as required.

- **Black-box Testing**

  In Black-Box Testing or Functional Testing, Developers are concerned about the output of the module and software, i.e. whether the software gives proper output as per the requirements or not. In another words, this testing aims to test behaviour of program against it specification without making any reference to the internal structure of the internal structure of the program or the algorithms used. Therefore, the source code is not needed, and so even purchased modules can be tested. The program just gets a certain input and its functionality is examined by observing the output.

  This can be done in the following ways:

  - Input Interface
  - Processing
  - Output Interface

  The programs get certain inputs. Then the program does its jobs and generates a certain output, which is collected by a second interface. This result is then compared to the expected output, which has been determined before the test.

- **White-box Testing**

  White box testing is an important primary testing approach. Here code is inspected to see what it does. This test is designed to check the code. Code is tested using code scripts, driver, etc. which are employed to directly interface with and drive the code. The tester can analyse code and use the knowledge about the structure of a component to drive the test data.

- **Performance Testing**

  Performance testing is design to test the runtime performance of the system within the context of the system. This test is performed at module level as well as at system level. Individual modules developed by Developers are tested for required performance.

7.Screenshot

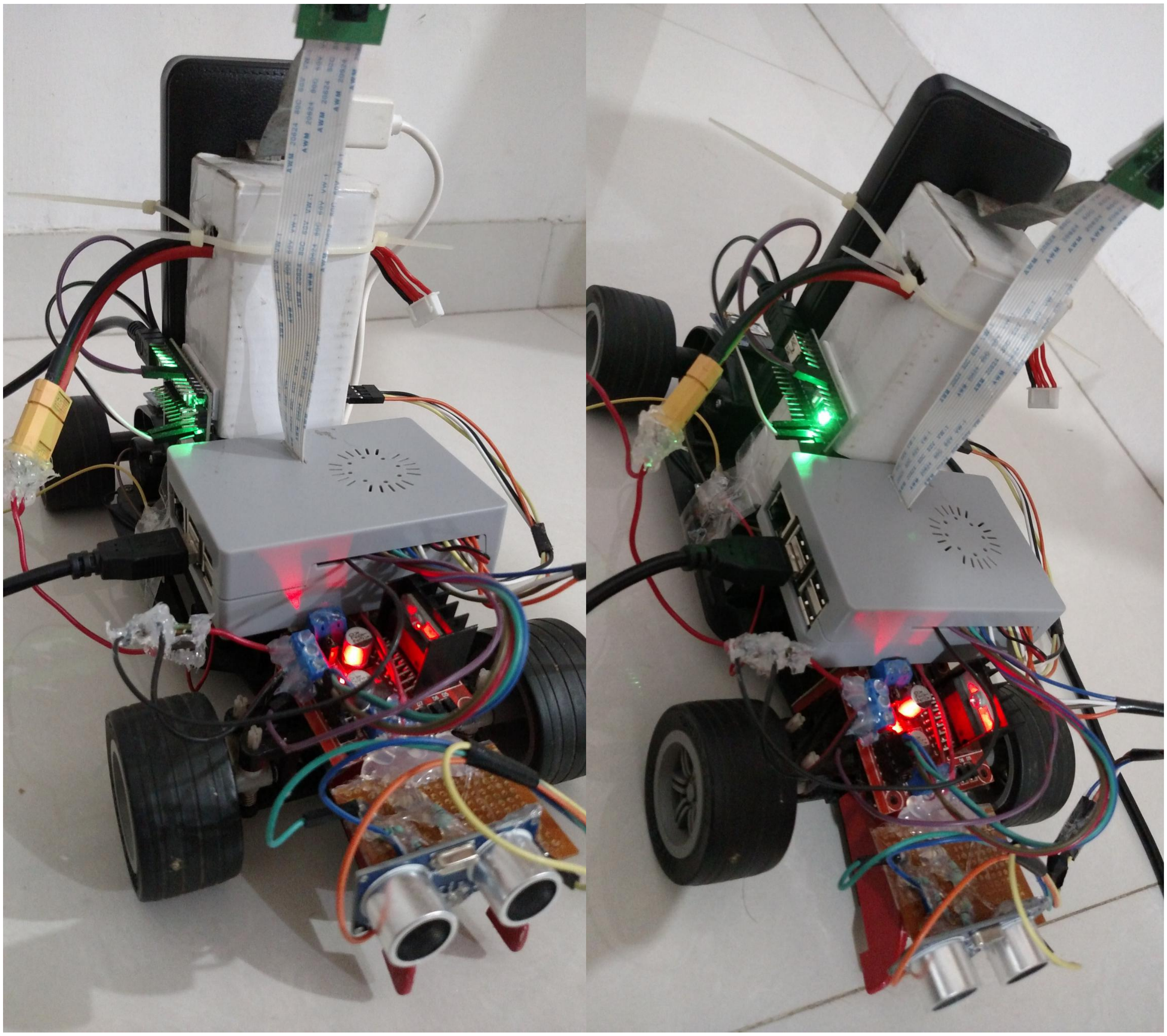

Fig. 7.1 Smart Rc Car Front

Fig 4.2 OpenCV

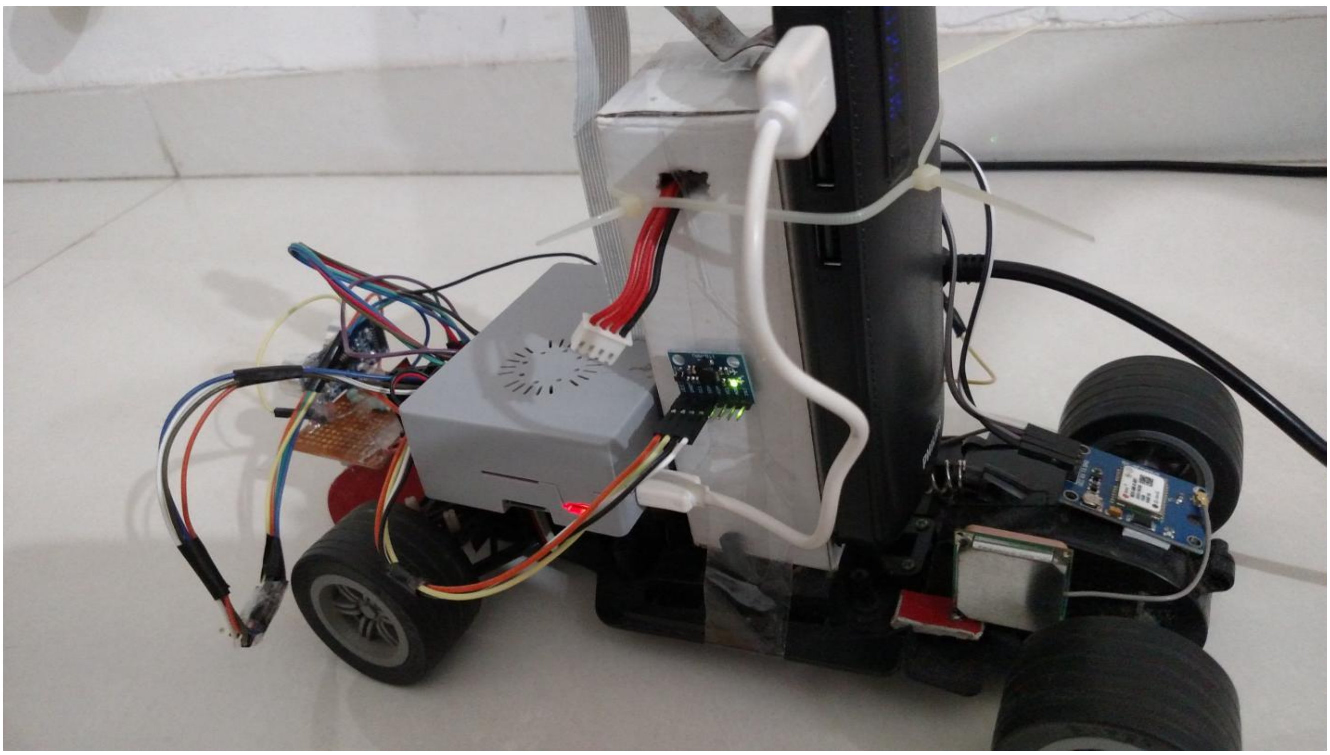

Fig 4.3 Smart Rc Back

Fig 4.4 Gps Neo6m

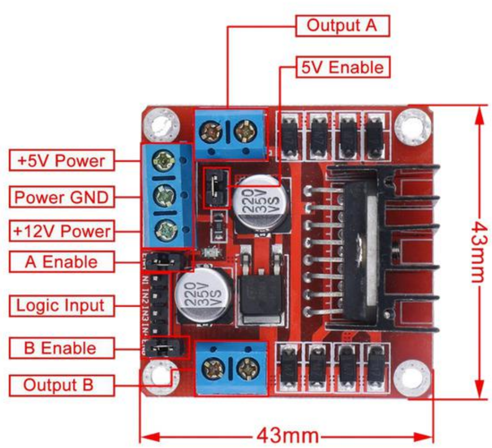

Fig 4.5 Motor Driver

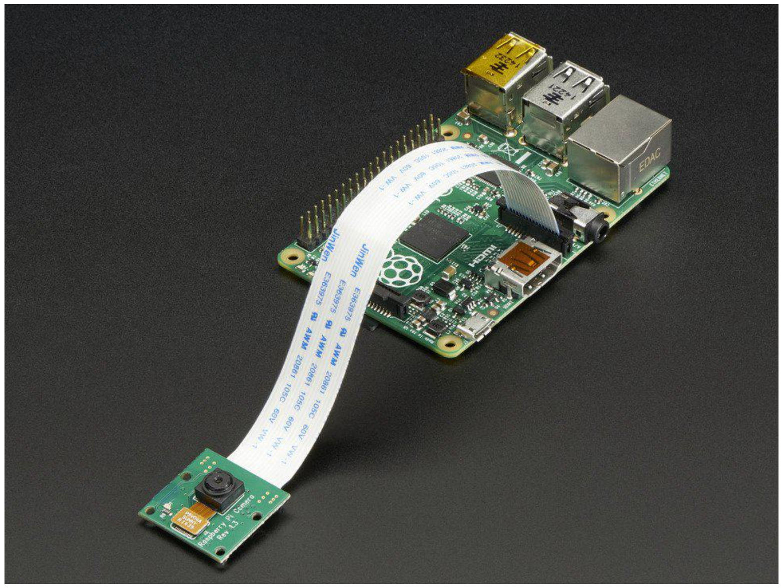

Fig 4.6 Connecting Pi Camera

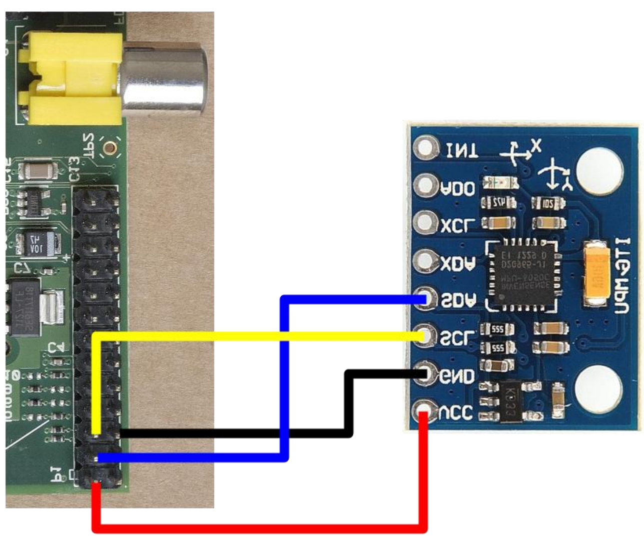

Fig 4.7 Connecting Accelerometer

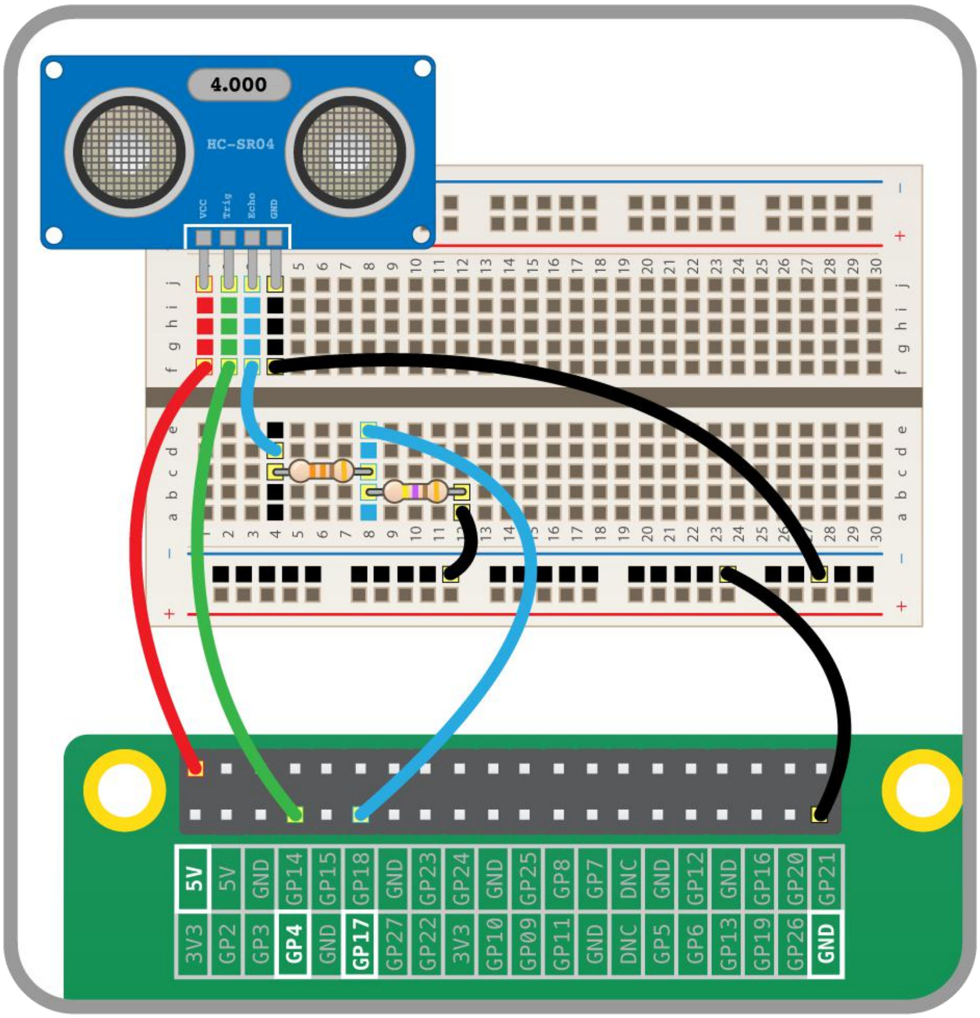

Fig 4.8 Connecting Distance Sensor

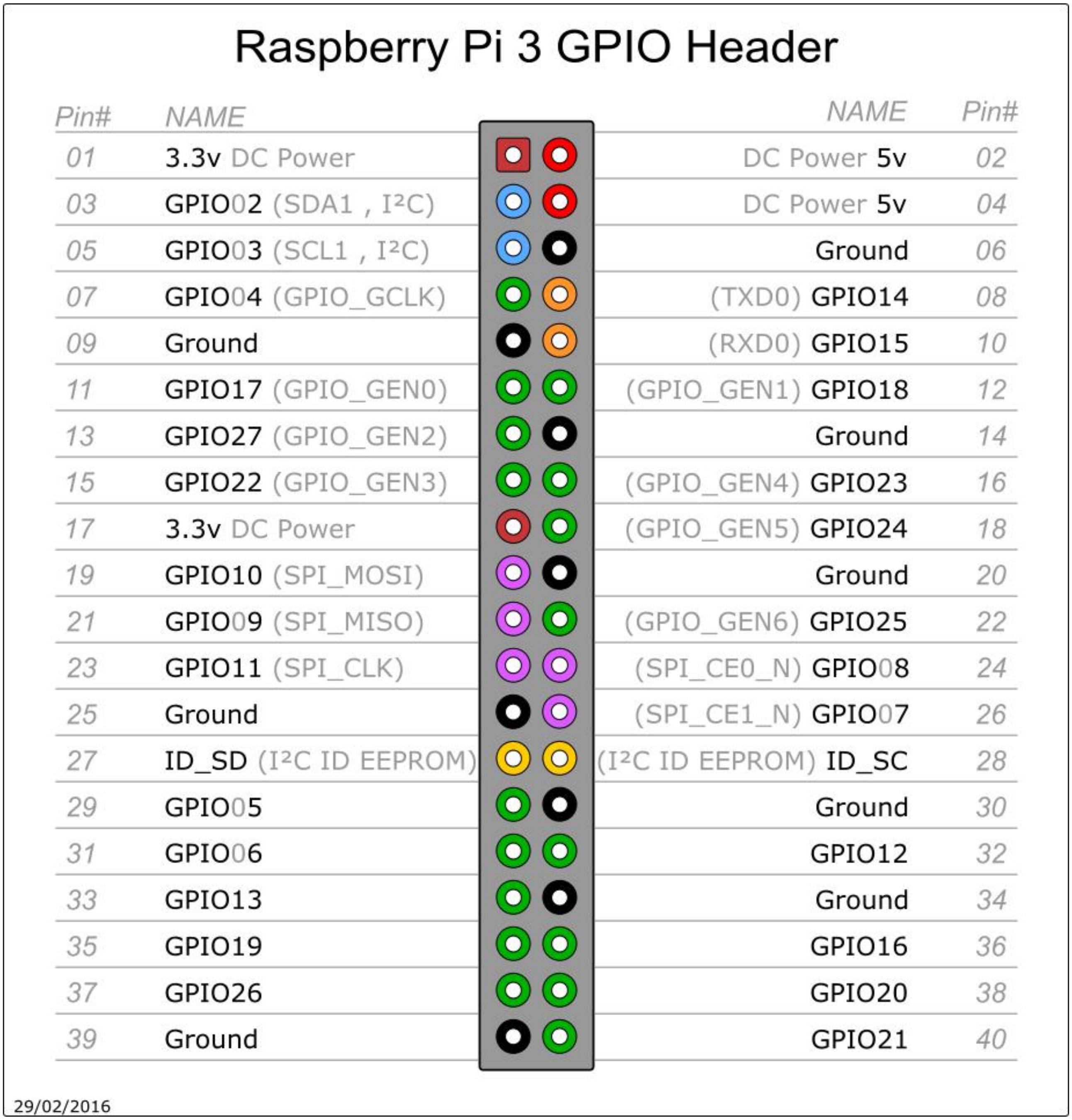

Fig 4.9 Raspberry Pi Pin Diagram

8. Limitation and Future Enhancement

8.1 Limitation

- Predictions can be made more accurate
- Other Object detection models can be added
- Speed of motor can be increased when prediction are made accurate
