## Acknowledgment

We are very thankful for Tribhuvan University and Paschimanchal Campus for providing students with such an ample opportunity to investigate and work on a minor project from which we can learn much more than the course curriculum.

We are also very much thankful to the Head of Department (HOD) and Project Supervisor, Mr. Hari Prasad Baral for guiding and mentoring us throughout the project. This kind of project allows students to explore the boundaries of the current technologies and beware of all the development occurring in the tech world. This kind of exposure promotes proper growth of the students' overall character and shapes us to be a proper Engineer to face the challenges that await us all in the professional world.

# Table of Content


- Acknowledgment
- Table of Content
- Abstract
- Introduction
- Problem Statement
- Objectives
- Literature Review
- Feasibility Study
    - Technical Feasibility
- Development Approach
    - Hardware Development
    - Software Development and Functioning
- Methodology
    - Input unit
    - Processing unit
    - Control unit
    - Neural Network
    - Object detection
    - Distance Measurement
    - Remote motor control with Raspberry PI 3
- Diagrams
    - Circuit Diagram
    - Block Diagram
    - Manual Mode Activity Diagram
    - Autonomous Mode Activity Diagram
    - Neural Network Diagram

Use Case Diagram

Future Aspects

Conclusion

## Abstract

In this project, Self-Driving RC Car, we tend to build an autonomous remote controlled vehicle that can, with the help of manual training, learn and drive on its own. It is based on machine learning using a neural network with the help of a constant image stream supplied by the camera module on the Raspberry Pi 3 which forward the raw data to the server, receives the movement command and decides on its direction. It is an artificial intelligence system that gets better the more it is trained under different scenarios. The average amount of data that needs to be fed to deliver a practical self-driving vehicle is in hundreds of thousands. Thus due to the lack of data and machine to process big data, our system is not close to the practical performance yet it performs fairly well in a similar environment.

## Introduction

This project implements an autonomous driving feature to a remote-controlled car for improved safety concerns. The main purpose of this project is to demonstrate the usability of artificial intelligence in the field of transportation and the concept of machine learning and neural networking. With the completion of this project, we have properly illustrated the usability of this technology in big vehicles and how this improves road safety.

The study proposed the development of an autonomous vehicle, meaning it doesn't require a driver to drive. It makes the optimal use of the traced path in the 5m*5m area and is trained by manual driving for hours.

## Problem Statement

With the growing effect of technology in our day to day life, it's no wonder that transportation cannot be affected by the trend. With every inch of map digitized and various sensors built, technology has gone far enough to drive a car without any driver. The self-driving car is still an experimental subject and will take some time to come fully in the streets. The need for technology and machine comes when we want to eliminate the risk of human judgments, which is very common in the field of driving. There is a number of accidents occurring on a daily basis all around the globe causing loss of human lives on the scale of thousands. Thus, to issue safely in driving and revolutionize the art of driving, self-driving cars have been introduced. Self-driving car not only allows the safest driving experience but with the development of algorithms like least distance finding and with the help of machine learning, we can now find the shortest path to our destination to travel most fuel-efficient. Although a lot of data has to be used to teach a machine to drive by itself, it is worth the price and effort and we believe that this is the future of driving. With the full application of automated vehicles, we can hope for more organized traffic and hopefully find low congestions in the road as compared to the present state.

## Objectives

Although being a new concept, the idea of self-driving cars is pretty much defined and straight forward. The application of machine learning and the neural network goes hand in hand in the development of this technology. Machine learning generally means programming the machine to learn by itself from the data it is provided and eliminate the redundancy of the programmer having to make every small minor change to its code every time there is fluctuation in the incoming data. This allows the full extent of the application of artificial intelligence and helps to create a more “perfect” machine overall. Thus, to develop a fully functioning self-driving car, we must accomplish the following objectives:

- Prepare a complete map of the surrounding environment, highlighting the paths the car must travel.
- Issue a good obstacle recognition system to prevent unwanted accidents
- Issue preventive measures to be taken by the car upon encounter with obstacles
- Feed the necessary data to the system to enable it to auto-drive upon request
- Use the remainder data to check its accuracy and compute the usability

## Literature Review

Given that this is a concept under development, its experiment has been conducted at various places. Companies like Tesla, Google, Apple, Audi has done plenty of research and developed some near-perfect autonomous vehicles.  At the CES 2017, Nvidia unveiled its latest progress in the field of autonomous driving by presenting an autonomous jeep that looked like an ordinary one from the exterior but housed a powerful programmed computer devoted to the self-driving feature. Apart from the real-world cars, a few experiments have been conducted on the miniature model cars like RC cars. The majority of the automated RC cars have been developed using Arduino while we chose to opt to use a Raspberry Pi. This uncommon choice of the device allows us to deliver better performance out of the car, meaning we can track obstacles more effectively and faster learning process on the machine.

Some demonstrative systems, precursory to autonomous cars, date back to the 1920s and 1930s. The first self-sufficient (and therefore, truly autonomous) cars appeared in the 1980s, with Carnegie Mellon University’s Navlab and ALV projects in 1984 and Mercedes-Benz and Bundeswehr University Munich’s Eureka Prometheus Project in 1987. A major milestone was achieved in 1995, with CMU's NavLab 5 completing the first autonomous long-distance drive in the United States. Of the 2,849 miles between Pittsburgh, PA and San Diego, CA,  2,797 miles were autonomous (98.2%), completed with an average speed of 63.8 miles per hour (102.3 km/h). Since then, numerous major companies and research organizations have developed working prototype autonomous vehicles.

## Feasibility Study

The importance of a preliminary investigation is to determine whether the system being developed is feasible or not. Self-Driving Car is introduced to decrease the number of vehicle accidents occurring due to the carelessness of the drivers. The introduction of Self-driving not only decreases the accident rate but also allows the existence of organized traffic which can be helpful to a country like ours considering the present traffic conditions. While It might be unfeasible to build a big Self Driving Car, It's worthwhile to build and learn from Remote controlled car.

The feasibility study did are:

### Technical Feasibility

Technical feasibility accesses the current resources (hardware and software) and technology, required to accomplish user requirements in the system within the allocated time and budget.

The "Self-Driving RC Car" is technically feasible. The primary technical requirements are listed down below:

**Hardware:**

- Raspberry Pi 3
- Pi Camera
- 5V Motors (*2)
- Relay (*4)
- Transistor
- Diodes
- Power Source (10000 mAh)
- Wheels (*3)

**Software:**

- Rasbian OS
- Node JS
- Python
- Tensorflow library
- Keras library

### Legal Feasibility

Legal feasibility determines whether the proposed system conflicts with the legal requirements. Our project well complies with the local legal standards set on the field of our project and thus the proposed venture is acceptable in accordance with the laws of the land.

## Development Approach

The development of our project has been conducted in two phases hardware and software. Utilizing the limited resources, we were supplied with, we managed to assemble the minimal hardware components needed to make our project run.

### Hardware Development

The primary component needed in our Self-Driving RC Car is the Raspberry Pi 3. It controls the motors driving the car itself and with its powerful processor analyzes the image stream supplied by the Pi Camera and then decides on how the car should operate on the required conditions.

The Raspberry Pi is connected with a custom-built relay module though which it drives the motors of the car. It is powered with the help of a portable power bank that is attached to the car itself. The Raspberry Pi, in turn, provides the necessary logic to control the motor from its GPIO pins.

## Software Development and Functioning

The software runs in two modes: auto and manual mode. The video stream from raspberry pi can be viewed from the computer. Input controls are to be provided in manual mode by the user while in auto mode the computer program itself sends the input controls. A TCP/IP server runs in a computer from which images captured from raspberry pi are streamed in the computer. With this server, input controls are also sent to the raspberry pi. A python script runs in the raspberry pi that sends high/low values to GPIO pins. In manual mode, a program runs on a computer that asks for input controls. The images are saved in a directory. The images and input controls (in form of numpy array) are saved in a CSV file as trained data.

During training, the images and input controls are split into train, validation and test data. The training data is passed to a complex convolutional network. This network adjusts weights and biases based on the data. Then hyperparameters are tuned with the help of validation data and accuracy is calculated using test data. If the accuracy is acceptable, then the weights and biases are saved as h5 file.

In auto mode, a program that is run on the computer takes each image frame from the Raspberry Pi camera and passes this numpy array of the image to the convolutional network. This convolutional net then predicts output for each of the image. At the same time, another program is run on the computer that detects an object and calculates distance. If then object is a stop signal and minimum distance is reached, then the stop signal is sent to a raspberry pi.

# Methodology

The system consists of three major parts:

- Input unit (pi camera and ultrasonic sensor)
- Processing unit (computer)
- Control unit (raspberry pi and motors)

### Input unit

A raspberry pi attached with a pi camera and ultrasonic sensor can be used to collect input data. Two scripts run on raspberry pi for streaming video and ultrasonic data to a computer via a local Wi-Fi connection using sockets.