- More sensor can be added for more data
- Data can be filtered more to provide better learning to neural network

8.2 Future Enhancement

- Can provide more data by using multiple cameras and other sensors
- The model can be trained on more advanced neural network to work more effectively
- The data collection can be made faster
- Multiple car implementation can improve training time

9.Conclusion

.In summry, smart rc car is an neural network based project which can be scaled in various ways to fit requirments. Any one can easily drive car to collect data required for training phase. This data can then be computed to generate neural network model. Hence, making it capable to drive autonomusly. The accuracy of model directly proportional to provided error free data. Additionaly, red light detection, green light detection, stop sign detection and obstical detection makes rc car much more safe.

References

Sites

- https://www.tutorialspoint.com/python/python_basic_syntax.htm
- https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_tutorials.html
- https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/
- https://pimylifeup.com/raspberry-pi-distance-sensor/
- https://www.youtube.com/watch?v=bNOlimnWZJE
- https://randomnerdtutorials.com/guide-to-neo-6m-gps-module-with-arduino/
- https://www.sunfounder.com/blog/rpi-ard/
- https://www.youtube.com/watch?v=NPTK0inTldw

Research Paper

- https://www.academia.edu/38649035/International_Journal_of_Signal_and_Image_Processing_IJSP_IJSP
- https://www.engpaper.com/raspberry-pi-2018.htm