### Processing unit

The processing unit handles the following tasks:

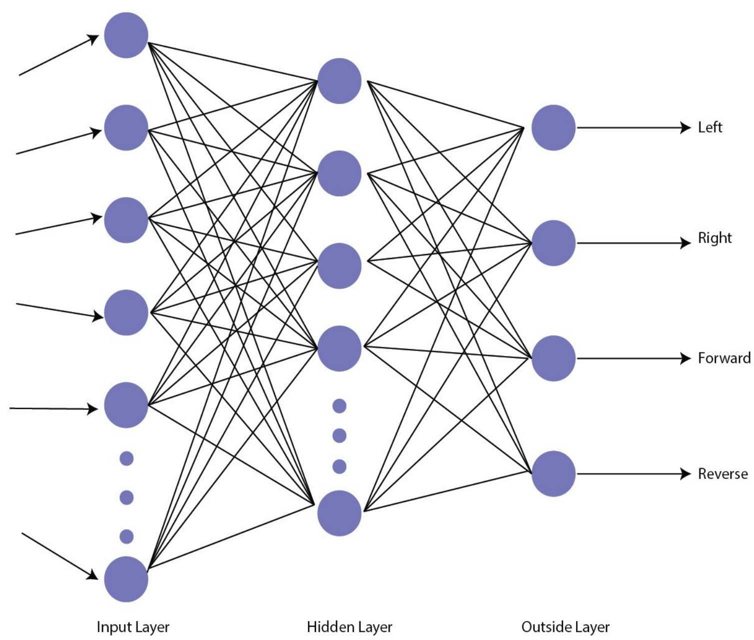
- Receiving video and ultrasonic data from raspberry pi
- Using the input video data to train the convolutional neural network and predict control output (i.e. steer)
- Detect objects and calculate distance.
- Use ultrasonic sensor’s data to detect collision.

### Control unit

The car is controlled using the raspberry pi. First, the input is sent manually from pc (which trains the neural network). Then, the predicted input from the neural network is used to control the car autonomously.

## Neural Network

Neural networks are the most advanced and efficient machine learning algorithms that mimic the human brain. In the neural network that we will be using, the input frames of images will be taken as input nodes and steering controls (forward, reverse, left, right) will be output labels.



## Object detection

To detect objects, Haar feature-based cascade classifiers can be used. Since each object requires its own classifier and follows the same process in training and detection, this project only focused on a stop sign and traffic light detection. Thus, can be done easily in OpenCV


## Distance Measurement

To measure the distance from an object (i.e. the stop sign triangle similarity can be used. Let's say we have a marker or object with a known width W. We then place this marker some distance D from our camera. We take pictures of our object using our camera and then measure the apparent width in pixels P. This allows us to derive the perceived focal length F of our camera:

$D = (W * F)/ P$

The width and pixels can be taken from python's OpenCV library

**Remote motor control with Raspberry PI 3**

A motor can be driven forward or backward depending on which way around current flows through it. Since there are two modes of operation of Self Driving Car; i.e. Training or Manual Operation and Self or Autonomous Operation. For Training or Manual operation, the control of the vehicle is done manually through commands from PC that are transferred to Raspberry PI or the commands from the Mobile phone transferred through a local Wi-Fi connection. A TCP connection is made by which the communication of PC and Raspberry PI is done. A database is accessed by Raspberry PI in which the control commands are stored.

Raspberry Pi Consists of 40 pins out of which 26 pins are GPIO pins and remaining are power and ground. Two motors from the vehicle have 4 pins that are two be connected to GPIO pins of RPi.

The four pins from the ultrasonic sensor are connected to RPi out of which two are connected to GPIO pins and the remaining two are connected to power and ground.

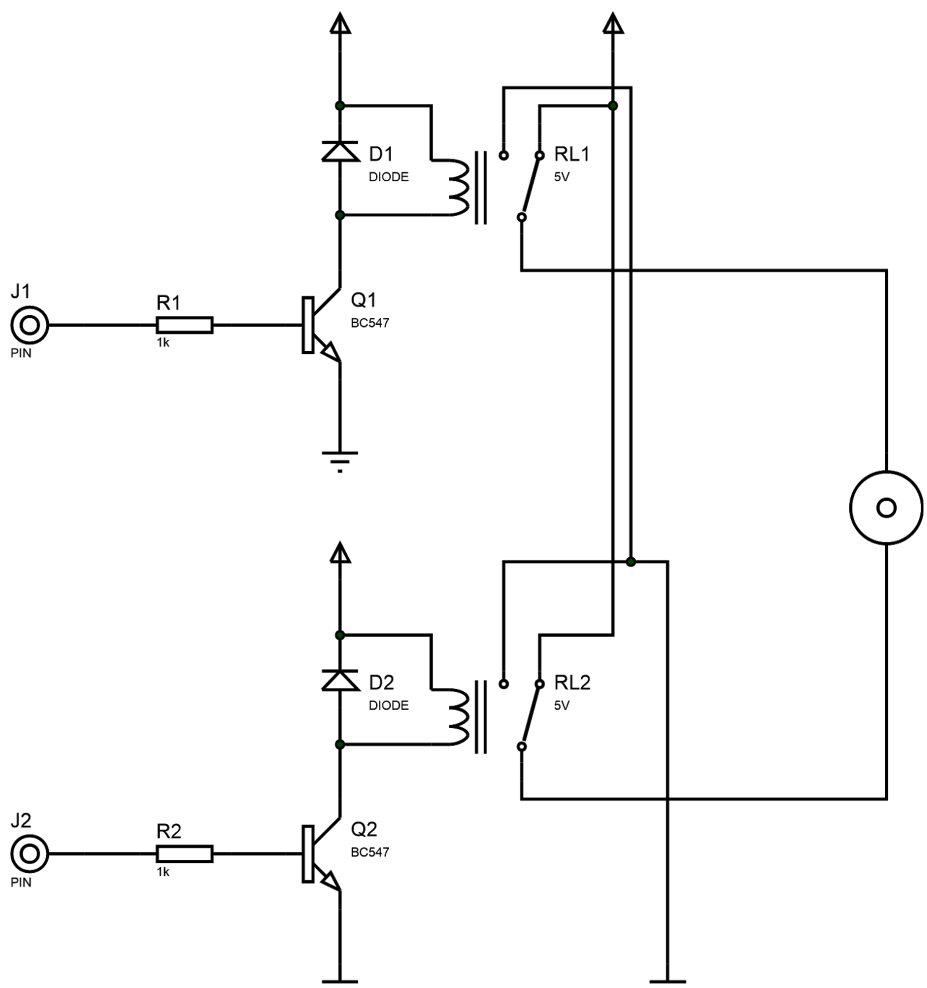
The Pi camera is connected to RPi through the camera slot.

The RPi is driven by a 6vpower supply.

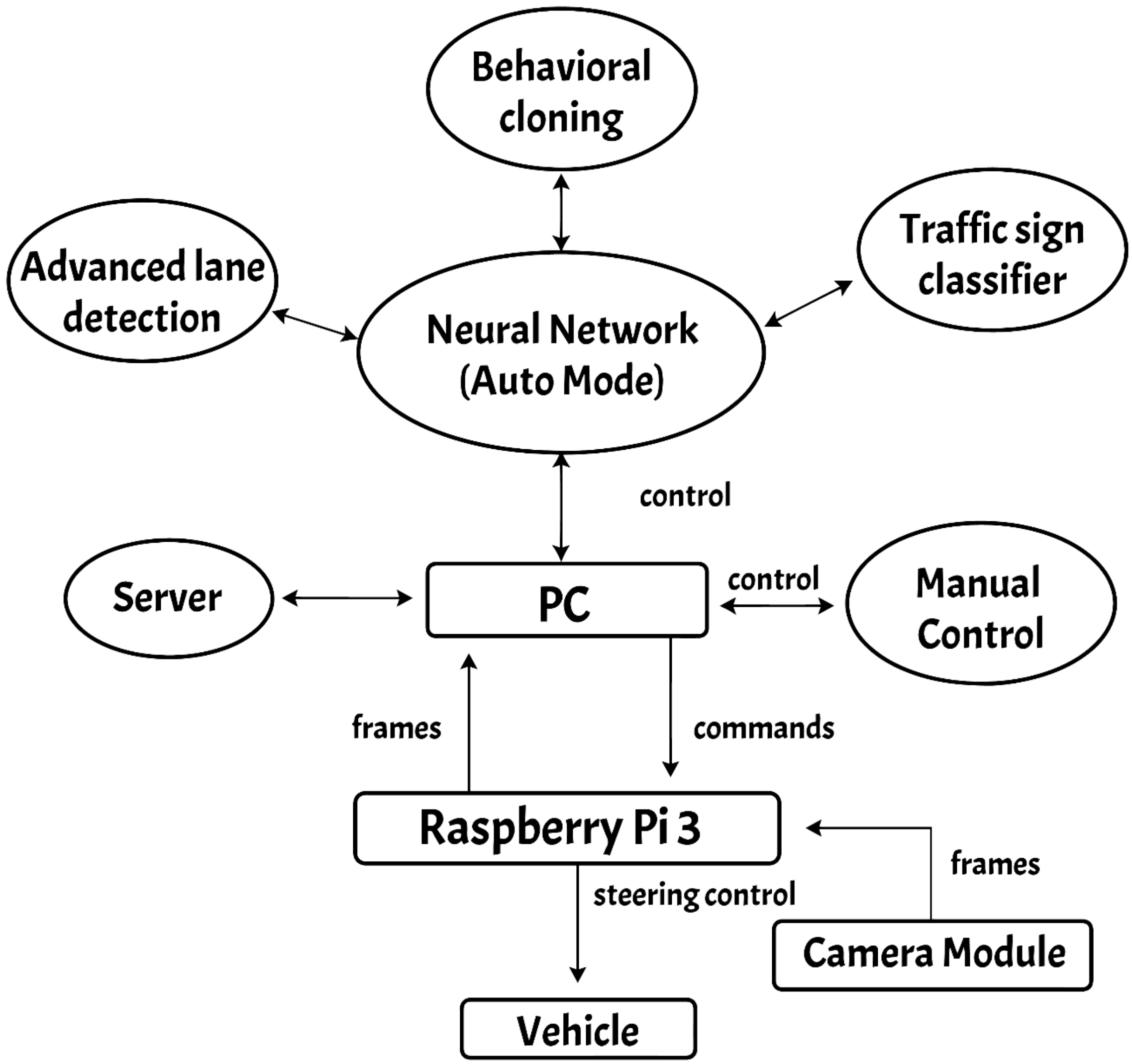
## Diagrams

Self driving RC car can be represented in various diagrams for ease of understanding. Circuit Diagram, Block Diagram, Manual Mode Activity Diagram, Autonomous Mode Activity Diagram, Convolution Layer Diagram and a Use Case Diagram are represented below.

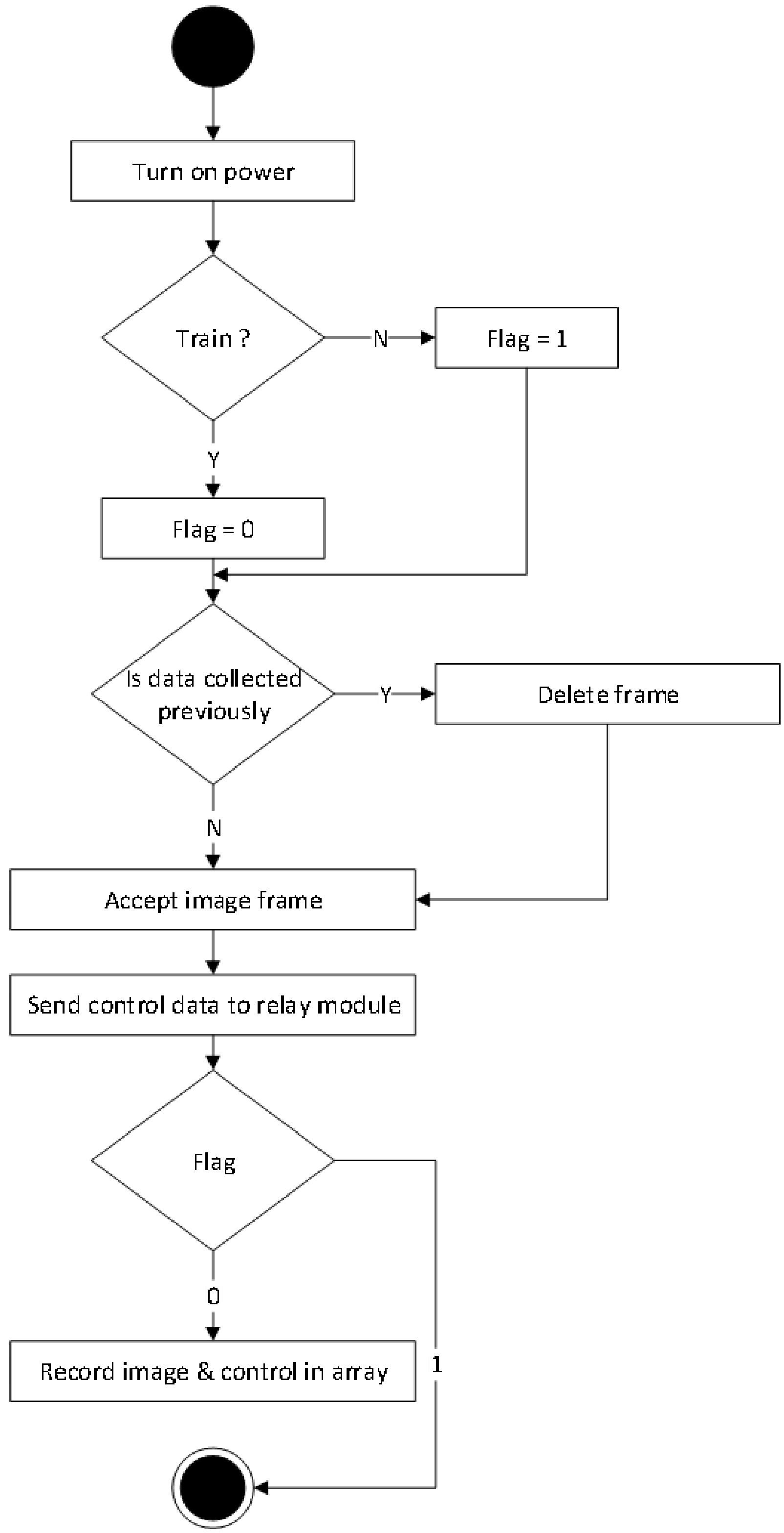
### Circuit Diagram



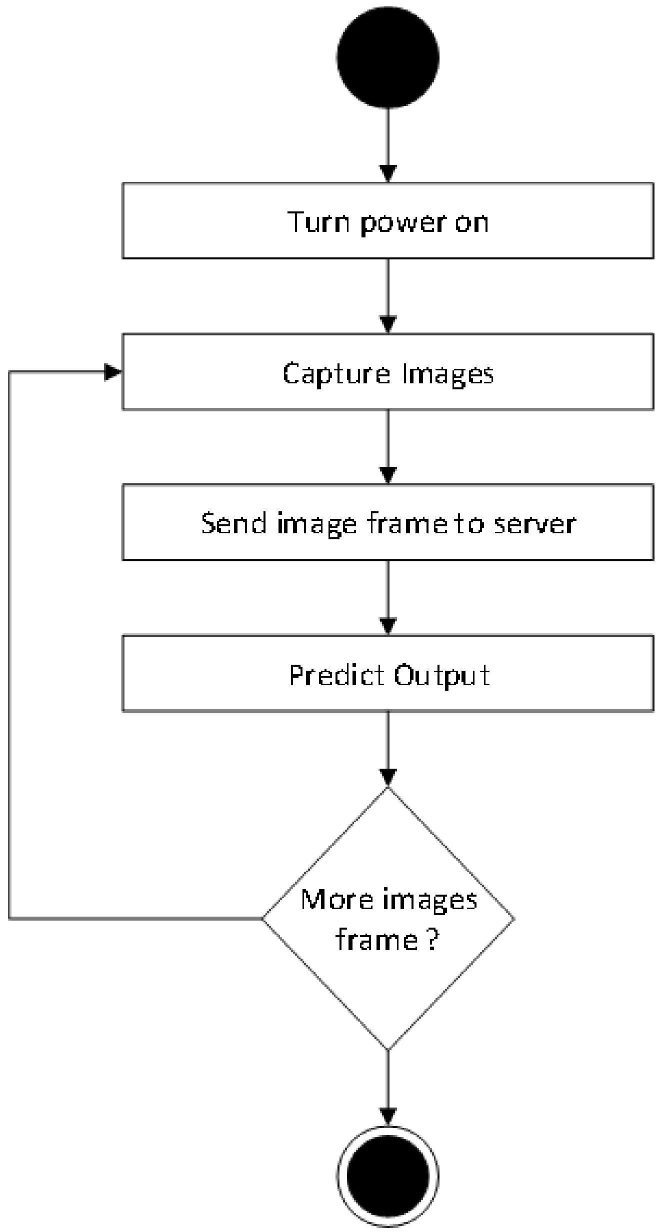
**Block Diagram**



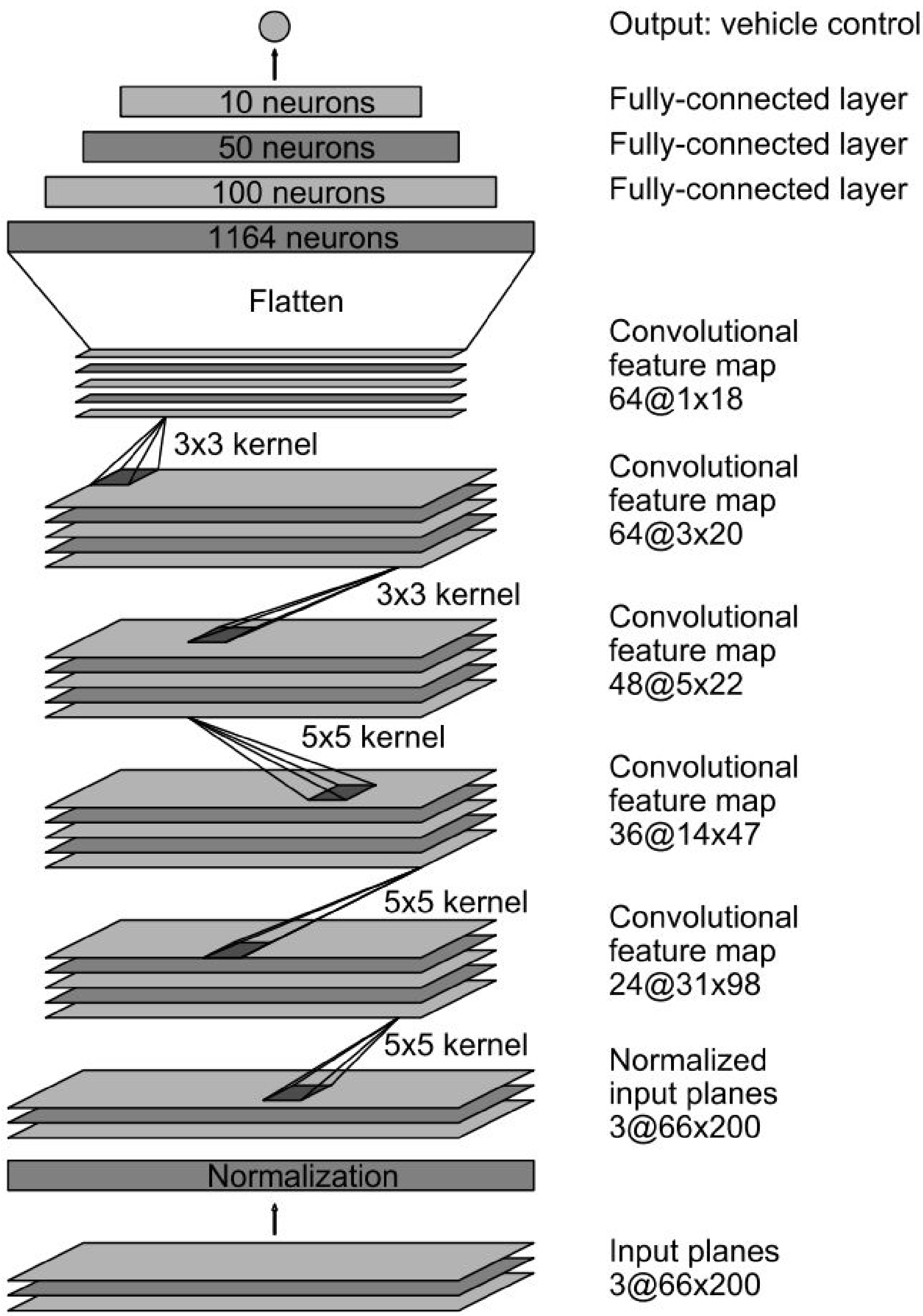
## Manual Mode Activity Diagram



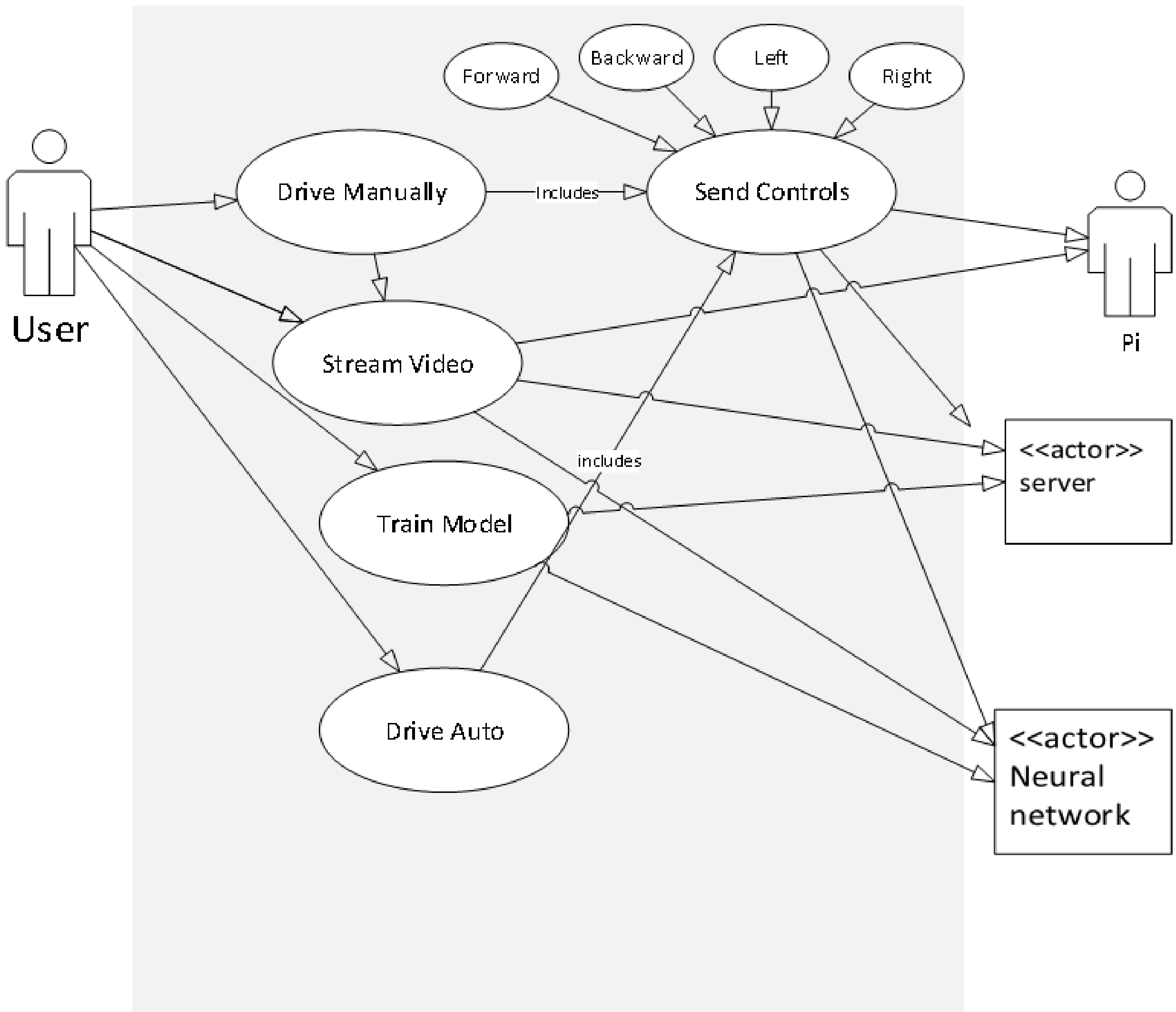
### Autonomous Mode Activity Diagram



**Convolution Layer Diagram**



## Use Case Diagram



## Algorithm

Power is turned on in both server (computer) and client (raspberry pi and components)

Raspberry pi camera, raspberry pi socket client and Rpi Rest API server is opened

Socket Server is opened from a computer

Server/client are connected

Check if its training mode or autonomous mode

If (training mode)

    Stream camera and capture camera frames from rpi camera via socket

    Enable controls from the computer

    While (controlling)

        Save image frame label data and control label

            Send control data through rest api

            Rpi responds to rest api client data and car is controlled

    Stop control

    Trained data is saved into npy file

Else

    Stream camera and capture camera frames from rpi camera via socket

        While (controlling)

        Compare image frames label with control label

        Predict control based on comparison

            Send control data through rest api

            Rpi responds to rest api client data and car is controlled

    Stop control

## Future Aspects

Considering the present development of technology in the transportation field, the idea of Self Driving Cars is still a futuristic concept. The current road conditions and lack of proper digitization will prove a major hindrance to deploy the concept right now on the road. Also, the required digital map for it to operate is not good enough for it to operate fully right now.

However, with the pace of digitization that has been going on, it is possible that the needs of the Self-Driving Car will be met on the upcoming years and the vehicle will be able to function fully on the road itself.

Mobilizing the vehicle will allow efficient fuel consumption and reduce the number of car accidents by a great deal. Although the initial purchase cost will be significantly greater than the normal vehicle, it will certainly pay off to purchase this kind of vehicle in the long run. For better performance, this type of vehicle needs to be trained in multiple scenarios and thus the more you use it the better it will be.

## Conclusion

Our project focuses on machine learning and exhibiting artificial intelligence by a vehicle. Our project is very much feasible under the futuristic circumstances more than the present conditions. This project was completed upon the estimated time and is now under constant training through which its computation and decision making will be refined more and more. With the availability of some additional equipment, we can further expand this project which can be more autonomous as well as more of a practical car.